Deep Image Retrieval: A Survey

Wei Chen, Yu Liu, Weiping Wang, Erwin M. Bakker, Theodoros Georgiou,
Paul Fieguth, Li Liu, *Senior Member, IEEE*, and Michael S. Lew

Abstract—In recent years a vast amount of visual content has been generated and shared from various fields, such as social media platforms, medical images, and robotics. This abundance of content creation and sharing has introduced new challenges. In particular, searching databases for similar content, *i.e.*, content based image retrieval (CBIR), is a long-established research area, and more efficient and accurate methods are needed for real time retrieval. Artificial intelligence has made progress in CBIR and has significantly facilitated the process of intelligent search. In this survey we organize and review recent CBIR works that are developed based on deep learning algorithms and techniques, including insights and techniques from recent papers. We identify and present the commonly-used benchmarks and evaluation methods used in the field. We collect common challenges and propose promising future directions. More specifically, we focus on image retrieval with deep learning and organize the state of the art methods according to the types of deep network structure, deep features, feature enhancement methods, and network fine-tuning strategies. Our survey considers a wide variety of recent methods, aiming to promote a global view of the field of instance-based CBIR.

Index Terms—Content based image retrieval, Deep learning, Convolutional neural networks, Literature survey

1 Introduction

Content based image retrieval (CBIR) is the problem of searching for semantically matched or similar images in a large image gallery by analyzing their visual content, given a query image that describes the user's needs. CBIR has been a longstanding research topic in the computer vision and multimedia community [1], [2]. With the present, exponentially increasing, amount of image and video data, the development of appropriate information systems that efficiently manage such large image collections is of utmost importance, with image searching being one of the most indispensable techniques. Thus there is nearly endless potential for applications of CBIR, such as person re-identification [3], remote sensing [4], medical image search [5], and shopping recommendation in online markets [6], among many others.

A broad categorization of CBIR methodologies depends on the level of retrieval, *i.e.*, instance level and category level. In instance level image retrieval, a query image of a particular object or scene (*e.g.*, the Eiffel Tower) is given and the goal is to find images containing the same object or scene that may be captured under different conditions [7], [8]. In contrast, the goal of category level retrieval is to find images of the same class as the query (*e.g.*, dogs, cars, *etc.*). Instance level retrieval is more challenging and promising as it satisfies specific objectives for many applications. Notice that we limit the focus of this survey to instance-level image retrieval and in the following, if not further specified, "image retrieval" and "instance retrieval" are considered equivalent and will be used interchangeably.

Finding a desired image can require a search among thousands, millions, or even billions of images. Hence, searching efficiently is as critical as searching accurately, to which continued efforts have been devoted [7], [8], [9], [10], [11]. To enable accurate and efficient retrieval of massive image collections, *compact yet rich feature representations* are at the core of CBIR.

In the past two decades, remarkable progress has been made in image feature representations, which mainly consist of two important periods: feature engineering and feature learning (particularly deep learning). In the feature engineering era (*i.e.*, pre-deep learning), the field was dominated by milestone hand-engineered feature descriptors, such as the Scale-Invariant Feature Transform (SIFT) [19]. The feature learning stage, the deep learning era since 2012, begins with artificial neural networks, particularly the breakthrough ImageNet and the Deep Convolutional Neural Network (DCNN) AlexNet [20]. Since then, deep learning has impacted a broad range of research areas, since DCNNs can learn powerful feature representations with multiple levels of abstraction directly from data. Deep learning techniques have attracted enormous attention and have brought about considerable breakthroughs in many computer vision tasks, including image classification [20], [21], [22], object detection [23], and image retrieval [10], [13], [14].

Excellent surveys for traditional image retrieval can be found in [1], [2], [8]. This paper, in contrast, focuses on deep learning based methods. A comparison of our work with other published surveys [8], [14], [15], [16] is shown in Table 1. Deep learning for image retrieval is comprised of the essential stages shown in Figure 1 and various methods, focusing on one or more stages, have been proposed to improve retrieval accuracy and efficiency. In this survey, we include comprehensive details about these methods, including feature fusion methods and network fine-tuning strategies *etc.* , motivated by the following questions that have been driving research in this domain:

1) *By using off-the-shelf models only, how do deep features outperform hand-crafted features?*
2) *In case of domain shifts across training datasets, how can we adapt off-the-shelf models to maintain or even improve retrieval performance?*
3) *Since deep features are generally high-dimensional, how can we effectively utilize them to perform efficient image retrieval, especially for large-scale datasets?*

Wei Chen, Erwin M. Bakker, Theodoros Georgiou, Michael S. Lew are with Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.
Yu Liu is with DUT-RU International School of Information Science and Engineering, Dalian University of Technology, China.
Weiping Wang is with College of Systems Engineering, NUDT, China.
Paul Fieguth is with the Systems Design Engineering Department, University of Waterloo, Canada.
Li Liu is with College of Systems Engineering, NUDT, China, and with Center for Machine Vision and Signal Analysis, University of Oulu, Finland.
Corresponding author: Li Liu, li.liu@oulu.fi

TABLE 1: A summary and comparison of the primary surveys in the field of image retrieval.

Title	Year	Published in	Main Content
Image Search from Thousands to Billions in 20 Years [12]	2013	TOMM	This paper gives a good presentation of image search achievements from 1970 to 2013, but the methods are not deep learning-based.
Deep Learning for Content-Based Image Retrieval: A Comprehensive Study [13]	2014	ACM MM	This paper introduces supervised metric learning methods for fine-tuning AlexNet. Details of instance-based image retrieval are limited.
Semantic Content-based Image Retrieval: A Comprehensive Study [14]	2015	JVCI	This paper presents a comprehensive study about CBIR using traditional methods; deep learning is introduced as a section with limited details.
Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement, and Retrieval [15]	2016	CSUR	A taxonomy is introduced to structure the growing literature of image retrieval. Deep learning methods for feature learning is introduced as future work.
Recent Advance in Content-based Image Retrieval: A Literature Survey [16]	2017	arXiv	This survey presents image retrieval from 2003 to 2016. Neural networks are introduced in a section and mainly discussed as a future direction.
Information Fusion in Content-based Image Retrieval: A Comprehensive Overview [17]	2017	Information Fusion	This paper presents information fusion strategies in content-based image retrieval. Deep convolutional networks for feature learning are introduced briefly but not covered thoroughly.
A Survey on Learning to Hash [18]	2018	T-PAMI	This paper focuses on hash learning algorithms and introduces the similarity-preserving methods and discusses their relationships.
SIFT Meets CNN: A Decade Survey of Instance Retrieval [8]	2018	T-PAMI	This paper presents a comprehensive review of instance retrieval based on SIFT and CNN methods.
Deep Image Retrieval: A Survey	2021	Ours	Our survey focuses on deep learning methods. We expand the review with in-depth details on CBIR, including structures of deep networks, types of deep features, feature enhancement strategies, and network fine-tuning.

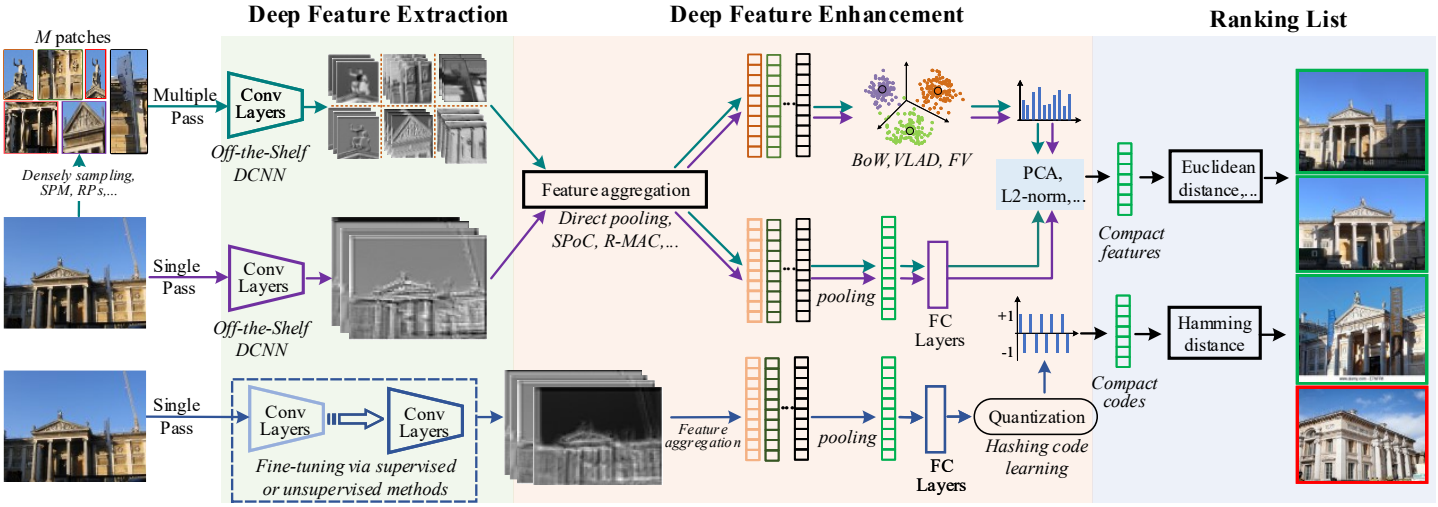


Fig. 1: In deep image retrieval, feature embedding and aggregation methods are used to enhance the discrimination of deep features. Similarity is measured on these enhanced features using Euclidean or Hamming distances.

1.1 Summary of Progress since 2012

After a highly successful image classification implementation based on AlexNet [20], significant exploration of DCNNs for retrieval tasks has been undertaken, broadly along the lines of the preceding three questions just identified, above. That is, the DCNN methods are divided into (1) off-the-shelf and (2) fine-tuned models, as shown in Figure 2, with parallel work on (3) effective features. Whether a DCNN is considered off-the-shelf or fine-tuned depends on whether the DCNN parameters are updated [24] or are based on DCNNs with fixed parameters [24], [25], [26]. Regarding how to use the features effectively, researchers have proposed encoding and aggregation methods, such as R-MAC [27], CroW [10], and SPoC [7].

Recent progress for improving image retrieval can be categorized into network-level and feature-level perspectives, for which a detailed sub-categorization is shown in Figure 3. The network-level perspective includes network architecture improvement and network fine-tuning strategies. The feature-level perspective includes feature extraction and feature enhancement methods. Broadly this survey will examine the four areas outlined as follows:

(1) *Improvements in Network Architectures* (Section 2)

Using stacked linear filters (*e.g.* convolution) and non-linear activation functions (ReLU, *etc.*), deep networks with different depths obtain features at different levels. Deeper networks with more layers provide a more powerful learning capacity so as to extract high-level abstract and semantic-aware features [21], [45]. It is also possible to concatenate multi-scale features in parallel, such as the Inception module in GoogLeNet [46], which we refer to as widening.

(2) *Deep Feature Extraction* (Section 3.1)

Neurons of FC layers and convolutional layers have different receptive fields, thus providing three ways to extract features: local features from convolutional layers [7], [27], global features from FC layers [31], [58] and fusions of two kinds of features [59], [60]; the fusion scheme includes layer-level and model-level methods. Deep features can be extracted from the whole image or from image patches, which corresponds to single pass and multiple pass feedforward schemes, respectively.

(3) *Deep Feature Enhancement* (Section 3.2)

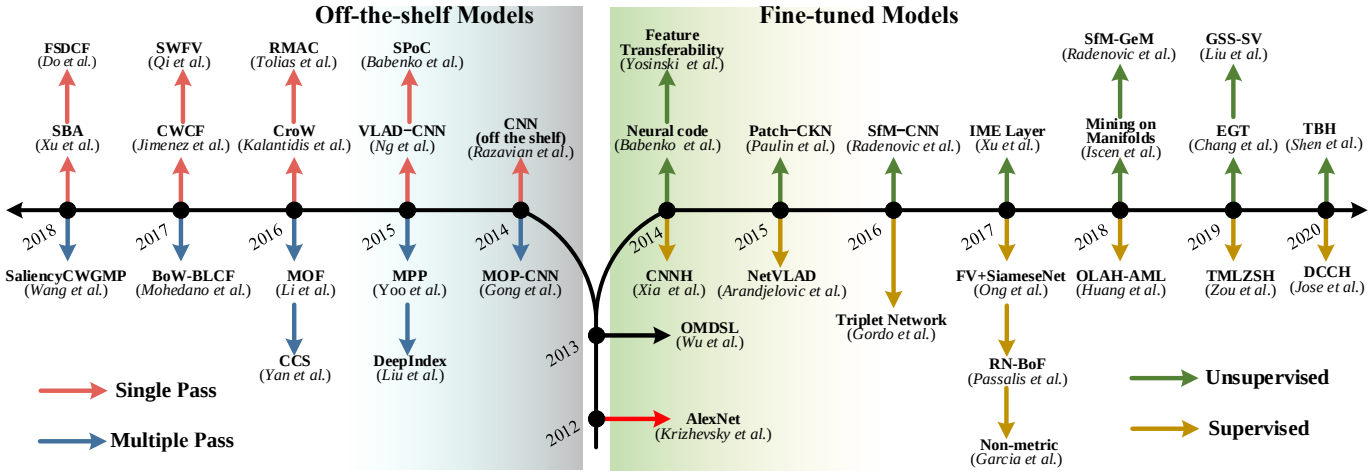


Fig. 2: Representative methods in deep image retrieval, which are most fundamentally categorized according to whether the DCNN parameters are updated [24]. Off-the-shelf models (left) have model parameters which are not further updated or tuned when extracting features for image retrieval. The relevant methods focus on improving representations quality either by feature enhancement [10], [28], [29], [30] when using single pass schemes or by extracting representations for image patches [31] when using multiple pass schemes. In contrast, in fine-tuned models (right) the model parameters are updated for the features to be fine-tuned towards the retrieval task and addresses the issue of domain shifts. The fine-tuning may be supervised [32], [33], [34], [35], [36], [37], [38] or unsupervised [39], [40], [41], [42], [43], [44]. See Sections 3 and 4 for details.

- Deep Learning for Image Retrieval
  - **Improvement in Deep Network Architectures (Section 2)**
    - Deepen Networks: AlexNet [20], VGG [45], ResNet [21], *etc.*
    - Widen Networks: GoogLeNet [46], DenseNet [22], *etc.*
  - Retrieval with Off-the-Shelf DCNN Models (Section 3)
    - **Deep Feature Extraction (Section 3.1)**
      - Network Feedforward Scheme (Section 3.1.1)
        - Single Feedforward Pass: MAC [47], R-MAC [27]
        - Multiple Feedforward Pass: SPM [31], RPNs [37]
      - Deep Feature Selection (Section 3.1.2)
        - Fully-connected Layer: Layer Concatenation [48]
        - Convolutional Layer: SPoC [7], CroW [10]
      - Feature Fusion Strategy (Section 3.1.3)
        - Layer-level Fusion: MoF [49], MOP [25]
        - Model-level Fusion: ConvNet fusion [45]
    - **Deep Feature Enhancement (Section 3.2)**
      - Feature Aggregation (Section 3.2.1)
      - Feature Embedding (Section 3.2.2)
      - Attention Mechanism (Section 3.2.3)
        - Non-parameteric: SPoC [7], TSWVF [50]
        - Parameteric: DeepFixNet+SAM [51], [52]
      - Deep Hash Embedding (Section 3.2.4)
        - Supervised Hashing: Metric Learning [34], [53]
        - Unsupervised Hashing: KNN [54], k-means [55]
  - **Retrieval via Learning DCNN Representations (Section 4)**
    - Supervised Fine-tuning (Section 4.1)
      - Classification-based Fine-tuning (Section 4.1.1)
      - Verification-based Fine-tuning (Section 4.1.2)
        - Transformation Matrix: Non-metric [35]
        - Siamese Networks: [36], [56]
        - Triplet Networks: [36], [56]
    - Unsupervised Fine-tuning (Section 4.2)
      - Manifold Learning Sample Mining: Diffusion Net [42]
      - AutoEncoder-based Fine-tuning: KNN [57], GANs [44]

Fig. 3: This survey is organized around four key aspects in deep image retrieval, shown in boldface.

Feature enhancement is used to improve the discriminative ability of deep features. Directly, aggregate features can be trained simultaneously with deep networks [17]; alternatively, feature embedding methods including BoW [61], VLAD [62], and FV [63] embed local features into global ones. These methods are trained with deep networks separately (codebook-based) or jointly (codebook-free). Further, hashing methods [18] encode the real-valued features into binary codes to improve retrieval efficiency. The feature enhancement strategy can significantly influence the efficiency of image retrieval.

(4) *Network Fine-tuning for Learning Representations* (Section 4)

Deep networks pre-trained on source datasets for image classification are transferred to new datasets for retrieval tasks. However, the retrieval performance is influenced by the domain shifts between the datasets. Therefore, it is necessary to fine-tune the deep networks to the specific domain [33], [55], [64], which can be realized by using supervised fine-tuning methods. However in most cases image labeling or annotation is time-consuming and difficult, so it is necessary to develop unsupervised methods for network fine-tuning.

1.2 Key Challenges

Deep learning has been successful in learning very powerful features. Nevertheless, several significant challenges remain with regards to

1) *reducing the semantic gap*,
2) *improving retrieval scalability*, and
3) *balancing retrieval accuracy and efficiency*.

We finish the introduction to this survey with a brief overview of each of these challenges:

1. Reducing the semantic gap: The semantic gap characterizes the difference, in any application, between the high-level concepts of humans and the low-level features typically derived from images [15]. There is significant interest in learning deep features which are higher-level and semantic-aware, to better preserve the similarities of images [15]. In the past few years,

various learning strategies, including feature fusion [25], [49] and feature enhancement methods [7], [27], [50] have been introduced into image retrieval. However, this area remains a major challenge and continues to require significant effort.

2. Improving retrieval scalability: The tremendous numbers and diversity of datasets lead to domain shifts for which existing retrieval systems may not be suited [8]. Currently available deep networks are initially trained for image classification tasks, which leads to a challenge in extracting features. Since such features are less scalable and perform comparatively poorly on the target retrieval datasets, so network fine-tuning on retrieval datasets is crucial for mitigating this challenge. The current dilemma is that the increase in retrieval datasets raises the difficulty of annotation, making the development of unsupervised fine-tuning methods a priority.

3. Balancing retrieval accuracy and efficiency: Deep features are usually high dimensional and contain more semantic-aware information to support higher accuracy, yet this higher accuracy is often at the expense of efficiency. Feature enhancement methods, like hash learning, are one approach to tackling this issue [18], [33], however hashing learning needs to carefully consider the loss function design, such as quantization loss [9], [11], to obtain optimal codes for high retrieval accuracy.

2 Popular Backbone DCNN Architectures

The hierarchical structure and extensive parameterization of DCNNs has led to their success in a remarkable diversity of computer vision tasks. For image retrieval, there are four models which predominantly serve as the networks for feature extraction, including AlexNet [20], VGG [45], GoogLeNet [46], and ResNet [21].

AlexNet is the first DCNN which improved ImageNet classification accuracy by a significant margin compared to conventional methods in ILSVRC 2012. It consists of 5 convolutional layers and 3 fully-connected layers. Input images are usually resized to a fixed size during training and testing stages.

Inspired by AlexNet, VGGNet has two widely used versions: VGG-16 and VGG-19, including 13 convolutional layers and 16 convolutional layers respectively, but where all of the convolutional filters are small (local), 3×3 in size. VGGNet is trained in a multi-scale manner where training images are cropped and re-scaled, which improves the feature invariance for the retrieval task.

Compared to AlexNet and VGGNet, GoogLeNet is deeper and wider but has fewer parameters within its 22 layers, leading to higher learning efficiency. GoogLeNet has repeatedly-used inception modules, each of which consists of four branches where 5×5, 3×3, and 1×1 filter sizes are used. These branches are concatenated spatially to obtain the final features for each module. It has been demonstrated that deeper architectures are beneficial for learning higher-level abstract features to mitigate the semantic gap [15].

Finally, ResNet is developed by adding more convolutional layers to extract more abstract features. Skip connections are added between convolutional layers to address the notorious vanishing gradient problem when training this network.

DCNN architectures have developed significantly during the past few years, for which we refer the reader to recent surveys [65], [66]. This paper focuses on introducing relevant techniques including feature fusion, feature enhancement, and network fine-tuning, based on popular DCNN backbones for performing image retrieval.

3 Retrieval with Off-the-Shelf DCNN Models

Because of their size, deep CNNs need to be trained on exceptionally large-scale datasets, and the available datasets of such size are those for image recognition and classification. One possible scheme then, is that deep models effectively trained for recognition and classification directly serve as the off-the-shelf feature detectors for the image retrieval task, the topic of interest in this survey. That is, one can propose to undertake image retrieval on the basis of DCNNs, trained for classification, and with their pre-trained parameters frozen.

There are limitations with this approach, such that the deep features may not outperform classical hand-crafted features. Most fundamentally, there is a model-transfer or domain-shift issue between tasks [8], [26], [67], meaning that models trained for classification do not necessarily extract features well suited to image retrieval. In particular, a classification decision can be made as long as the features remain within the classification boundaries, therefore the layers from such models may show insufficient capacity for retrieval tasks where feature matching is more important than the final classification probabilities. This section will survey the strategies which have been developed to improve the quality of feature representations, particularly based on feature extraction / fusion (Section 3.1) and feature enhancement (Section 3.2).

3.1 Deep Feature Extraction

3.1.1 Network Feedforward Scheme

a. Single Feedforward Pass Methods.

Single feedforward pass methods take the whole image and feed it into an off-the-shelf model to extract features. The approach is relatively efficient since the input image is fed only once. For these methods, both the fully-connected layer and last convolutional layer can be used as feature extractors [68].

The fully-connected layer has a global receptive field. After normalization and dimensionality reduction, these features are used for direct similarity measurement without further processing and admitting efficient search strategies [24], [25], [33].

Using the fully-connected layer lacks geometric invariance and spatial information, and thus the last convolutional layer can be examined instead. The research focus associated with the use of convolutional features is to improve their discrimination, where representative strategies are shown in Figure 4. For instance, one direction is to treat regions in feature maps as different sub-vectors, thus combinations of different sub-vectors of all feature maps are used to represent the input image.

b. Multiple Feedforward Pass Methods.

Compared to single-pass schemes, multiple pass methods are more time-consuming [8] because several patches are generated from an input image and are both fed into the network before being encoded as a final global feature.

Multiple-pass strategies can lead to higher retrieval accuracy since representations are produced from two stages: patch detection and patch description. Multi-scale image patches are obtained using sliding windows [25], [69] or spatial pyramid model [31], as illustrated in Figure 5. For example, Xu *et al*. [70] randomly sample windows within an image at different scales and positions, then "edgeness" scores are calculated to represent the edge density within the windows.

These patch detection methods lack retrieval efficiency for large-scale datasets since irrelevant patches are also fed

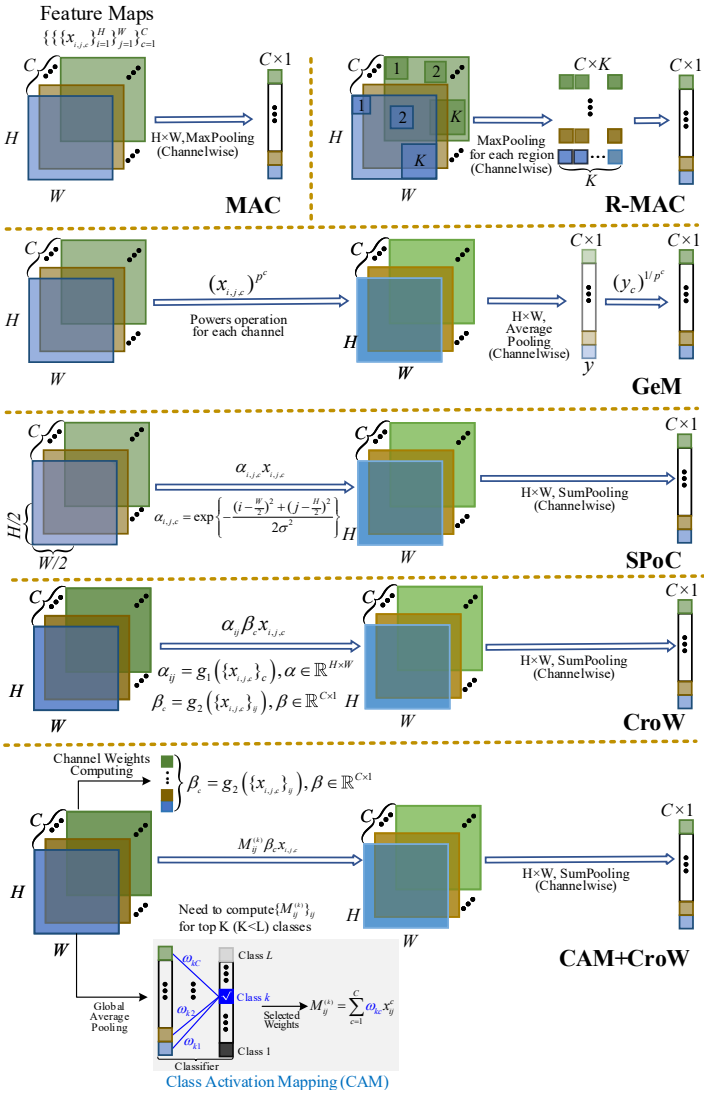


Fig. 4: Representative methods in single feedforward frameworks, focusing on convolutional feature maps x with size $H \times W \times C$: MAC [47], R-MAC [27], GeM pooling [41], SPoC with the Gaussian weighting scheme [7], CroW [10], and CAM+CroW [28]. Note that $g_1(\cdot)$ and $g_2(\cdot)$ represent spatial-wise and channel-wise weighting functions, respectively.

into deep networks, therefore it is necessary to analyze image patches [27]. As an example, Cao *et al*. [71] propose to merge image patches into larger regions with different hyper-parameters, then the hyper-parameter selection is viewed as an optimization problem under the target of maximizing the similarity between features of the query and the candidates.

Instead of generating multi-scale image patches randomly or densely, region proposal methods introduce a degree of purpose in processing image objects. Region proposals can be generated using object detectors, such as selective search [72] and edge boxes [73]. Aside from using object detectors, Region proposals can also be learned using deep networks, such as region proposal networks (RPNs) [23], [37] and convolutional kernel networks (CKNs) [74], and then to apply these deep networks into end-to-end fine-tuning scenarios for learning similarity [75], [76].

3.1.2 Deep Feature Selection

a. Extracted from Fully-connected Layers

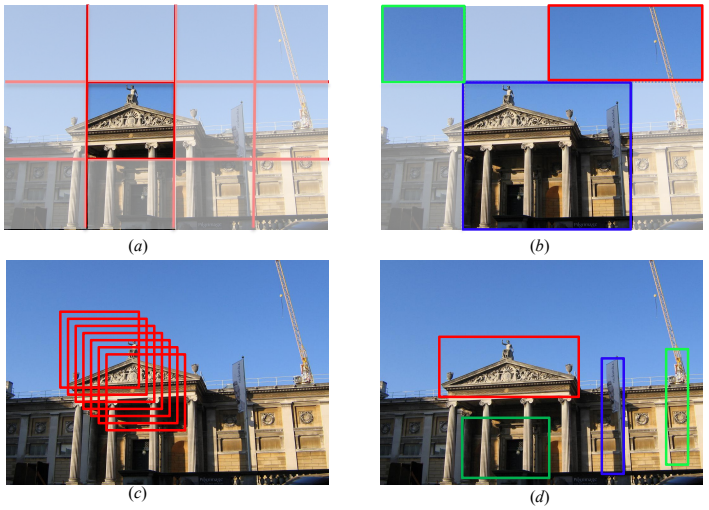


Fig. 5: Image patch generation schemes: (a) Rigid grid; (b) Spatial pyramid modeling (SPM); (c) Dense patch sampling; (d) Region proposals (RPs) from region proposal networks.

It is straightforward to select a fully-connected layer as a feature extractor [24], [25], [33], [48]. With PCA dimensionality reduction and normalization [24], images' similarity can be measured. Only the fully-connected layer may limit the overall retrieval accuracy, Jun *et al*. [48] concatenate features from multiple fully-connected layers, and Song *et al*. [75] indicate that making a direct connection between the first fully-connected layer and the last layer achieves coarse-to-fine improvements.

As noted, a fully-connected layer has a global receptive field in which each neuron has connections to all neurons of the previous layer. This property leads to two obvious limitations for image retrieval: a lack of spatial information and a lack of local geometric invariance [48].

For the first limitation, researchers focus on the inputs of networks, *i.e.*, using multiple feedforward passes [24]. Compared to taking as input the whole image, discriminative features from the image patches better retain spatial information.

For the second limitation, a lack of local geometric invariance affects the robustness to image transformations such as truncation and occlusion. For this, several works introduce methods to leverage intermediate convolutional layers [7], [25], [47], [77].

b. Extracted from Convolutional Layers

Features from convolutional layers (usually the last layer) preserve more structural details which are especially beneficial for instance-level retrieval [47]. The neurons in a convolutional layer are connected only to a local region of the input feature maps. The smaller receptive field ensures that the produced features preserve more local structural information and are more robust to image transformations like truncation and occlusion [7]. Usually, the robustness of convolutional features is improved after pooling.

A convolutional layer arranges the spatial information well and produces location-adaptive features [78], [79]. Various image retrieval methods use convolutional layers as local detectors [7], [27], [28], [47], [77], [79]. For instance, Razavian *et al*. [47] make the first attempt to perform spatial max pooling on the feature maps of an off-the-shelf DCNN model; Babenko *et al*. [7] propose sum-pooling convolutional features (SPoC) to obtain compact descriptors pre-processed with a Gaussian center prior (see Figure 4). Ng *et al*. [79] explore the correlations

between activations at different locations on the feature maps, thus improving the final feature descriptor. Kulkarni *et al*. [80] use the BoW model to embed convolutional features separately. Yue *et al*. [77] replace BoW [61] with VLAD [62], and are the first to encode local features into VLAD features. This idea inspired another milestone work [38] where, for the first time, VLAD is used as a layer plugged into the last convolutional layer. The plugged-in layer is end-to-end trainable via back-propagation.

3.1.3 Feature Fusion Strategy

a. Layer-level Fusion

Fusing features from different layers aims at combining different feature properties within a feature extractor. It is possible to fuse multiple fully-connected layers in a deep network [48]: For instance, Yu *et al*. [81] explore different methods to fuse the activations from different fully-connected layers and introduce the best-performed P_i-fusion strategy to aggregate the features with different balancing weights, and Jun *et al*. [48] construct multiple fully-connected layers in parallel on the top of ResNet backbone, then concatenate the global features from these layers to obtain the combined global features.

Features from fully-connected layers (global features) and features from convolutional layers (local features) can complement each other when measuring semantic similarity and can, to some extent, guarantee retrieval performance [82].

Global features and local features can be concatenated directly [82], [83], [84]. Before concatenation, convolutional feature maps are filtered by sliding windows or region proposal nets. Pooling-based methods can be applied for feature fusion as well. For example, Li *et al*. [49] propose a Multi-layer Orderless Fusion (MOF) approach, which is inspired by Multi-layer Orderless Pooling (MOP) [25] for image retrieval. However local features can not play a decisive role in distinguishing subtle feature differences because global and local features are treated identically. For this limitation, Yu *et al*. [82] propose using a mapping function to take more advantage of local features in which they are used to refine the return ranking lists. In their work, the exponential mapping function is the key for tapping the complementary strengths of the convolutional layers and fully-connected layers. Similarly, Cao *et al*. [84] unify the global and local descriptors for two-stage image retrieval in which attentively selected local features are employed to refine the results obtained using global features.

It is worth introducing a fusion scheme to explore *which* layer combination is better for fusion given their differences of extracting features. For instance, Chatfield *et al*. [60] demonstrate that fusing convolutional layers and fully-connected layers outperforms the methods that fuse only convolutional layers. In the end, fusing two convolutional layers with one fully-connected layer achieves the best performance.

b. Model-level Fusion

It is possible to combine features on different models; such fusion focuses on model complementarity to achieve improved performance, categorized into *intra-model* and *inter-model*.

Generally, intra-model fusion suggests multiple deep models having similar or highly compatible structures, while inter-model fusion involves models with more differing structures. For instance, the widely-used dropout strategy in AlexNet [20] can be regarded as intra-model fusion: with random connections of different neurons between two fully-connected layers, each training epoch can be viewed as the combinations of different models. As a second example, Simonyan *et al*. [45] introduce a ConvNet fusion strategy to improve the feature learning capacity of VGG where VGG-16 and VGG-19 are fused. This intra-model fusion strategy reduces the top-5 error by 2.7% in image classification compared to a single counterpart network. Similarly, Liu *et al*. [85] mix different VGG variants to strengthen the learning for fine-grained vehicle retrieval. Ding *et al*. [86] propose a selective deep ensemble framework to combine ResNet-26 and ResNet-50 improve the accuracy of fine-grained instance retrieval. To attend to different parts of the object in an image, Kim *et al*. [87] train an ensemble of three attention modules to learn features with different diversities. Each module is based on different Inception blocks in GoogLeNet.

Inter-model fusion is a way to bridge different features given the fact that different deep networks have different receptive fields [31], [52], [78], [88] [89], [90]. For instance, a two-stream attention network [52] is introduced to implement image retrieval where the mainstream network for semantic prediction is VGG-16 while the auxiliary stream network for predicting attention maps is DeepFixNet [91]. Similarly, considering the importance and necessity of inter-model fusion to bridge the gap between mid-level and high-level features, Liu *et al*. [31] and Zheng *et al*. [78] combine VGG-19 and AlexNet to learn combined features, while Ozaki *et al*. [89] make an ensemble to concatenate descriptors from six different models to boost retrieval performance. To illustrate the effect of different parameter choices within the model ensemble, Xuan *et al*. [90] combine ResNet and Inception V1 [46] for retrieval, concentrating on the embedding size and number of embedded features.

Inter-model and intra-model fusion are relevant to model selection. There are some strategies to determine *how* to combine the features from two models. It is straightforward to fuse all types of features from the candidate models and then learning a metric based on the concatenated features [52], which is a kind of "*early fusion*" strategy. Alternatively, it is also possible to learn optimal metrics separately for the features from each model, and then to uniformly combine these metrics for final retrieval ranking [32], which is a kind of "*late fusion*" strategy.

Discussion. Layer-level fusion and model-level fusion are conditioned on the fact that the involved components (layers or whole networks) have different feature description capacities. For these two fusion strategies, the key question is *what features are the best to be combined?* Some explorations have been made for answering this question based on off-the-shelf deep models. For example, Xuan *et al*. [90] illustrate the effect of combining different numbers of features and different sizes within the ensemble. Chen *et al*. [92] analyze the performance of embedded features from image classification and object detection models with respect to image retrieval. They study the discrimination of feature embeddings of different off-the-shelf models which, to some extent, implicitly guides the model selection when conducting the inter-model level fusion for feature learning.

3.2 Deep Feature Enhancement

3.2.1 Feature Aggregation

Feature enhancement methods aggregate or embed features to improve the discrimination of deep features. In terms of feature aggregation, sum/average pooling and max pooling are two commonly used methods applied on convolutional feature maps. In particular, sum/average pooling is less discriminative, because it takes into account all activated outputs from a convolutional layer, as a result it weakens the effect of highly

activated features [29]. On the contrary, max pooling is particularly well suited for sparse features that have a low probability of being active. Max pooling may be inferior to sum/average pooling if the output feature maps are no longer sparse [93].

Convolutional feature maps can be directly aggregated to produce global features by spatial pooling. For example, Razavian *et al*. [47], [69] apply max pooling on the convolutional features for retrieval. Babenko *et al*. [7] leverage sum pooling with a Gaussian weighting scheme to aggregate convolutional features (*i.e.* SPoC). Note that this operation usually is followed by L2 normalization and PCA dimensionality reduction.

As an alternative to the holistic approach, it is also possible to pool some regions in a feature map [7], [47], [78], such as done by R-MAC [27]. Also, it is shown that the pooling strategy used in the last convolutional layer usually yields superior accuracy over other shallower convolutional layers and even fully-connected layers [78].

3.2.2 Feature Embedding

Apart from direct pooling or regional pooling, it is possible to embed the convolutional feature maps into a high dimensional space to obtain compact features. The widely used embedding methods include BoW, VLAD, and FV. The embedded features' dimensionality can be reduced using PCA. Note that BoW and VLAD can be extended by using other metrics, such as Hamming distance [94]. Here we briefly describe the principle of the embedding methods for the case of Euclidean distance metric.

BoW [61] is a widely adopted encoding method. BoW encoding leads to a sparse vector of occurrence. Specifically, let $\vec{X} = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$ be a set of local features, each of which has dimensionality D. BoW requires a pre-defined codebook $\vec{C} = \{\vec{c}_1, \vec{c}_2, ..., \vec{c}_K\}$ with K centroids to cluster these local descriptors, and maps each descriptor $\vec{x}_t$ to the nearest word $\vec{c}_k$. For each centroid $\vec{c}_k$, one can count and normalize the number of occurrences by

$$g(\vec{c}_k) = \frac{1}{T}\sum_{t=1}^{T}\phi(\vec{x}_t, \vec{c}_k) \tag{1}$$

$$\phi(\vec{x}_t, \vec{c}_k) = \begin{cases} 1 & \text{if } \vec{c}_k \text{ is the closest codeword for } \vec{x}_t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Thus BoW considers the number of descriptors belonging to each codebook $\vec{c}_k$ (*i.e.* 0-order feature statistics), then BoW representation is the concatenation of all mapped vectors:

$$G_{_{BoW}}(\vec{X}) = \left[\ g(\vec{c}_1), \cdots, g(\vec{c}_k), \cdots, g(\vec{c}_K)\ \right]^\top \tag{3}$$

BoW representation is the histogram of the number of local descriptors assigned to each visual word, so that its dimension is equal to the number of centroids. This method is simple to implement to encode local descriptors, such as convolutional feature maps [49], [68], [80]. However, the embedded vectors are high dimensional and sparse, which are not well suited to large-scale datasets in terms of efficiency.

VLAD [62] stores the sum of residuals for each visual word. Specifically, similar to BoW, it generates K visual word centroids, then each feature $\vec{x}_t$ is assigned to its nearest visual centroid $\vec{c}_k$ and computes the difference $(\vec{x}_t - \vec{c}_k)$:

$$g(\vec{c}_k) = \frac{1}{T}\sum_{t=1}^{T}\phi(\vec{x}_t, \vec{c}_k)(\vec{x}_t - \vec{c}_k) \tag{4}$$

where $\phi(\vec{x}_t, \vec{c}_k)$ as defined in (2). Finally, the VLAD representation is stacked by the residuals for all centroids, with dimension $(D \times K)$, *i.e.*,

$$G_{_{VLAD}}(\vec{X}) = \left[\ \cdots, g(\vec{c}_k)^\top, \cdots\ \right]^\top. \tag{5}$$

VLAD captures first order feature statistics, *i.e.*, $(\vec{x}_t - \vec{c}_k)$. Similar to BoW, the performance of VLAD is affected by the number of clusters, thereby larger centroids produce larger vectors that are harder to index. For image retrieval, for the first time, Ng *et al*. [77] embed the feature maps from the last convolutional layer into VLAD representations, which is proved to have higher effectiveness than BoW.

The FV method [63] extends BoW by encoding the first and second order statistics continuously. FV clusters the set of local descriptors by a Gaussian Mixture Model (GMM), with K components, to generate a dictionary $C = \{\mu_k; \Sigma_k; w_k\}_{k=1}^{K}$, where w_k, μ_k, Σ_k denote the weight, mean vector, and covariance matrix of the k-th Gaussian component, respectively [95]. The covariance can be simplified by keeping only its diagonal elements, *i.e.*, $\sigma_k = \sqrt{diag(\Sigma_k)}$. For each local feature x_t, a GMM is given by

$$\gamma_k(\vec{x}_t) = w_k \times p_k(\vec{x}_t)/(\sum_{j=1}^{K} w_j p_j(x_t)) \quad \sum_{j=1}^{K} w_k = 1 \tag{6}$$

where $p_k(\vec{x}_t) = \mathcal{N}(\vec{x}_t, \mu_k, \sigma_k^2)$. All local features are assigned into each component k in the dictionary, which is computed as

$$\begin{aligned} g_{w_k} &= \frac{1}{T\sqrt{w_k}}\sum_{t=1}^{T}(\gamma_k(\vec{x}_t) - w_k) \\ g_{u_k} &= \frac{\gamma_k(\vec{x}_t)}{T\sqrt{w_k}}\sum_{t=1}^{T}\left(\frac{\vec{x}_t - \mu_k}{\sigma_k}\right), \\ g_{\sigma_k^2} &= \frac{\gamma_k(\vec{x}_t)}{T\sqrt{2w_k}}\sum_{t=1}^{T}\left[\left(\frac{\vec{x}_t - \mu_i}{\sigma_k}\right)^2 - 1\right] \end{aligned} \tag{7}$$

The FV representation is produced by concatenating vectors from the K components:

$$G_{_{FV}}(\vec{X}) = \left[\ g_{w_1}, \cdots, g_{w_K}, g_{u_1}, \cdots, g_{u_K}, g_{\sigma_1^2}, \cdots, g_{\sigma_K^2}\ \right]^\top \tag{8}$$

The FV representation defines a kernel from a generative process and captures more statistics than BoW and VLAD. FV vectors do not increase computational costs significantly but require more memory. Applying FV without memory controls may lead to suboptimal performance [96].

Discussion. Traditionally, sum pooling and max pooling are directly plugged into deep networks and the whole model is used in an end-to-end way, whereas the embedding methods, including BoW, VLAD, and FV, are initially trained separately with pre-defined vocabularies [31], [100]. For these three methods, one needs to pay attention to their properties before choosing one of them to embed deep features. For instance, BoW and VLAD are computed in the rigid Euclidean space where the performance is closely related to the number of centroids. The FV embedding method can capture higher order statistics than BoW or VLAD, thus the FV embedding improves the effectiveness of feature enhancement at the expense of a higher memory cost. Further, when any one of these methods is used, it is necessary to integrate them as a "layer" of deep networks so as to guarantee training and testing efficiency. For example, the VLAD method is integrated into deep networks where each spatial column feature is used to construct clusters via k-means [77]. This idea led to a follow-up approach,

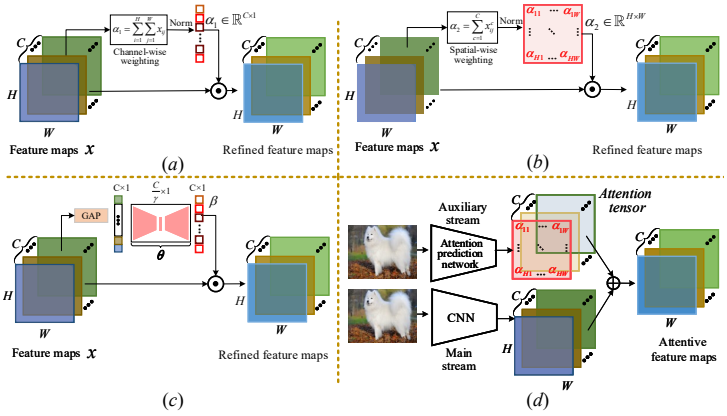


Fig. 6: Attention mechanisms are shown, divided into two categories. (a)-(b) Non-parametric mechanisms: The attention is based on convolutional feature maps x with size $H \times W \times C$. Channel-wise attention in (a) produces a C-dimensional importance vector α_1 [10], [30]. Spatial-wise attention in (b) computes a 2-dimensional attention map α_2 [10], [28], [59], [79]. (c)-(d) Parametric mechanisms: The attention weights β are provided by a sub-network with trainable parameters (*e.g.* θ in (c)) [97], [98]. Likewise, some off-the-shelf models [91], [99] can predict the attention maps from the input image directly.

NetVLAD [38], where deep networks are fine-tuned with the VLAD vector. The FV embedding method is also explored and combined with deep networks for retrieval tasks [36], [101].

3.2.3 Attention Mechanisms

The core idea of attention mechanisms is to highlight the most relevant features and to avoid the influence of irrelevant activations, realized by computing an attention map. Approaches to obtain attention maps can be categorized into two groups: non-parametric and parametric-based, as shown in Figure 6, where the main difference is whether the importance weights in the attention map are learnable.

Non-parametric weighting is a straightforward method to highlight feature importance. The corresponding attention maps can be obtained by channel-wise or spatial sum-pooling, as in Figure 6(a,b). For the spatial-wise pooling of Figure 6(b), Babenko *et al*. [7] apply a Gaussian center prior scheme to spatially weight the activations of a convolutional layer prior to aggregation. Kalantidis *et al*. [10] propose a more effective CroW method to weight and pool feature maps. These spatial-wise methods only concentrate on weighting activations at different spatial locations, without considering the relations between these activations. Instead, Ng *et al*. [79] explore the correlations among activations at different spatial locations on the convolutional feature maps. In addition to spatial-wise attention mechanisms, channel-wise weighting methods of Figure 6(a) are also popular non-parametric attention mechanisms. Xu *et al*. [30] rank the weighted feature maps to build the "probabilistic proposals" to further select regional features. Similarly, Jimenez *et al*. [28] combine CroW and R-MAC to propose Classes Activation Maps (CAM) to weight feature maps for each class. Qi *et al*. [50] introduce Truncated Spatial Weighted FV (TSWVF) to enhance the representation of Fisher Vector.

Attention maps can be learned from deep networks, as shown in Figure 6(c,d), where the input can be either image patches or feature maps from the previous convolutional layer. The parametric attention methods are more adaptive and are commonly used in supervised metric learning. For example, Li *et al*. [97] propose stacked fully-connected layers to learn an attention model for multi-scale image patches. Similarly, Noh *et al*. [98] design a 2-layer CNN with a softplus output layer to compute scores which indicate the importance of different image regions. Inspired by R-MAC, Kim *et al*. [102] employ a pre-trained ResNet101 to train a context-aware attention network using multi-scale feature maps.

Instead of using feature maps as inputs, a whole image can be used to learn feature importance, for which specific networks are needed. For example, Mohedano [51] explore different saliency models, including DeepFixNet [91] and Saliency Attentive Model (SAM) [99], to learn salient regions for input images. Similarly, Yang *et al*. [52] introduce a two-stream network for image retrieval in which the auxiliary stream, DeepFixNet, is used specifically for predicting attention maps.

In a nutshell, attention mechanisms offer deep networks the capacity to highlight the most important regions of a given image, widely used in computer vision. For image retrieval specifically, attention mechanisms can be combined with supervised metric learning [79], [87], [103].

3.2.4 Deep Hash Embedding

Real-valued features extracted by deep networks are typically high-dimensional, and therefore are not well-satisfied to retrieval efficiency. As a result, there is significant motivation to transform deep features into more compact codes. Hashing algorithms have been widely used for large-scale image search due to their computational and storage efficiency [18], [104].

Hash functions can be plugged as a layer into deep networks, so that hash codes can be trained and optimized with deep networks simultaneously. During hash function training, the hash codes of originally similar images are embedded as close as possible, and the hash codes of dissimilar images are as separated as possible. A hash function $h(\cdot)$ for binarizing features of an image x may be formulated as

$$b_k = h(x) = h(f(x;\boldsymbol{\theta})) \quad k = 1, \ldots, K \tag{9}$$

then an image can be represented by the generated hash codes $\boldsymbol{b} \in \{+1, -1\}^K$. Because hash codes are non-differentiable their optimization is difficult, so $h(\cdot)$ can be relaxed to be differentiable by using *tanh* or *sigmoid* functions [18].

When binarizing real-valued features, it is crucial (1) to preserve image similarity and (2) to improve hash code quality [18]. These two aspects are at the heart of hashing algorithms to maximize retrieval accuracy.

a. Hash Functions to Preserve Image Similarity

Preserving similarity seeks to minimize the inconsistencies between the real-valued features and corresponding hash codes, for which a variety of strategies have been adopted.

The design of loss function can significantly influence similarity preservation, which includes both supervised and unsupervised approaches. With the class label available, many loss functions are designed to learn hash codes in a Hamming space. As a straightforward method, one can optimize the difference between matrices computed from the binary codes and their supervision labels [105]. Other studies regularize hash codes with a center vector, for instance a class-specific center loss is devised to encourage hash codes of images to be close to the corresponding centers, reducing the intra-class variations [104]. Similarly, Kang *et al*. [106] introduce a max-margin t-distribution loss which concentrates more similar data into

a Hamming ball centered at the query term, such that a reduced penalization is applied to data points within the ball, a method which improves the robustness of hash codes when the supervision labels may be inaccurate. Moreover metric learning, including Siamese loss [53], triplet loss [34], [107], [108], and adversarial learning [107], [109], is used to retain semantic similarity where only dissimilar pairs keep their distance within a margin. In terms of unsupervised hashing learning, it is essential to capture some relevance among samples, which has been accomplished by using Bayes classifiers [110], KNN graphs [54], [57], k-means algorithms [55], and network structures such as AutoEncoders [111], [112], [113] and generative adversarial networks [44], [54], [114], [115].

Separate from the loss function, it is also important to design deep network frameworks for learning. For instance, Long *et al*. [108] apply unshared-weight CNNs on two datasets where a triplet loss and an adversarial loss are utilized to address the domain shifts. Considering the lack of label information, Cao *et al*. [109] present coined Pair Conditional WGAN, a new extension of Wasserstein generative adversarial networks (WGAN), to generate more samples conditioned on the similarity information.

b. Improving Hash Function Quality

Improving hash function quality aims at making the binary codes uniformly distributed, that is, maximally filling and using the hash code space, normally on the basis of bit uncorrelation and bit balance [18]. Bit uncorrelation implies that different bits are as independent as possible and have little redundancy of information, so that a given set of bits can aggregate more information within a given code length. In principle, bit uncorrelation can be formulated as $\boldsymbol{b}\boldsymbol{b}^\top = \boldsymbol{I}$ in which $\boldsymbol{I}$ is an identity matrix of size K. For example, it can be encouraged via regularization terms such as orthogonality [116] and mutual information [117]. Bit balance means that each bit should have a 50% chance of being +1 or -1, thereby maximizing code variance and information [18]. Mathematically, this condition is constrained by using this regularization term $\boldsymbol{b} \cdot \mathbf{1} = 0$ where $\mathbf{1}$ is a K-dimensional vector with all elements equal to 1.

4 Retrieval via Learning DCNN Representations

In Section 3, we presented feature fusion and enhancement strategies for which off-the-shelf DCNNs only serve as extractors to obtain features. However, in most cases, deep features may not be sufficient for high accuracy retrieval, even with the strategies which were discussed. In order for models to have higher scalability and to be more effective for retrieval, a common practice is network fine-tuning, *i.e*., updating the pre-stored parameters [26], [64]. However fine-tuning does not contradict or render irrelevant feature processing methods of Section 3; indeed, those strategies are complementary and can be incorporated as part of network fine-tuning.

This section focuses on supervised and unsupervised fine-tuning methods for the updating of network parameters.

4.1 Supervised Fine-tuning

4.1.1 Classification-based Fine-tuning

When class labels of a new dataset are available, it is preferable to begin with a previously-trained DCNN, trained on a separate dataset, with the backbone DCNN typically chosen from one of AlexNet, VGG, GoogLeNet, or ResNet. The DCNN can then be subsequently fine-tuned, as shown in Figure 7(a), by optimizing its parameters on the basis of a cross entropy loss L_{CE}:

$$L_{CE}(\hat{p}_i, y_i) = -\sum_{i}^{c}(y_i \times log(\hat{p}_i)) \tag{10}$$

Here y_i and $\hat{p}_i$ are the ground-truth labels and the predicted logits, respectively, and c is the total number of categories. The milestone work in such fine-tuning is [33], in which AlexNet is re-trained on the Landmarks dataset with 672 pre-defined categories. The fine-tuned network produces superior features on landmark-related datasets like Holidays [118], Oxford-5k, and Oxford-105k [119]. The newly-updated layers are used as global or local feature detectors for image retrieval.

A classification-based fine-tuning method improves the *model-level* adaptability for new datasets, which, to some extent, has mitigated the issue of model transfer for image retrieval. However, there still exists room to improve in terms of classification-based supervised learning. On the one hand, the fine-tuned networks are quite robust to inter-class variability, but may have some difficulties in learning discriminative intra-class variability to distinguish particular objects. On the other hand, class label annotation is time-consuming and labor-intensive for some practical applications. To this end, verification-based fine-tuning methods are combined with classification methods to further improve network capacity.

4.1.2 Verification-based Fine-tuning

With affinity information indicating similar and dissimilar pairs, verification-based fine-tuning methods learn an optimal metric which minimizes or maximizes the distance of pairs to validate and maintain their similarity. Compared to classification-based learning, verification-based learning focuses on both inter-class and intra-class samples. Verification-based learning involves two types of information [13]:

1) A pair-wise constraint, corresponding to a Siamese network as in Figure 7(c), in which input images are paired with either a positive or negative sample;
2) A triplet constraint, associated with triplet networks as in Figure 7(e), in which anchor images are paired with both similar and dissimilar samples [13].

These verification-based learning methods are categorized into globally supervised approaches (Figure 7(c,d)) and locally supervised approaches (Figure 7(g,h)), where the former learn a metric on global features by satisfying all constraints, whereas the latter focus on local areas by only satisfying the given local constraints (*e.g.* region proposals).

To be specific, consider a triplet set $X=\{(x_a, x_p, x_n)\}$ in a mini-batch, where (x_a, x_p) indicates a similar pair and (x_a, x_n) a dissimilar pair. Features $f(x;\boldsymbol{\theta})$ of one image are extracted by a network $f(\cdot)$ with parameters $\boldsymbol{\theta}$, for which we can represent the affinity information for each similar or dissimilar pair as

$$D_{ij} = D(x_i, x_j) = ||f(x_i;\boldsymbol{\theta}) - f(x_j;\boldsymbol{\theta})||_2^2 \tag{11}$$

a. Refining with Transformation Matrix.

Learning the similarity among the input samples can be implemented by optimizing the weights of a linear transformation matrix [35]. It transforms the concatenated feature pairs into a common latent space using a transformation matrix $\boldsymbol{W} \in \mathbb{R}^{2d\times 1}$, where d is the feature dimension. The similarity score of these pairs are predicted via a sub-network $S_W(x_i, x_j) =$

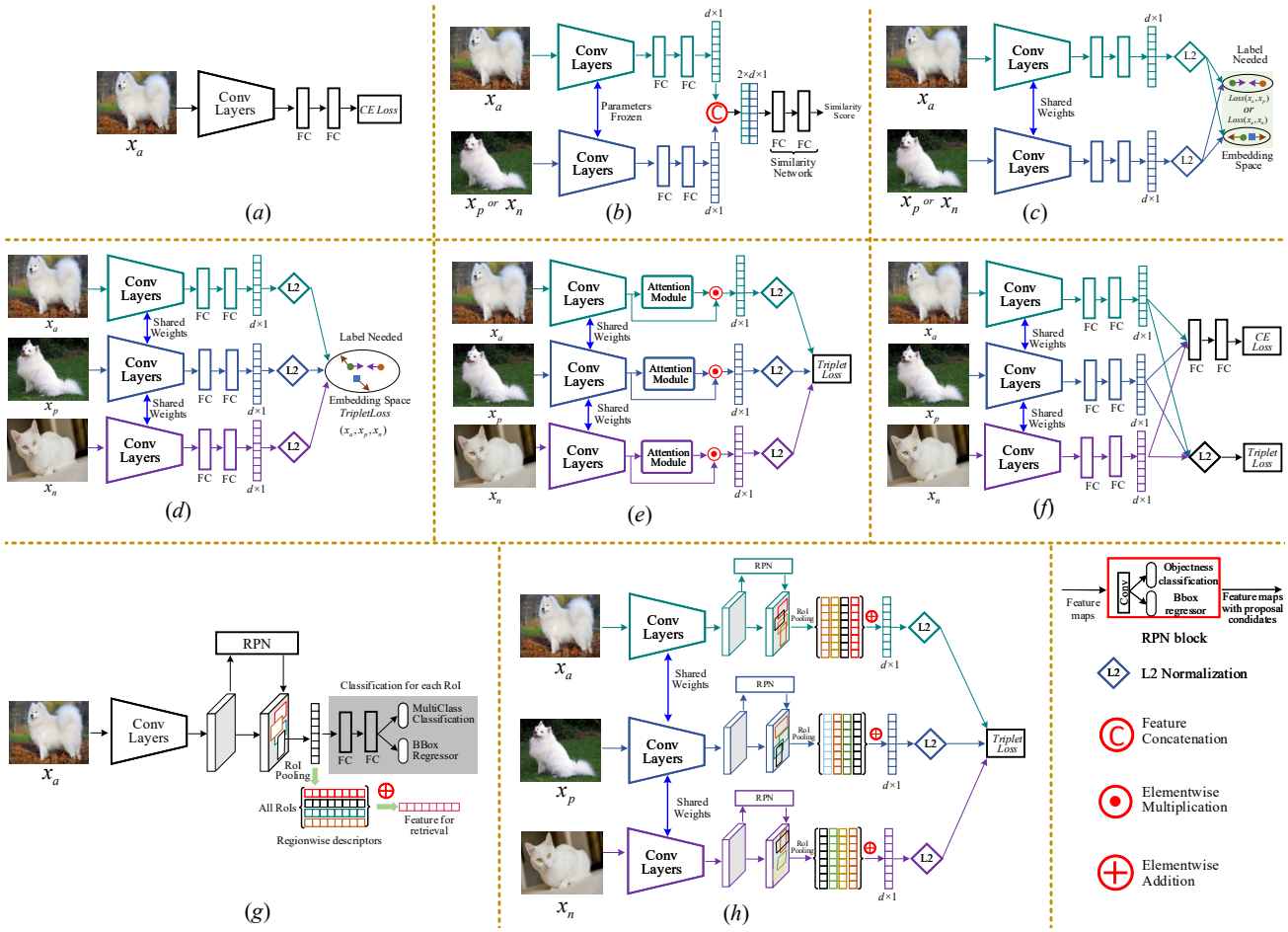


Fig. 7: Schemes of supervised fine-tuning. Anchor, positive, and negative images are indicated by x_a, x_p, x_n, respectively. (a) classification-based; (b) using a transformation matrix for learning the similarity of image pairs; (c) Siamese networks; (d) triplet loss for fine-tuning; (e) an attention block into DCNNs to highlight regions; (f) combining classification-based and verification-based loss for fine-tuning; (g) region proposal networks (RPNs) to locate the RoI and highlight specific regions or instances; (h) inserting the RPNs of (g) into DCNNs, such that the RPNs extract regions or instances at the convolutional layer.

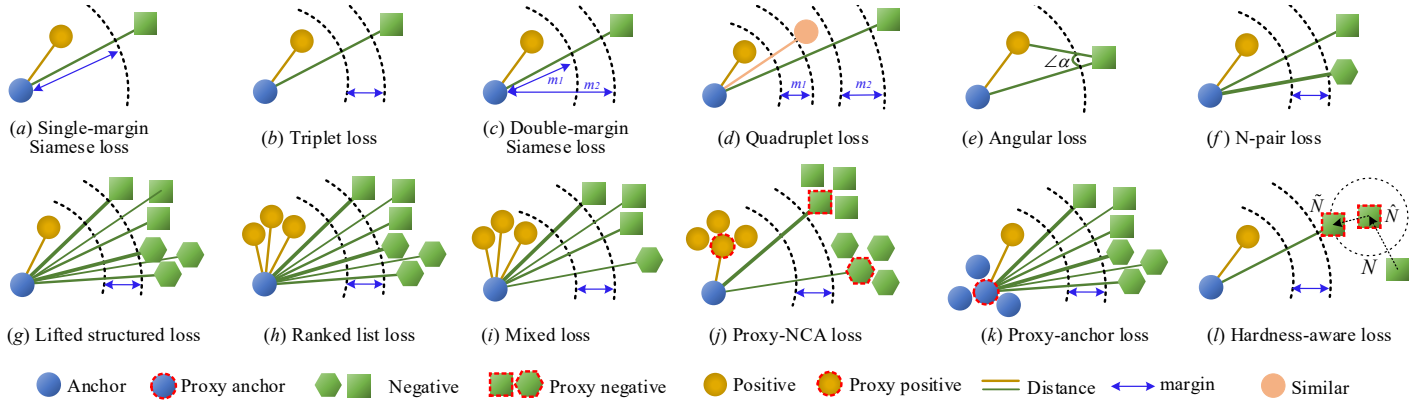


Fig. 8: Illustrations of sample mining strategies in metric learning. Here, we illustrate three classes, where shapes indicate different classes. Multiple pairs are considered in some loss terms and assigned with distinct weights during training, indicated by different line width. (a)-(c) have been introduced in the text. (d) Quadruplet loss [120]: a sample similar to the anchor is used to construct a double margin. (e) Angular loss [121]: the angle at the negative of triple triangles is computed to obtain higher order geometric constraints. (f) N-pair loss [122]: a positive sample is identified from $N-1$ negative samples of N-1 classes. (g) Lifted structured loss [123]: the structure relationships of three positive and three negative samples are considered. (h) Ranked list loss [124]: all samples to explore intrinsic structured information are considered. (i) Mixed loss [125]: three positive and three negative samples are captured which are initially closely distributed, where another anchor-negative pair initially lies very close to the anchor. (j) Proxy-NCA loss [126]: proxy positive and negative samples for each class are computed and trained with a true anchor sample. (k) Proxy-anchor loss [127]: the anchor sample is represented by a proxy. (l) Hardness-aware loss [128]: the synthetic negative is mapped from an existing hard negative, the hard levels manipulated adaptively within a certain range.

$f_W(f(x_i;\boldsymbol{\theta}) \cup f(x_j;\boldsymbol{\theta});\boldsymbol{W})$ [35], [129]. In other words, the sub-network f_W predicts how similar the feature pairs are. Given the affinity information of feature pairs $S_{ij} = S(x_i, x_j) \in \{0, 1\}$, the binary labels 0 and 1 indicate the similar (positive) or dissimilar (negative) pairs, respectively. The training of function f_W can be achieved by using a regression loss:

$$\begin{aligned} L_W(x_i, x_j) = &|S_W(x_i, x_j) - S_{ij}(sim(x_i, x_j) + m) - \\ &(1 - S_{ij})(sim(x_i, x_j) - m)| \end{aligned} \tag{12}$$

where $sim(x_i, x_j)$ can be the cosine function for guiding training $\boldsymbol{W}$ and m is a margin. By optimizing the regression loss and updating the transformation matrix $\boldsymbol{W}$, deep networks maximize the similarity of similar pairs and minimize that of dissimilar pairs. It is worth noting that the pre-stored parameters in the deep models are frozen when optimizing $\boldsymbol{W}$. The pipeline of this approach is depicted in Figure 7(b) where the weights of the two DCNNs are not necessarily shared.

b. Fine-tuning with Siamese Networks.

Siamese networks represent important options in implementing metric learning for fine-tuning, as shown in Figure 7(c). It is a structure composed of two branches that share the same weights across the layers. Siamese networks are trained on paired data, consisting of an image pair (x_i, x_j) such that $S(x_i, x_j) \in \{0, 1\}$. A Siamese loss function, illustrated in Figure 8(a), is formulated as

$$\begin{aligned} L_{Siam}(x_i, x_j) = &\frac{1}{2} S(x_i, x_j) D(x_i, x_j) + \\ &\frac{1}{2}(1 - S(x_i, x_j)) \max(0,\ m - D(x_i, x_j)) \end{aligned} \tag{13}$$

A standard Siamese network and Siamese loss are used to learn the similarity between semantically relevant samples under different scenarios. For example, Simo *et al*. [130] introduce a Siamese network to learn the similarity between paired image patches, which focuses more on the specific regions within an image. Ong *et al*. [36] leverage the Siamese network to learn image features which are then fed into the Fisher Vector model for further encoding. In addition, Siamese networks can also be applied to hashing learning in which the Euclidean distance formulation $D(\cdot)$ in Eq. 13 is replaced by the Hamming distance [53].

c. Fine-tuning with Triplet Networks.

Triplet networks [129] optimize similar and dissimilar pairs simultaneously. As shown in Figure 7(d) and Figure 8(b), the plain triplet networks adopt a ranking loss for training:

$$L_{Triplet}(x_a, x_p, x_n) = \max(0, m + D(x_a, x_p) - D(x_a, x_n))) \tag{14}$$

which indicates that the distance of an anchor-negative pair $D(x_a, x_n)$ should be larger than that of an anchor-positive pair $D(x_a, x_p)$ by a certain margin m. The triplet loss is used to learn fine-grained image features [56], [88] and for constraining hash code learning [34], [107], [108].

To focus on specific regions or objects, local supervised metric learning has been explored [42], [76], [131], [132]. In these methods, some regions or objects are extracted using region proposal networks (RPNs) [23] which subsequently can be plugged into deep networks and trained in an end-to-end manner, such as shown in Figure 7(g), in which Faster R-CNN [23] is fine-tuned for instance search [76]. RPNs yield the regressed bounding box coordinates of objects and are trained by the multi-class classification loss. The final networks extract better regional features by RoI pooling and perform spatial ranking for instance retrieval.

RPNs [23] enable deep models to learn regional features for particular instances or objects [37], [132]. RPNs used in the triplet formulation are shown in Figure 7(h). For training, besides the triplet loss, regression loss (PRNs loss) is used to minimize the regressed bounding box according to ground-truth region of interest. In some cases, jointly training an RPN loss and triplet loss leads to unstable results. This is addressed in [37] by first training a CNN to produce R-MAC using a rigid grid, after which the parameters in convolutional layers are fixed and RPNs are trained to replace the rigid grid.

Attention mechanisms can also be combined with metric learning for fine-tuning [103], [131], as in Figure 7(e), where the attention module is typically end-to-end trainable and takes as input the convolutional feature maps. For instance, Song *et al*. [131] introduce a convolutional attention layer to explore spatial-semantic information, highlighting regions in images to significantly improve the discrimination for inter-class and intra-class features for image retrieval.

Recent studies [48], [83] have jointly optimized the triplet loss and classification loss function, as shown in Figure 7(f). Fine-tuned models that use only a triplet constraint may possess inferior classification accuracy for similar instances [83], since the classification loss does not predict the intra-class similarity, rather locates the relevant images at different levels. Given these considerations, it is natural to combine and optimize triplet constraint and classification loss jointly [48]. The overall joint function is formulated as

$$L_{Joint} = \alpha \cdot L_{Triplet}(x_{i,a}, x_{i,p}, x_{i,n}) + \beta \cdot L_{CE}(\hat{p}_i, y_i) \tag{15}$$

where the cross-entropy loss (CE loss) L_{CE} is defined in Eq. (10) and the triplet loss $L_{Triplet}$ in Eq. (14). α and β are trade-off hyper-parameters to tune the two loss functions.

An implicit drawback of the Siamese loss in Eq. 13 is that it may penalize similar image pairs even if the margin between these pairs is small or zero, which may degrade performance [133], since the constraint is too strong and unbalanced. At the same time, it is hard to map the features of similar pairs to the same point when images contain complex contents or scenes. To tackle this limitation, Cao *et al*. [134] adopt a double-margin Siamese loss [133], illustrated in Figure 8(c), to relax the penalty for similar pairs. Specifically, the threshold between the similar pairs is set to a margin m_1 instead of being zero. In this case, the original single-margin Siamese loss is re-formulated as

$$\begin{aligned} L(x_i, x_j) = &\frac{1}{2} S(x_i, x_j) \max(0, D(x_i, x_j) - m_1) + \\ &\frac{1}{2}(1 - S(x_i, x_j)) \max(0, m_2 - D(x_i, x_j)) \end{aligned} \tag{16}$$

where $m_1 > 0$ and $m_2 > 0$ are the margins affecting the similar and dissimilar pairs, respectively. Therefore, the double margin Siamese loss only applies a contrastive force when the distance of a similar pair is larger than m_1. The mAP metric of retrieval is improved when using the double margin Siamese loss [133].

Discussion. Most verification-based supervised learning methods rely on the basic Siamese or triplet networks. The follow-up studies are focusing on exploring methods to further improve their capacities for robust feature similarity estimation. Generally, the network structure, loss function, and sample selection are important factors for the success of verification-based methods.

A variety of loss functions have been proposed recently [120], [122], [123], [124], [126]. Some of these use more samples

or additional constraints. For example, Chen *et al*. [120] incorporate Quadruplet samples for constraining relationships between anchor, positive, negative, and similar images. The N-pair loss [122] and the lifted structured loss [123] even define constraints on all images and employ the structural information of samples in a mini-batch.

The sampling strategy can greatly affect the feature learning and training convergence rate. To date, many sampling strategies such as clustering have been introduced, of which 12 are illustrated in Figure 8. Aside from sampling within a mini-batch, other work explores mining samples outside a mini-batch even from the whole dataset. This may be beneficial for stabilizing optimization due to a larger data diversity and richer training information. For example, Wang *et al*. [135] propose a cross-batch memory (XBM) mechanism that memorizes the embedding of past iterations, allowing the model to collect sufficient hard negative pairs across multiple mini-batches. Harwood *et al*. [136] provide a framework named smart mining to collect hard samples from the entire training set. It is reasonable to achieve better performance when more samples are used to fine-tune a network. However, the possible additional computational cost during training is a core issue to be addressed.

Directly optimizing the average precision (AP) metric using the listwise AP loss [137] is one way to consider a large number of image simultaneously. Training with this loss has been demonstrated to improve retrieval performance [137], [138], [139], however average precision, as a metric, is normally non-differentiable and non-smooth. To directly optimize the AP loss, the AP metric needs to be relaxed by using methods such as soft-binning approximation [137], [138] or sigmoid function [139].

4.2 Unsupervised Fine-tuning

Supervised network fine-tuning becomes infeasible when there is not enough supervisory information because such information is costly to assemble or unavailable. Given these limitations, unsupervised fine-tuning methods for image retrieval are quite necessary but less studied [140].

For unsupervised fine-tuning, two broad directions are to mine relevance among features via manifold learning to obtain ranking information, and to devise novel unsupervised frameworks (*e.g.* AutoEncoders), each discussed below.

4.2.1 Mining Samples with Manifold Learning

Manifold learning focuses on capturing intrinsic correlations on the manifold structure to mine or deduce revelance, as illsutrated in Figure 9. Initial similarities between the original extracted features are used to construct an affinity matrix, which is then re-evaluated and updated using manifold learning [141]. According to the manifold similarity in the updated affinity matrix, positive and hard negative samples are selected for metric learning using verification-based loss functions such as pair loss [42], [142], triplet loss [143], [144], or N-pair loss [140], *etc*. Note that this is different from the aforementioned methods for verification-based fine-tuning methods, where the hard positive and negative samples are explicitly selected from an ordered dataset according to the given affinity information.

It is important to capture the geometry of the manifold of deep features, generally involving two steps [141] known as a diffusion process. First, the affinity matrix (Figure 9) is interpreted as a weighted kNN graph, where each vector is represented by a node, and edges are defined by the pairwise

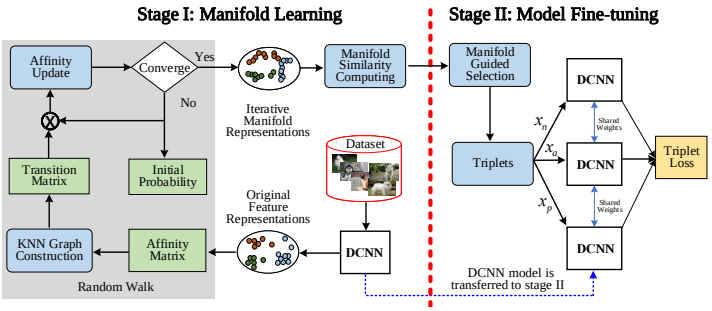


Fig. 9: Paradigm of manifold learning for unsupervised metric learning, based on triplet loss.

affinities of two connected nodes. Then, the pairwise affinities are re-evaluated in the context of all other elements by diffusing the similarity values through the graph [43], [142], [143], [144]. Some new similarity diffusion methods have recently been proposed, like the regularized diffusion process (RDP) [145] and the regional diffusion mechanism [142]. For more details on diffusion methods we refer to the survey [141].

Most existing algorithms follow a similar principle (*e.g.* random walk [141]). The differences among methods lie primarily in three aspects:

1) **Similarity initialization**, which affects the subsequent KNN graph construction in an affinity matrix. Usually, an inner product [43], [140] or Euclidean distance [40] is directly computed for the affinities. A Guassian kernel function can be used for affinity initialization [141], [144], or Iscen *et al*. [142] consider regional similarity from image patches to build the affinity matrix.
2) **Transition matrix definition**, a row-stochastic matrix [141], determines the probabilities of transiting from one node to another in the graph. These probabilities are proportional to the affinities between nodes, which can be measured by Geodesic distance (*e.g.* the summation of weights of relevant edges).
3) **Iteration scheme**, to re-valuate and update the values in affinity matrix by the manifold similarity until some kind of convergence is achieved. Most algorithms are iteration-based [141], [143], as illustrated in Figure 9.

Diffusion process algorithms are indispensable for unsupervised fine-tuning. Better image similarity is guaranteed when it is improved based on initialization (*e.g.* regional similarity [142] or high order information [40]). However, the diffusion process requires more computation and searching due to the iteration scheme [144], a limitation which cannot meet the efficiency requirements of image retrieval. To mitigate this, Nicolas *et al*. [140] apply the closed-form convergence solution of a random walk in each mini-batch to estimate the manifold similarities instead of running many iterations. Some studies replace the diffusion process on a kNN graph with a diffusion network [42], which is derived from graph convolution networks [146]. Their end-to-end framework allows efficient computation during the training and testing stages.

Once the manifold space is learned, samples are mined by computing geodesic distances based on the Floyd-Warshall algorithm or by comparing the set difference [143]. The selected samples are fed into deep networks to perform fine-tuning.

It is possible to explore proximity information, to cluster in Euclidean space, splitting the training set into different groups. For example, Tzelepi *et al*. [147] explore a fully unsupervised

fine-tuning method by clustering, in which the kNN algorithm is used to compute the k nearest features, then fine-tuned to minimize the squared distance between each query feature and its k nearest features. As a second example, Radenovic *et al*. [39], [41] use Structure-from-Motion (SfM) for clustering to explore sample reconstructions to select images for triplet loss. Clustering methods depend on the Euclidean distance, making it difficult to reveal the intrinsic relationship between objects.

4.2.2 AutoEncoder-based Frameworks

An AutoEncoder is a kind of neural network that aims to reconstruct its output as closely as possible to its input. In principle, an input image is encoded as features into a latent space, and these features are then reconstructed to the original input image using a decoder. The encoder and decoder can be both be convolutional neural networks.

In an AutoEncoder, there exist different levels (*e.g.* pixel-level or instance-level) of reconstruction. These different reconstructions affect the effectiveness of an AutoEncoder, in that pixel-level reconstructions may degrade the learned features of an encoder by focusing on trivial variations in a reconstructed image, since natural images typically contains many detailed factors of location, color, and pose.

An AutoEncoder is an optional framework for supporting other methods, for example the implementation of unsupervised hash learning [44], [111], [112], [113]. Except for the reconstruction loss [44], [113], it is highly necessary to mine feature relevance to explore other objective functions. This is usually realized by using clustering algorithms [113] since features from an off-the-shelf network initially contain rich semantic information to keep their semantic structure [54], [57], [110]. For example, Gu *et al*. [113] introduce a modified cross-entropy based on the k-means clustering algorithm where a deep model learns to cluster iteratively and yields binary codes while retaining the structures of the input data distributions. Zhou *et al*. [57] and Deng *et al*. [54] propose a self-taught hashing algorithm using a kNN graph construction to generate pseudo labels that are used to analyze and guide network training. Other techniques such as Bayes Nets are also used to predict sample similarity, such as in the work of Yang *et al*. [110], which adopts a Bayes optimal classifier to assign semantic similarity labels to data pairs which have a higher similarity probability.

AutoEncoders can also be integrated into other frameworks, such as graph convolutional networks [146] and object detection models [148] to learn better binary latent variables. For example, Shen *et al*. [44] combine graph convolutional networks [146] to learn the hash codes from an AutoEncoder. In this method, the similarity matrix for graph learning is computed on the binary latent variables from the Encoder. Generative adversarial networks (GANs) are also explored in the unsupervised hashing framework [44], [54], [114], [115]. The adversarial loss in GANs is the classical objective to use. By optimizing this loss, the synthesized images generated from hash codes gradually keep semantic similarity consistent for the original images. The pixel-level and feature-level content loss are used to improve the generated image quality [114]. Some other losses are employed in GANs to enhance hash code learning. For instance, a distance matching regularizer is utilized to propagate the correlations between high-dimensional real-valued features and low-dimensional hash codes [149], or two loss functions that aim at promoting independence of binary codes [115]. In summary, using GANs for unsupervised hash learning is promising, but there remains much room for further exploration.

5 State of the Art Performance

5.1 Datasets

To demonstrate the effectiveness of methods, we choose four commonly-used datasets for performance comparison: Holidays, Oxford-5k (including the extended Oxford-105k), Paris-6k (including the extended Paris-106k) and UKBench.

UKBench (UKB) [150] consists of 10,200 images of objects. The whole dataset has 2,550 groups of images, each group having four images of the same object from different viewpoints or illumination conditions. Each image in the dataset can be used as a query image.

Holidays [118] consists of 1,491 images collected from personal holiday albums. Most images are scene-related. The dataset comprises 500 groups of similar images with a query image for each group. In each group, the first image is used as a query image for performance evaluation.

Oxford-5k [119] consists of 5,062 images for 11 Oxford buildings. Each image is represented by five queries by a hand-drawn bounding box, thus there are 55 query Regions of Interest (RoI) in total. An additional disjoint set of 100,000 distractor images is added to obtain Oxford-100k.

Paris-6k [151] includes 6,412 images collected from Flickr. It is categorized into 12 groups about specific Paris architectures. The dataset has 500 query images for evaluation, and 55 queries with bounding boxes. Images are annotated with the same four types of labels as used in the Oxford-5k dataset.

Annotations and evaluation protocols in Oxford-5k and Paris-6k are updated; additional queries and distractor images are added into the two datasets, producing the *Revisited Oxford* and *Revisited Paris* datasets [152]. Due to the popularity of Oxford-5k and Paris-6k, we primarily undertake performance evaluations on the original datasets.

5.2 Evaluation Metrics

Average precision (AP) refers to the coverage area under the precision-recall curve. A larger AP implies a higher precision-recall curve and better retrieval accuracy. AP can be calculated as

$$AP = \frac{\sum_{k=1}^{N} P(k) \cdot rel(k)}{R} \tag{17}$$

where R denotes the number of relevant results for the query image from the total number N of images. $P(k)$ is the precision of the top k retrieved images, and $rel(k)$ is an indicator function equal to 1 if the item within rank k is a relevant image and 0 otherwise. Mean average precision (mAP) is adopted for the evaluation over all query images,

$$\frac{1}{Q}\sum_{q=1}^{Q} AP(q) \tag{18}$$

where Q is the number of query images.

Additionally, **N-S score** is a metric used for UKBench [150]. In this dataset, there are four relevant images for each query. The N-S score is the average, four times, for the top-four precision over the dataset.

5.3 Performance Comparison and Analysis

Overview. We conclude with the performance over these 4 datasets from 2014 to 2020 in Figure 10(a). At early period, DCNNs acted as powerful extractors and achieved good results, *e.g.*, mAP is 78.34% in [13] on Oxford-5k. Subsequently, the results increased significantly when some crucial factors

were adopted, including feature fusion [153], [154], [155], feature aggregation [27], [47], and network fine-tuning [145], [156]. For instance, the accuracy on UKBench reaches an mAP of 98.8% in [155] when an undirected graph is defined to fuse features and estimate their correlations. Network fine-tuning improves performance greatly. The accuracy increases steadily from 78.34% [13] to 96.2% [157] on the Oxford-5k dataset when manifold learning is used to fine-tune deep networks.

We report the results of methods using off-the-shelf models (Table 3) and fine-tuning networks (Table 4). In Table 3, single pass and multiple pass are analyzed, while supervised fine-tuning and unsupervised fine-tuning are compared in Table 4.

Evaluation for single feedforward pass. The common practice using this scheme is to enhance feature discrimination. In Table 3, we observe that fully-connected layers used as feature extractors may reach a lower accuracy (*e.g.*, 74.7% on Holidays in [33]), compared to the counterpart convolutional layers because the fully-connected layers lack structural information. Layer-level feature fusion strategy improves retrieval accuracy. For example, Yu *et al*. [82] combined three layers (*Conv4*, *Conv5*, and *FC6*) (*e.g.*, an mAP of 91.4% on Holidays), outperforming the performance of non-fusion method in [7] (*e.g.*, mAP is 80.2%). Moreover, convolutional features embedded by BoW model reach a competitive performance on Oxford-5k and Paris-6k (73.9% and 82.0%, respectively), while its codebook size is 25k, which may affect the retrieval efficiency. For single pass scheme, methods shown in Figure 4 improve the discrimination of convolutional feature maps and perform differently in Table 3 (*e.g.*, 66.9% of R-MAC [151], 58.9% of SPoC [7] on Oxford-5k). We view this as a critical factors and further analyze.

Evaluation for multiple feedforward pass. The methods exemplified in Figure 5 are reported their results in multiple pass scheme. Among them, extracting image patches densely using Overfeat [158] can reach best results on the 4 datasets [24]. Using rigid grid method reach competitive results (*e.g.*, an mAP of 87.2% on Paris-6k) [100]. These two methods consider more patches, even background information when used for feature extraction. Instead of generating patches densely, region proposals and spatial pyramid modeling have a degree of purpose in processing image objects. This may be more efficient and less memory demanding. Using multiple-pass scheme, spatial information is maintained better than the case using the single-pass method. For example, a shallower network (AlexNet) and region proposal networks are used in [72], its result on UKBench is 3.81 (N-Score), higher than the one using deeper networks, such as [7], [33], [82]. Besides feeding image patches into the same network, model-level fusion also exploit complementary spatial information to improve the retrieval accuracy. For instance, as reported in [31], which combines AlexNet and VGG, the results on Holidays (81.74% of mAP) and UKBench (3.32 of N-Score) are better than these in [49] (76.75% and 3.00, respectively).

Evaluation for supervised fine-tuning. Compared to the off-the-shelf models, fine-tuning deep networks usually improves accuracy, see Table 4. For instance, the result on Oxford-5k [27] by using a pre-trained VGG is improved from 66.9% to 81.5% in [36] when a single-margin Siamese loss is used. Similar trends can be also observed on the Paris-6k dataset. Although classification-based fine-tuning method is not excel at learning intra-class variability (*e.g.*, an mAP of 55.7% on Oxford-5k in [33]), its performance may be improved with powerful DCNNs and feature enhancement methods

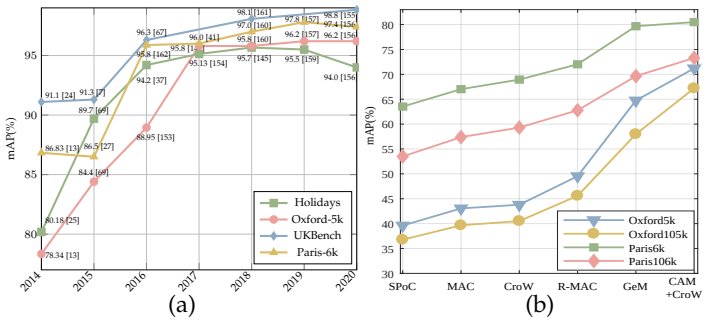


Fig. 10: (a) Performance improvement from 2014 to 2020. (b) mAP comparison of the feature aggregation methods shown in Figure 4.

such as the attention mechanism in [98], leading to an mAP of 83.8% on Oxford-5k. As for verification-based fine-tuning methods, in some cases, the loss used for fine-tuning is essential for performance improvement. For example, RPN is re-trained using regression loss on Oxford-5k and Paris-6k (75.1% and 80.7%, respectively) [76]. Its results are lower than the results from [35] (88.2% and 88.2%, respectively) where a transformation matrix is used to learn visual similarity. However, when RPN is trained by using triplet loss such as [132], the effectiveness of retrieval is improved significantly where the results are 86.1% (on Oxford-5k) and 94.5% (on Paris-6k). Further, feature embedding methods are important for retrieval accuracy. For example, Ong *et al*. [36] embedded *Conv5* feature maps by Fisher Vector and achieved an mAP of 81.5% on Oxford-5k, while embedding feature maps by using VLAD achieves an mAP of 62.5% on this dataset [38], [39].

Evaluation for unsupervised fine-tuning. Compared to supervised fine-tuning, unsupervised fine-tuning methods are relatively less explored. The difficulty for unsupervised fine-tuning is to mine relevance of samples without ground-truth labels. In general, unsupervised fine-tuning methods produce lower performance than the supervised fine-tuning methods. For instance, supervised fine-tuning network by using Siamese loss in [163] achieves an mAP 88.4% on Holidays, while unsupervised fine-tuning network using the same loss function in [39], [41], [143] achieve 82.5%, 83.1%, and 87.5%, respectively. However, unsupervised fine-tuning methods can achieve a similar accuracy even outperform the supervised fine-tuning if a suited feature embedding method is used. For instance, Zhao *et al*. [144] explore global feature structure with modeling the manifold learning, producing an mAP of 85.4% (on Oxford-5k) and 96.3% (on Paris-6k). This is similar to the supervised method [132], whose results are 86.1% (on Oxford-5k) and 94.5% (on Paris-6k). As another example, the precision of ResNet-101 fine-tuned by cross-entropy loss achieves to 83.8% on Oxford-5k [98], while the precision is further improved to 92.0% when IME layer is used to embed features and fine-tuned in an unsupervised way [40]. Note that fine-tuning strategies are related to the type of the target retrieval datasets. As demonstrated in [101], fine-tuning on different datasets may hurt the final performance.

Retrieval efficiency is also an important criterion in deep image retrieval. Deep learning methods are usually based on large-size datasets. The training and testing of retrieval methods are mostly done on GPUs. Most prior works focus more on retrieval accuracy but less on efficiency. We report the retrieval accuracy and retrieval efficiency on the 4 datasets in

TABLE 2: Evaluations of mAP (%), N-S score, and average search time per image. "†" refers to the query time is evaluated in a global diffusion manner, while "†" refers to the time is evaluated in a regional diffusion way.

	Oxford-5k (+100k)		Paris-6k (+100k)		Holidays		UKB	
	mAP	Time	mAP	Time	mAP	Time	N-S	Time
[145]	91.3 (88.4)	5.45 *ms* (809 *ms*)	-	-	95.66	3.11 *ms*	3.93	4.91 *ms*
[157]	92.6 (91.8)	2 *ms* (10 *ms*)	-	-	-	-	-	-
[142]†	85.7 (-)	20 *ms* (-)	94.1 (-)	20 *ms* (-)	-	-	-	-
[142]‡	95.8 (-)	600 *ms* (-)	96.9 (-)	700 *ms* (-)	-	-	-	-
[164]	64.9 (58.8)	0.81 *ms* (0.82 *ms*)	-	-	-	-	-	-
[41]	64.8 (57.9)	0.77 *ms* (0.73 *ms*)	-	-	-	-	-	-
[35]	55.5 (-)	0.35 *ms* (-)	71.0 (-)	0.35 *ms* (-)	-	-	-	-

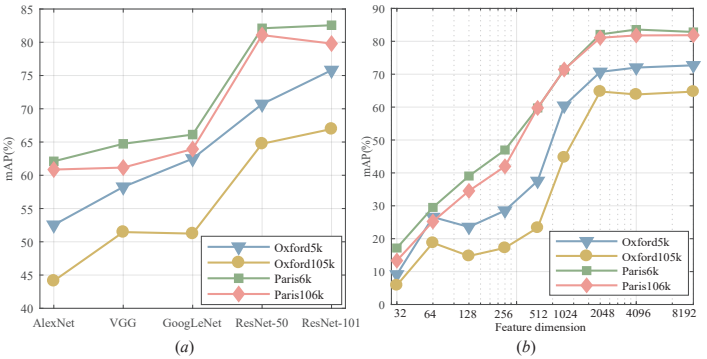


Fig. 11: (a) The effectiveness of different DCNNs on 4 datasets. All models are fine-tuned by the same loss function. The results are tested on the convolutional features with default dimension; (b) The impact of feature dimension on retrieval performance. These features are extracted by using ResNet-50.

Table 2. The recorded time (in *ms*) indicates the average time for searching each query image. In Table 2, we observe some important trends. First, in general, the average retrieval time for each query image is less than 1*s*. Concretely, the recorded time is up to 809*ms* on Oxford-105k in [145], whose mAP is 88.4%. The retrieval time is 600*ms* on Oxford-5k and 700*ms* on Paris-6k in [142], whose time cost is caused by processing 21 regional features on each query image. Second, we observe the retrieval accuracy-efficiency balancing issue, which is significantly obvious on the Oxford-5k dataset. The average retrieval time are both less than 1*ms* in prior work [35], [41], [164], whose mAPs are lower than 70% (*i.e.*, 55.5%, 64.8%, and 64.9%, respectively). In contrast, the prior approaches [145], [157], [142], reach relatively higher mAPs (*i.e.*, 91.3%, 92.6%, and 95.8%, respectively), while this higher accuracy is at the expense of efficiency (more than 2*ms* even up to 600*ms*). Therefore, the trade-off of accuracy and efficiency is also an important factor to take into account in deep image retrieval, especially for large-scale datasets.

In addition, we discuss other important factors which are common for retrieval, including the depth of networks, retrieval feature dimension, and feature aggregation methods.

Network depth. We compare the efficacy of DCNNs depth, following the fine-tuning protocols [1] in [41]. For fair comparisons, all convolutional features from these backbone DCNNs are aggregated by MAC method [47], and fine-tuned by using the same learning rate. That means, the adopted methods are the same except the DCNNs have different depths. We use the default feature dimension (*i.e.* AlexNet (256-d), VGG (512-d), GoogLeNet (1024-d), ResNet-50/101 (2048-d)). The results are reported in Figure 11(a). We observe that the deeper networks is more beneficial for accuracy boosts, due to extracting more discriminative features.

Feature dimension. We focus on varying the feature dimension of ResNet-50 from 32-d to 8192-d, by adding a fully-connected layers on the top of pooled convolutional features. The results are shown in Figure 11(b). It is expected that higher-dimension features capture much more semantics and are helpful for retrieval. However, the performance tends to be stable when the dimension is very large. For ResNet-50, we observe that the 2048-d feature can already produce competitive results.

Feature aggregation methods. Here, we further discuss the methods of embedding convolutional feature maps, as illustrated in Figure 4. We use the off-the-shelf VGG (without updating parameters) on the Oxford and Paris datasets. The results are reported in Figure 10(b). We observe that different ways to aggregate the same off-the-shelf DCNN make differences for retrieval performance. These reported results provide a reference for feature aggregation when one uses convolutional layers for performing retrieval tasks.

1. https://github.com/filipradenovic/cnnimageretrieval-pytorch

6 Conclusions and Future Directions

In this survey, we reviewed deep learning methods for image retrieval, and categorized it into deep image retrieval of off-the-shelf models and fine-tuned models according to the parameter updates of deep networks. Concretely, the off-the-shelf group is concerned with obtaining high-quality features by freezing the pre-stored parameters where network feedforward schemes, layer selection, and feature fusion methods are presented. While fine-tuned based methods deal with updating networks with optimal parameters for feature learning in both supervised and unsupervised approaches. For each group, we presented the corresponding methods and compared their differences. The corresponding experimental results are collected and analyzed for all the categorized works.

Deep learning has shown significant progress and spotlighted its capacity for image retrieval. Despite the great success, there are still many unsolved problems. Here, we introduce some promising trends as future research directions. We hope that this survey not only provides a better understanding of image retrieval but also facilitates future research activities and application developments in this field.

1. Zero-shot Learning for Image Retrieval. The popularity of media platforms and the rapid development of novel techniques makes it very convenient for people to share their images. As a result, the number of images increases immensely. In this case, there often exist "*unseen*" images or categories. However, most datasets are static and offer a limited amount of objects and categories for feature learning. Thus, the retrieval algorithms or systems may suffer from the scarcity of the appropriate training data for these unseen images. Therefore, it is needed to extend conventional image retrieval methods to a zero-shot learning scenario, where we can retrieve both seen and unseen categories from the system. Furthermore, combined with unsupervised methods, the zero-shot learning algorithms can significantly improve the flexibility and generalization of image retrieval systems.

2. End-to-End Unsupervised Retrieval. Using supervisory information, network training or fine-tuning is more likely to mitigate the semantic gap. However, the sophisticated supervised learning algorithms are in most cases not very general because there is usually not enough supervisory information available. Thereby, it is necessary to explore unsupervised image retrieval, which has been studied less [140]. Therefore, as a solution, the earlier noted manifold learning is a way to mine the samples using relevance context information. The self-supervision information is learned based on graph discovery in the manifold space. However, the whole training process is not end-to-end yet. Currently, graph convolutional networks [146] have been used to replace the diffusion process for end-to-end training [42].

3. Incremental Image Retrieval. Current image retrieval focuses on static datasets and is not suited for incremental scenarios [165], [166]. That is, most of these approaches assume that images from all categories are available during training. This assumption may be restrictive in real-world applications as new categories are constantly emerging. Repetitive fine-tuning on both old and new images is time-consuming and inefficient, while fine-tuning only on the new images may lead to catastrophic forgetting, thereby resulting in severe degradation of the retrieval performance for the old categories. Therefore, one practical direction would be to build an up-to-date retrieval model to handle incremental streams of new categories, while retaining its previous performance on existing categories without forgetting.

4. Deploy Image Retrieval for Practical Applications. Existing image retrieval technologies are trained and evaluated on standard benchmarks such as the Oxford and Paris datasets, and various metric learning methods are explored for retrieval on fine-grained datasets. However, these technologies are still far from the real-world applications such as face search, fashion search, person re-identification, shopping recommendation system, or medical image retrieval. In these practical applications, the purpose of image retrieval, may not just be retrieving images for general content on standard benchmarks, but also for more refined information. It is challenging to deploy image retrieval for specific scenario. For example, as a specific instance search topic, person re-identification systems may encounter images with low-resolution or with inferior quality due to inadequate illumination. Existing techniques such as Attention mechanisms and the region proposal networks *etc*. can be adopted to guarantee performance. On the other hand, it is valuable to explore multi-modal retrieval in practical applications. That means, image retrieval can also be combined with other auxiliary modalities such as words, phrases, and sentences to meet different retrieval expectations of users.

ACKNOWLEDGMENT

The authors would like to thank the pioneer researchers in deep image retrieval and other related fields.

REFERENCES

[1] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 2, no. 1, pp. 1–19, 2006.

[3] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015, pp. 1116–1124.

[4] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Comput. Vis. Image Underst.*, vol. 184, pp. 22–30, 2019.

[5] L. R. Nair, K. Subramaniam, and G. Prasannavenkatesan, "A review on multiple approaches to medical image retrieval system," in *Intelligent Computing in Engineering*, 2020, vol. 1125, pp. 501–509.

[6] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016, pp. 1096–1104.

[7] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *ICCV*, 2015, pp. 1269–1277.

[8] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, 2018.

[9] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval." in *AAAI*, 2016, pp. 3457–3463.

[10] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *ECCV*, 2016, pp. 685–701.

[11] R. Furuta, N. Inoue, and T. Yamasaki, "Efficient and interactive spatial-semantic image retrieval," in *MMM*, 2018, pp. 190–202.

[12] L. Zhang and Y. Rui, "Image search from thousands to billions in 20 years," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 9, no. 1s, p. 36, 2013.

[13] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *ACM MM*, 2014, pp. 157–166.

[14] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 20–54, 2015.

[15] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. Snoek, and A. D. Bimbo, "Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval," *ACM Comput. Surv. (CSUR)*, vol. 49, no. 1, pp. 1–39, 2016.

[16] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *arXiv preprint arXiv:1706.06064*, 2017.

[17] L. Piras and G. Giacinto, "Information fusion in content based image retrieval: A comprehensive overview," *Inf. Fusion*, vol. 37, pp. 50–60, 2017.

[18] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, 2018.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015, pp. 91–99.

[24] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *CVPR workshops*, 2014, pp. 806–813.

[25] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014, pp. 392–407.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NeurIPS*, 2014, pp. 3320–3328.

[27] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *ICLR*, 2015, pp. 1–12.

[28] A. Jiménez, J. M. Alvarez, and X. Giró Nieto, "Class-weighted convolutional features for visual instance search," in *BMVC*, 2017, pp. 1–12.

[29] T.-T. Do, T. Hoang, D.-K. L. Tan, H. Le, T. V. Nguyen, and N.-M. Cheung, "From selective deep convolutional features to compact binary representations for image retrieval," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 15, no. 2, pp. 1–22, 2019.

[30] J. Xu, C. Wang, C. Qi, C. Shi, and B. Xiao, "Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval," in *AAAI*, 2018, pp. 7436–7443.

[31] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "Deepindex for accurate and efficient image retrieval," in *ICMR*, 2015, pp. 43–50.

[32] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao, "Online multimodal deep similarity learning with application to image retrieval," in *ACM MM*, 2013, pp. 153–162.

[33] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural

codes for image retrieval," in *ECCV*, 2014, pp. 584–599.
[34] C.-Q. Huang, S.-M. Yang, Y. Pan, and H.-J. Lai, "Object-location-aware hashing for multi-label image retrieval via automatic mask learning," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4490–4502, 2018.
[35] N. Garcia and G. Vogiatzis, "Learning non-metric visual similarity for image retrieval," *Image Vis. Comput.*, vol. 82, pp. 18–25, 2019.
[36] E.-J. Ong, S. Husain, and M. Bober, "Siamese network of deep fisher-vector descriptors for image retrieval," *arXiv preprint arXiv:1702.00338*, 2017.
[37] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *ECCV*, 2016, pp. 241–257.
[38] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *CVPR*, 2016, pp. 5297–5307.
[39] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *ECCV*, 2016, pp. 3–20.
[40] J. Xu, C. Wang, C. Qi, C. Shi, and B. Xiao, "Iterative manifold embedding layer learned by incomplete data for large-scale image retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1551–1562, 2018.
[41] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, 2017.
[42] C. Liu, G. Yu, M. Volkovs, C. Chang, H. Rai, J. Ma, and S. K. Gorti, "Guided similarity separation for image retrieval," in *NeurIPS*, 2019, pp. 1554–1564.
[43] C. Chang, G. Yu, C. Liu, and M. Volkovs, "Explore-exploit graph traversal for image retrieval," in *CVPR*, 2019, pp. 9423–9431.
[44] Y. Shen, J. Qin, J. Chen, M. Yu, L. Liu, F. Zhu, F. Shen, and L. Shao, "Auto-encoding twin-bottleneck hashing," in *CVPR*, 2020, pp. 2818–2827.
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.
[47] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Trans. Media Technol. Appl.*, vol. 4, no. 3, pp. 251–258, 2016.
[48] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim, "Combination of multiple global descriptors for image retrieval," *arXiv preprint arXiv:1903.10663*, 2019.
[49] Y. Li, X. Kong, L. Zheng, and Q. Tian, "Exploiting hierarchical activations of neural network for image retrieval," in *ACM MM*, 2016, pp. 132–136.
[50] C. Qi, C. Shi, J. Xu, C. Wang, and B. Xiao, "Spatial weighted fisher vector for image retrieval," in *ICME*, 2017, pp. 463–468.
[51] E. Mohedano, K. McGuinness, X. Giró-i Nieto, and N. E. O'Connor, "Saliency weighted convolutional features for instance search," in *CBMI*, 2018, pp. 1–6.
[52] F. Yang, J. Li, S. Wei, Q. Zheng, T. Liu, and Y. Zhao, "Two-stream attentive CNNs for image retrieval," in *ACM MM*, 2017, pp. 1513–1521.
[53] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *CVPR*, 2016, pp. 2064–2072.
[54] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4032–4044, 2019.
[55] H. Hu, K. Wang, C. Lv, J. Wu, and Z. Yang, "Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, pp. 739–754, 2018.
[56] D. Deng, R. Wang, H. Wu, H. He, Q. Li, and X. Luo, "Learning deep similarity models with focus ranking for fabric image retrieval," *Image Vis. Comput.*, vol. 70, pp. 11–20, 2018.
[57] K. Zhou, Y. Liu, J. Song, L. Yan, F. Zou, and F. Shen, "Deep self-taught hashing for image retrieval," in *ACM MM*, 2015, pp. 1215–1218.
[58] K. Yan, Y. Wang, D. Liang, T. Huang, and Y. Tian, "CNN vs. SIFT for image retrieval: Alternative or complementary?" in *ACM MM*, 2016, pp. 407–411.
[59] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2868–2881, 2017.
[60] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *BMVC*, 2014.
[61] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *CVPR*, 2003, pp. 1470–1477.
[62] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010, pp. 3304–3311.
[63] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *NeurIPS*, 1999, pp. 487–493.
[64] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *CVPR*, 2014, pp. 1717–1724.
[65] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.
[66] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.
[67] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, 2016.
[68] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marqués, and X. Giro-i Nieto, "Bags of local convolutional features for scalable instance search," in *ICMR*, 2016, pp. 327–331.
[69] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," in *ICLR*, 2015.
[70] D. Xu, X. Alameda-Pineda, J. Song, E. Ricci, and N. Sebe, "Cross-paced representation learning with partial curricula for sketch-based image retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4410–4421, 2018.
[71] J. Cao, L. Liu, P. Wang, Z. Huang, C. Shen, and H. T. Shen, "Where to focus: Query adaptive matching for instance retrieval using convolutional feature maps," *arXiv preprint arXiv:1606.06811*, 2016.
[72] K. Reddy Mopuri and R. Venkatesh Babu, "Object level deep feature pooling for compact image representation," in *CVPR Workshops*, 2015, pp. 62–70.
[73] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014, pp. 391–405.
[74] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional kernel networks," in *NeurIPS*, 2014, pp. 2627–2635.
[75] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval." in *ICCV*, 2017, pp. 5552–5561.
[76] A. Salvador, X. Giró-i Nieto, F. Marqués, and S. Satoh, "Faster r-cnn features for instance search," in *CVPR Workshops*, 2016, pp. 9–16.
[77] J. Yue-Hei Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *CVPR workshops*, 2015, pp. 53–61.
[78] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in CNN feature transfer," *CoRR*, vol. abs/1604.00133, 2016.
[79] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk, "SOLAR: Second-order loss and attention for image retrieval," in *ECCV*, 2020, pp. 253–270.
[80] P. Kulkarni, J. Zepeda, F. Jurie, P. Perez, and L. Chevallier, "Hybrid multi-layer deep CNN/aggregator feature for image classification," in *ICASSP*, 2015, pp. 1379–1383.
[81] D. Yu, Y. Liu, Y. Pang, Z. Li, and H. Li, "A multi-layer deep fusion convolutional neural network for sketch based image retrieval," *Neurocomputing*, vol. 296, pp. 23–32, 2018.
[82] W. Yu, K. Yang, H. Yao, X. Sun, and P. Xu, "Exploiting the complementary strengths of multi-layer CNN features for image retrieval," *Neurocomputing*, vol. 237, pp. 235–241, 2017.
[83] C. Shen, C. Zhou, Z. Jin, W. Chu, R. Jiang, Y. Chen, and X.-S. Hua, "Learning feature embedding with strong neural activations for fine-grained retrieval," in *ACM MM*, 2017, pp. 424–432.
[84] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for efficient image search," in *ECCV*, 2020, pp. 726–743.
[85] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *CVPR*, 2016, pp. 2167–2175.
[86] Z. Ding, L. Song, X. Zhang, and Z. Xu, "Selective deep ensemble for instance retrieval," *Multimed. Tools. Appl.*, pp. 1–17, 2018.
[87] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon, "Attention-based ensemble for deep metric learning," in *ECCV*, 2018, pp. 736–751.
[88] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression," *Comput. Graph.*, vol. 71, pp. 77–87, 2018.
[89] K. Ozaki and S. Yokoo, "Large-scale landmark retrieval/recognition under a noisy and diverse dataset," in *CVPR Workshop*, 2019.
[90] H. Xuan, R. Souvenir, and R. Pless, "Deep randomized ensembles for metric learning," in *ECCV*, 2018, pp. 723–734.
[91] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *CVPR*, 2016, pp. 598–606.

[92] B.-C. Chen, L. S. Davis, and S.-N. Lim, "An analysis of object embeddings for image retrieval," *arXiv preprint arXiv:1905.11903*.

[93] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *ICML*, 2010, pp. 111–118.

[94] F. Wang, W.-L. Zhao, C.-W. Ngo, and B. Merialdo, "A hamming embedding kernel with informative bag-of-visual words for video semantic indexing," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 10, no. 3, pp. 1–20, 2014.

[95] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, 2012.

[96] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.

[97] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5012–5024, 2016.

[98] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Largescale image retrieval with attentive deep local features," in *ICCV*, 2017, pp. 3456–3465.

[99] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, 2018.

[100] J. Cao, Z. Huang, and H. T. Shen, "Local deep descriptors in bag-of-words for image retrieval," in *ACM MM*, 2017, pp. 52–58.

[101] V. Chandrasekhar, J. Lin, O. Morere, H. Goh, and A. Veillard, "A practical guide to CNNs and fisher vectors for image instance retrieval," *Signal Process.*, vol. 128, pp. 426–439, 2016.

[102] J. Kim and S.-E. Yoon, "Regional attention based deep feature for image retrieval," in *BMVC*, 2018, pp. 209–223.

[103] B. Chen and W. Deng, "Hybrid-attention based decoupled metric learning for zero-shot image retrieval," in *CVPR*, 2019, pp. 2750–2759.

[104] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Trans. Neural. Netw. Learn. Syst.*, 2019.

[105] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," in *CVPR*, 2017, pp. 2862–2871.

[106] R. Kang, Y. Cao, M. Long, J. Wang, and P. S. Yu, "Maximum-margin hamming hashing," in *ICCV*, 2019, pp. 8252–8261.

[107] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *CVPR*, 2015, pp. 1556–1564.

[108] F. Long, T. Yao, Q. Dai, X. Tian, J. Luo, and T. Mei, "Deep domain adaptation hashing with adversarial learning," in *ACM SIGIR*, 2018, pp. 725–734.

[109] Y. Cao, B. Liu, M. Long, J. Wang, and M. KLiss, "Hashgan: Deep learning to hash with pair conditional wasserstein gan," in *CVPR*, 2018, pp. 1287–1296.

[110] E. Yang, T. Liu, C. Deng, W. Liu, and D. Tao, "Distillhash: Unsupervised deep hashing by distilling data pairs," in *CVPR*, 2019, pp. 2946–2955.

[111] M. A. Carreira-Perpinán and R. Raziperchikolaei, "Hashing with binary autoencoders," in *CVPR*, 2015, pp. 557–566.

[112] T.-T. Do, D.-K. Le Tan, T. T. Pham, and N.-M. Cheung, "Simultaneous feature aggregating and hashing for large-scale image search," in *CVPR*, 2017, pp. 6618–6627.

[113] Y. Gu, S. Wang, H. Zhang, Y. Yao, W. Yang, and L. Liu, "Clustering-driven unsupervised deep hashing for image retrieval," *Neurocomputing*, vol. 368, pp. 114–123, 2019.

[114] J. Song, "Binary generative adversarial networks for image retrieval," in *AAAI*, 2017.

[115] K. G. Dizaji, F. Zheng, N. S. Nourabadi, Y. Yang, C. Deng, and H. Huang, "Unsupervised deep generative adversarial hashing network," in *CVPR*, 2018, pp. 3664–3673.

[116] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *CVPR*, 2015, pp. 2475–2483.

[117] F. Cakir, K. He, S. A. Bargal, and S. Sclaroff, "Hashing with mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2424–2437, 2019.

[118] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008, pp. 304–317.

[119] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007, pp. 1–8.

[120] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *CVPR*, 2017, pp. 403–412.

[121] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *ICCV*, 2017, pp. 2593–2601.

[122] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016, pp. 1857–1865.

[123] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016, pp. 4004–4012.

[124] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *CVPR*, 2019, pp. 5207–5216.

[125] L. Chen and Y. He, "Dress fashionably: Learn fashion collocation with deep mixed-category metric learning," in *AAAI*, 2018, pp. 2103–2110.

[126] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *ICCV*, 2017, pp. 360–368.

[127] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *CVPR*, 2020, pp. 3238–3247.

[128] W. Zheng, Z. Chen, J. Lu, and J. Zhou, "Hardness-aware deep metric learning," in *CVPR*, 2019, pp. 72–81.

[129] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *CVPR*, 2014, pp. 1386–1393.

[130] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *ICCV*, 2015, pp. 118–126.

[131] J. Song, T. He, L. Gao, X. Xu, and H. T. Shen, "Deep region hashing for efficient large-scale instance search from images," *arXiv preprint arXiv:1701.07901*, 2017.

[132] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 237–254, 2017.

[133] J. Lin, O. Morere, A. Veillard, L.-Y. Duan, H. Goh, and V. Chandrasekhar, "Deephash for image instance retrieval: Getting regularization, depth and fine-tuning right," in *ICMR*, 2017, pp. 133–141.

[134] J. Cao, Z. Huang, P. Wang, C. Li, X. Sun, and H. T. Shen, "Quartet-net learning for visual instance retrieval," in *ACM MM*, 2016, pp. 456–460.

[135] X. Wang, H. Zhang, W. Huang, and M. R. Scott, "Cross-batch memory for embedding learning," in *CVPR*, 2020, pp. 6388–6397.

[136] B. Harwood, B. Kumar, G. Carneiro, I. Reid, T. Drummond *et al.*, "Smart mining for deep metric learning," in *ICCV*, 2017, pp. 2821–2829.

[137] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *CVPR*, 2018, pp. 596–605.

[138] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *ICCV*, 2019, pp. 5107–5116.

[139] A. Brown, W. Xie, V. Kalogeiton, and A. Zisserman, "Smooth-ap: Smoothing the path towards large-scale image retrieval," in *ECCV*, 2020, pp. 677–694.

[140] N. Aziere and S. Todorovic, "Ensemble deep manifold similarity learning using hard proxies," in *CVPR*, 2019, pp. 7299–7307.

[141] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *CVPR*, 2013, pp. 1320–1327.

[142] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations," in *CVPR*, 2017, pp. 2077–2086.

[143] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Mining on manifolds: Metric learning without labels," in *CVPR*, 2018, pp. 7642–7651.

[144] Y. Zhao, L. Wang, L. Zhou, Y. Shi, and Y. Gao, "Modelling diffusion process by deep neural networks for image retrieval," in *BMVC*, 2018, pp. 161–174.

[145] B. Song, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1213–1226, 2018.

[146] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[147] T. Maria and T. Anastasios, "Deep convolutional image retrieval: A general framework," *Signal Process. Image Commun.*, vol. 63, pp. 30–43, 2018.

[148] R.-C. Tu, X.-L. Mao, B.-S. Feng, and S.-Y. Yu, "Object detection based deep unsupervised hashing," in *IJCAI*, 2019, pp. 3606–3612.

[149] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski, "Bingan: learning compact binary descriptors with a regularized gan," in *NeurIPS*, 2018, pp. 3612–3622.

[150] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.

[151] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008, pp. 1–8.

[152] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *CVPR*, 2018.
[153] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *Int. J. Comput. Vis.*, vol. 120, no. 1, pp. 1–13, 2016.
[154] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105, 2017.
[155] L. P. Valem and D. C. G. Pedronette, "Graph-based selective rank fusion for unsupervised image retrieval," *Pattern Recognit Lett*, 2020.
[156] L. T. Alemu and M. Pelillo, "Multi-feature fusion for image retrieval using constrained dominant sets," *Image Vis Comput*, vol. 94, p. 103862, 2020.
[157] F. Yang, R. Hinami, Y. Matsui, S. Ly, and S. Satoh, "Efficient image retrieval via decoupling diffusion into online and offline processing," in *AAAI*, vol. 33, 2019, pp. 9087–9094.
[158] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.
[159] S. S. Husain and M. Bober, "Remap: Multi-layer entropy-guided pooling of dense cnn features for image retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5201–5213, 2019.
[160] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum, "Fast spectral ranking for similarity search," in *CVPR*, 2018, pp. 7632–7641.
[161] J. Yang, J. Liang, H. Shen, K. Wang, P. L. Rosin, and M.-H. Yang, "Dynamic match kernel with deep convolutional features for image retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5288–5302, 2018.
[162] H.-F. Yang, K. Lin, and C.-S. Chen, "Cross-batch reference learning for deep classification and retrieval," in *ACM MM*, 2016, pp. 1237–1246.
[163] Y. Lv, W. Zhou, Q. Tian, S. Sun, and H. Li, "Retrieval oriented deep feature learning with complementary supervision mining," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4945–4957, 2018.
[164] Q. Wang, J. Lai, L. Claesen, Z. Yang, L. Lei, and W. Liu, "A novel feature representation: Aggregating convolution kernels for image retrieval," *Neural Networks*, vol. 130, pp. 1–10, 2020.
[165] W. Chen, Y. Liu, W. Wang, T. Tuytelaars, E. M. Bakker, and M. Lew, "On the exploration of incremental learning for fine-grained image retrieval," in *BMVC*, 2020.
[166] D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang, "Deep incremental hashing network for efficient image retrieval," in *CVPR*, 2019, pp. 9069–9077.

TABLE 3: Performance evaluation of off-the-shelf DCNN models. "•" indicates that the models or layers are combined to learn features; "PCA$_w$ indicates PCA with whitening on the extracted features to improve robustness; "MP" means Max Pooling; "SP" means Sum Pooling. The CNN-M network with "$*$" has an architecture similar to that of AlexNet. "-" means that the results were not reported.

Type	Method	Backbone DCNN	Output Layer	Feature Enhance.	Feature Dimension	Holidays	UKB	Oxford5k (+100k)	Paris6k (+100k)	Brief Conclusions and Highlights
Single Pass	Neural codes [33]	AlexNet	FC6	PCA	128	74.7	3.42 (N-S)	43.3 (38.6)	-	Compressed neural codes of different layers are explored. AlexNet is also fine-tuned for retrieval.
	R-MAC [27]	VGG16	Conv5	R-MAC + PCA$_w$	512	-	-	66.9 (61.6)	83.0 (75.7)	Adopting sliding windows with different scales on the convolutional feature maps to preserve spatial information.
	CroW [10]	VGG16	Conv5	CroW + PCA$_w$	256	85.1	-	68.4 (63.7)	76.5 (69.1)	The spatial- and channel-wise weighting mechanisms are utilized to highlight crucial convolutional features.
	BLCF [68]	VGG16	Conv5	BoW + PCA$_w$	25k	-	-	73.9 (59.3)	82.0 (64.8)	Both global features and local features are explored, demonstrating that local features have higher accuracy.
	SPoC [7]	VGG16	Conv5	SPoC + PCA$_w$	256	80.2	3.65 (N-S)	58.9 (57.8)	-	Exploring Gassian weighting scheme *i.e.*, the centering prior, to improve the discrimination of features.
	Multi-layer CNN [82]	VGG16	FC6 • Conv4∼5	SP	4096	91.4	3.68 (N-S)	61.5 (-)	-	Layer-level feature fusion and the complementary properties of different layers are explored.
Multiple Pass	Deepindex [31]	AlexNet • VGG19	FC6-7 • FC17-18	BoW + PCA	512	81.7	3.32 (N-S)	-	75.4 (-)	Exploring layer-level and model-level fusion methods. Image patches are extracted using spatial pyramid modeling.
	MOF [49]	CNN-M* [60]	FC7 • Conv	SP or MP + BoW	20k	76.8	3.00 (N-S)	-	-	Exploring layer-level fusion scheme. Image patches are extracted using spatial pyramid modeling.
	Multi-scale CNN [47]	VGG16	Conv5	SP or MP + PCA$_w$	32k	89.6	95.1 (mAP)	84.3 (-)	87.9 (-)	Image patches are extracted in a dense manner. Geometric invariance is considered when aggregating patch features.
	CNNaug-ss [24]	Overfeat [158]	FC	PCA$_w$	15k	84.3	91.1 (mAP)	68.0 (-)	79.5 (-)	Image patches are extracted densely. Image regions at different locations with different sizes are included.
	MOP-CNN [25]	AlexNet	FC7	VLAD + PCA$_w$	2048	80.2	-	-	-	Image patches are extracted densely. Multi-scale patch features are further embedded into VLAD descriptors.
	CCS [58]	GoogLeNet	Conv	VLAD + PCA$_w$	128	84.1	3.81 (N-S)	64.8 (-)	76.8 (-)	Object proposals are extracted by RPNs. Object-level and point-level feature concatenation schemes are explored.
	OLDFP [72]	AlexNet	FC6	MP + PCA$_w$	512	88.5	3.81 (N-S)	60.7 (-)	66.2 (-)	Exploring the impact of proposal number. Patches are extracted by RPNs and the features are encoded in an orderless way.
	LDD [100]	VGG19	Conv5	BoW + PCA$_w$	500k	84.6	-	83.3 (-)	87.2 (-)	Image patches are obtained using a uniform square mesh. Patch features are encoded into BoW descriptors.

TABLE 4: Performance evaluation of methods in which DCNN models are fine-tuned, in a supervised or an unsupervised manner. "CE Loss" means the models are fine-tuned using the classification-based loss function in the form of Eq. 10. "Siamese Loss" is in the form of Eq. 13. "Regression Loss" is in the form of Eq. 12. "Triplet Loss" is in the form of Eq. 14.

Type	Method	Backbone DCNN	Output Layer	Feature Enhance.	Loss Function	Feature Dimension	Holidays	UKB	Oxford5k (+100k)	Paris6k (+100k)	Brief Conclusions and Highlights
Supervised Fine-tuning	DELF [98]	ResNet-101	Conv4 Block	Attention + PCA$_w$	CE Loss	2048	-	-	83.8 (82.6)	85.0 (81.7)	Exploring the FCN to construct feature pyramids of different sizes.
	Neural codes [33]	AlexNet	FC6	PCA	CE Loss	128	78.9	3.29 (N-S)	55.7 (52.3)	-	The first work which fine-tunes deep networks for image retrieval. Compressed neural codes and different layers are explored.
	Non-metric [35]	VGG16	Conv5	PCA$_w$	Regression Loss	512	-	-	88.2 (82.1)	88.2 (82.9)	Visual similarity learning of similar and dissimilar pairs is performed by a neural network, optimized using regression loss.
	Faster R-CNN [76]	VGG16	Conv5	MP / SP	Regression Loss	512	-	-	75.1 (-)	80.7 (-)	RPN is fine-tuned, based on bounding box coordinates and class scores for specific region query which is region-targeted.
	SIAM-FV [36]	VGG16	Conv5	FV + PCA$_w$	Siamese Loss	512	-	-	81.5 (76.6)	82.4 (-)	Fisher Vector is integrated on top of VGG and is trained with VGG simultaneously.
	SIFT-CNN [163]	VGG16	Conv5	SP	Siamese Loss	512	88.4	3.91 (N-S)	-	-	SIFT features are used as supervisory information for mining positive and negative samples.
	Quartet-Net [134]	VGG16	FC6	PCA	Siamese Loss	128	71.2	87.5 (mAP)	48.5 (-)	48.8 (-)	Quartet-net learning is explored to improve feature discrimination where double-margin contrastive loss is used.
	NetVLAD [38]	VGG16	VLAD Layer	PCA$_w$	Triplet Loss	256	79.9	-	62.5 (-)	72.0 (-)	VLAD is integrated at the last convolutional layer of VGG16 network as a plugged layer.
	Deep Retri-eval [132]	ResNet-101	Conv5 Block	MP + PCA$_w$	Triplet Loss	2048	90.3	-	86.1 (82.8)	94.5 (90.6)	Dataset is cleaned automatically. Features are encoded by R-MAC. RPN is used to extract the most relevant regions.
Unsupervised Fine-tuning	MoM [143]	VGG16	Conv5	MP + PCA$_w$	Siamese Loss	64	87.5	-	78.2 (72.6)	85.1 (78.0)	Exploring manifold learning for mining dis/similar samples. Features are tested globally and regionally.
	GeM [41]	VGG16	Conv5	GeM Pooling	Siamese Loss	512	83.1	-	82.0 (76.9)	79.7 (72.6)	Fine-tuning CNNs on an unordered dataset. Samples are selected from an automated 3D reconstruction system.
	SfM-CNN [39]	VGG16	Conv5	PCA$_w$	Siamese Loss	512	82.5	-	77.0 (69.2)	83.8 (76.4)	Employing Structure-from-Motion to select positive and negative samples from unordered images.
	IME-CNN [40]	ResNet-101	IME Layer	MP	Regression Loss	2048	-	-	92.0 (87.2)	96.6 (93.3)	Graph-based manifold learning is explored within an IME layer to mine the matching and non-matching pairs in unordered datasets.
	MDP-CNN [144]	ResNet-101	Conv5 Block	SP	Triplet Loss	2048	-	-	85.4 (85.1)	96.3 (94.7)	Exploring global feature structure by modeling the manifold learning to select positive and negative pairs.

A Decade Survey of Content Based Image Retrieval using Deep Learning

Shiv Ram Dubey

Abstract—The content based image retrieval aims to find the similar images from a large scale dataset against a query image. Generally, the similarity between the representative features of the query image and dataset images is used to rank the images for retrieval. In early days, various hand designed feature descriptors have been investigated based on the visual cues such as color, texture, shape, etc. that represent the images. However, the deep learning has emerged as a dominating alternative of hand-designed feature engineering from a decade. It learns the features automatically from the data. This paper presents a comprehensive survey of deep learning based developments in the past decade for content based image retrieval. The categorization of existing state-of-the-art methods from different perspectives is also performed for greater understanding of the progress. The taxonomy used in this survey covers different supervision, different networks, different descriptor type and different retrieval type. A performance analysis is also performed using the state-of-the-art methods. The insights are also presented for the benefit of the researchers to observe the progress and to make the best choices. The survey presented in this paper will help in further research progress in image retrieval using deep learning.

Index Terms—Content Based Image Retrieval; Deep Learning; Convolutional Neural Networks; Survey; Supervised and Unsupervised Learning.

1 Introduction

Image retrieval is a well studied problem of image matching where the similar images are retrieved from a database w.r.t. a given query image [1], [2]. Basically, the similarity between the query image and the database images is used to rank the database images in decreasing order of similarity [3]. Thus, the performance of any image retrieval method depends upon the similarity computation between images. Ideally, the similarity score computation method between two images should be discriminative, robust and efficient. The easiest way to compute the similarity between two images is to find the sum of absolute difference of corresponding pixels in both the images, i.e., L_1 distance. This method is also referred as the template matching. However, this approach is not robust against the image geometric and photometric changes, such as translation, rotation, viewpoint, illumination, etc. It is demonstrated in Fig. 1 with the help of two pictures of the same category of Corel dataset [4] and corresponding representative intensity values of a window. Another problem with this approach is that it is not efficient due to the high dimensionality of the image which leads to the high computation requirement to find the similarity between the query and database images.

1.1 Hand-crafted Descriptor based Image Retrieval

In order to make the retrieval robust to geometric and photometric changes, the similarity between images is computed based on the content of images. Basically, the content of the images (i.e., the visual appearance) in terms of the color, texture, shape, gradient, etc. are represented in the form of a feature descriptor [6]. The similarity between the feature vectors of the corresponding images is treated as the similarity between the images. Thus, the performance of any content based image retrieval (CBIR) method heavily depends upon the feature descriptor representation of the image. Any feature descriptor representation method is expected to have the discriminating ability, robustness and low dimensionality. Fig. 2 illustrates the effect of descriptor function in terms of its robustness. The rotation and scale hybrid descriptor (RSHD) function [7] is used to show the rotation invariance between an image taken from Corel-dataset [4] and its rotated version. It can be seen in Fig. 2 that the raw intensity values based comparison does not work, however the descriptor based comparison works given that the descriptor function is able to capture the relevant information from the image. Various feature descriptor representation methods have been investigated to compute the similarity between the two images for content based image retrieval. The feature descriptor representation utilizes the visual cues of the images selected manually based on the need [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. These approaches are also termed as the hand-designed or hand-engineered feature description. Moreover, generally these methods are unsupervised as they do not need the data to design the feature representation method. Various survey has been also conducted time to time to present the progress in content based image retrieval, including

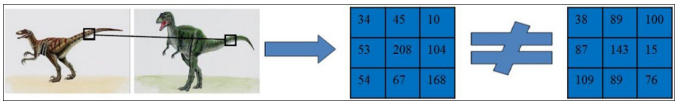


Fig. 1: Comparing pixels of two regions (images are taken from Corel-database [4]). The presented raw intensity values are only indicative not actual. The uses of raw intensity values for the image similarity computation is not a good idea as it is not robust against the geometric and photometric changes. This figure has been originally appeared in [5].

S.R. Dubey is with the Computer Vision Group, Indian Institute of Information Technology, Sri City, Chittoor, Andhra Pradesh-517646, India (e-mail: shivram1987@gmail.com, srdubey@iiits.in).

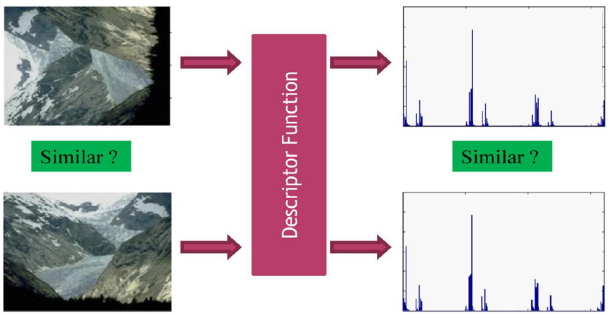


Fig. 2: Depicting rotation robustness of descriptor function. The image is taken from Corel-database [4]. The 1^{st} image is 90^{o} rotated version of 2^{nd} image in counter-clockwise direction. The rotation and scale invariant hybrid descriptor (RSHD) [7] is used as the descriptor function in this example. In spite of having the differences in the intensity values at the corresponding pixels between both images, the feature descriptors are very much similar. This figure has been originally appeared in [5].

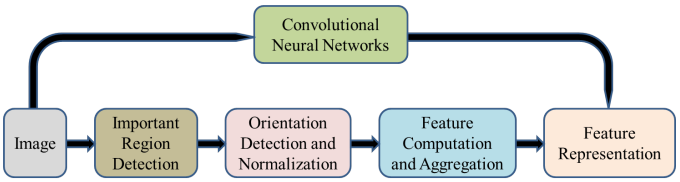


Fig. 3: The pipeline of state-of-the-art feature representation is replaced by the CNN based feature representation with increased discriminative ability and robustness.

[2] in 2000, [20] in 2002, [21] in 2004, [22] in 2006, [23] in 2007, [24] in 2008, [25] in 2014 and [26] in 2017. The hand-engineering feature for image retrieval was a very active research area. However, its performance was limited as the hand-engineered features are not able to represent the image characteristics in an accurate manner.

1.2 Distance Metric Learning based Image Retrieval

The distance metric learning has been also used very extensively for feature vectors representation [27]. It is also explored well for image retrieval [28]. Some notable deep metric learning based image retrieval approaches include Contextual constraints distance metric learning [29], Kernel-based distance metric learning [30], Visuality-preserving distance metric learning [31], Rank-based distance metric learning [32], Semi-supervised distance metric learning [33], etc. Generally, the deep metric learning based approaches have shown the promising retrieval performance compared to hand-crafted approaches. However, most of the existing deep metric learning based methods rely on the linear distance functions which limits its discriminative ability and robustness to represent the non-linear data for image retrieval. Moreover, it is also not able to handle the multi-modal retrieval effectively.

1.3 Deep Learning based Image Retrieval

From a decade, a shift has been observed in feature representation from hand-engineering to learning-based after the emergence of deep learning [34], [35]. This transition is depicted in Fig. 3 where the convoltional neural networks based feature learning replaces

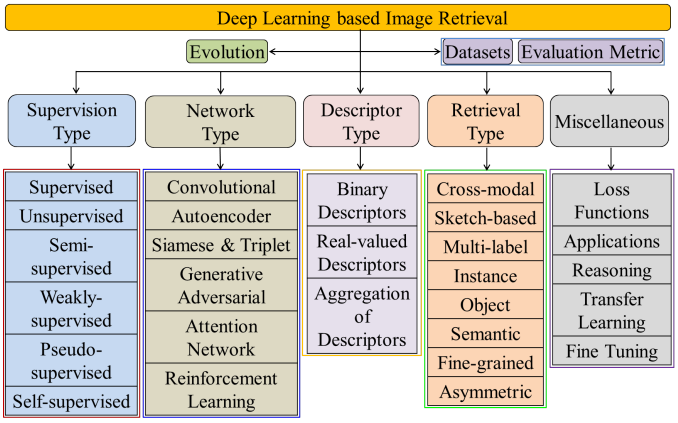


Fig. 4: Taxonomy used in this survey to categorize the existing deep learning based image retrieval approaches.

the state-of-the-art pipeline of traditional hand-engineered feature representation. The deep learning is a hierarchical feature representation technique to learn the abstract features from data which are important for that dataset and application [36]. Based on the type of data to be processed, different architectures came into existence such as Artificial Neural Network (ANN)/ Multilayer Perceptron (MLP) for 1-D data [37], [38], [39], Convolutional Neural Networks (CNN) for image data [40], [41], [42], and Reurrent Neural Networks (RNN) for time-series data [43], [44], [45]. The CNN features off-the-shelf have shown very promising performance for the object recognition and retrieval tasks in terms of the discriminative power and robustness [34]. A huge progress has been made in this decade to utilize the power of deep learning for content based image retrieval [46], [47], [48], [49]. Thus, this survey mainly focuses over the progress in state-of-the-art deep learning based models and features for content based image retrieval from its inception. A taxonomy of state-of-the-art deep learning approaches for image retrieval is portrayed in Fig. 4.

The major contributions of this survey, w.r.t. the existing literature, can be outlined as follows:

1) As per my best knowledge, this survey can be seen as the first of its kind to cover the deep learning based image retrieval approaches very comprehensively in terms of evolution of image retrieval using deep learning, different supervision type, network type, descriptor type, retrieval type and other aspects.
2) In contrast to the recent reviews [47], [28], [48], this survey specifically covers the progress in image retrieval using deep learning progress in 2011-2020 decade rather than hand-crafted and distance metric learning based approaches. Moreover, we provide a very informative taxonomy (refer Fig. 4) with wide coverage of existing deep learning based image retrieval approaches as compared to the recent survey [49].
3) This survey enriches the reader with the state-of-the-art image retrieval using deep learning methods with analysis from various perspectives.
4) This paper also presents the brief highlights and important discussions along with the comprehensive comparisons on benchmark datasets using the state-of-the-art deep learning based image retrieval approaches (Refer Table 3, 4, and 5).

TABLE 1: The summary of large-scale datasets for deep learning based image retrieval.

Dataset	Year	#Classes	Training	Test	Image Type
CIFAR-10 [50]	2009	10	50,000	10,000	Object Category Images
NUS-WIDE [51]	2009	21	97,214	65,075	Scene Images
MNIST [52]	1998	10	60,000	10,000	Handwritten Digit Images
SVHN [53]	2011	10	73257	26032	House Number Images
SUN397 [54]	2010	397	100,754	8,000	Scene Images
UT-ZAP50K [55]	2014	8	42,025	8,000	Shoes Images
Yahoo-1M [56]	2015	116	1,011,723	112,363	Clothing Images
ILSVRC2012 [57]	2012	1,000	∼1.2 M	50,000	Object Category Images
MS COCO [58]	2014	80	82,783	40,504	Common Object Images
MIRFlicker-1M [59]	2010	-	1 M	-	Scene Images
Google Landmarks [60]	2017	15 K	∼1 M	-	Landmark Images
Google Landmarks v2 [61]	2020	200 K	5 M	-	Landmark Images

This survey is organized as follows: the background is presented in Section 2 in terms of the datasets and evaluation measures; the evolution of deep learning based image retrieval is compiled in Section 3; the categorization of existing approaches based on the supervision type, network type, descriptor type, and retrieval type are discussed in Section 4, 5, 6, and 7, respectively; Some other aspects are highlighted in Section 8; the performance comparison of the popular methods is performed in Section 9; conclusions and future directions are presented in Section 10.

2 Background

In this section the background is presented in terms of the commonly used performance evaluation metrics and benchmark retrieval datasets.

2.1 Retrieval Evaluation Measures

In order to judge the performance of image retrieval approaches, precision, recall and f-score are the common evaluation metrics. The mean average precision (mAP) is very commonly used in the literature. The precision is defined as the percentage of correctly retrieved images out of the total number of retrieved images. The recall is another performance measure being used for image retrieval by computing the percentage of correctly retrieved images out of the total number of relevant images present in the dataset. The f-score is computed from the harmonic mean of precision and recall as $(2 \times precision \times recall)/(precision + recall)$. Thus, the f-score provides a trade-off between precision and recall.

2.2 Datasets

With the inception of deep learning models, various large-scale datasets have been created to facilitate the research in image recognition and retrieval. The details of large-scale datasets are summarized in Table 1. Datasets having various types of images are available to test the deep learning based approaches such as object category datasets [50], [57], [58], scene datasets [51], [54], [62], digit datasets [52], [53], apparel datasets [55], [56], landmark datasets [60], [61], etc. The CIFAR-10 dataset is very widely used object category datset [50]. The ImageNet (ILSVRC2012), a large-scale dataset, is also an object category dataset with more than a million number of images [57]. The MS COCO dataset [58] created for common object detection is also utilized for image retrieval purpose. Among scene image datasets commonly used for retrieval purpose, the NUS-WIDE dataset is from National University of Singapore [51]; the Sun397 is a scene understanding dataset from 397 categories with more than one lakh images [54], [63]; and the MIRFlicker-1M [62] dataset consists of a million images downloaded from the social photography site Flickr. The MNIST dataset is one of the old and large-scale digit image datasets [52] consisting of optical characters. The SVHN is another digit dataset [53] from the street view house number images which is more complex than MNIST dataset. The shoes apparel dataset, namely UT-ZAP50K [55], consists of roughly 50K images. The Yahoo-1M is another apparel large-scale dataset used in [56] for image retrieval. The Google landmarks dataset is having around a million landmark images [60]. The extended version of Google landmarks (i.e., v2) [61] contains around 5 million landmark images. There are more datasets used for retrieval in the literature, such as Corel, Oxford, Paris, etc., however, these are not the large-scale datasets. The CIFAR-10, MNIST, SVHN and ImageNet are the widely used datasets in majority of the research.

3 Evolution of Deep Learning for Content Based Image Retrieval (CBIR)

The deep learning based generation of descriptors or hash codes is the recent trends large-scale content based image retrieval, due to its computational efficiency and retrieval quality [28]. The deep learning driven features led to the improved retrieval quality. Recently, it has received increasing attention to utilize the features for image retrieval using end-to-end representation learning. In this section, a journey of deep learning models for image retrieval from 2011 to 2020 is presented. A chronological overview of different methods is illustrated in Fig. 5. Rest of this section highlights the selected methods in chronological manner.

3.1 Chronological Overview: 2011 - 2015

3.1.1 2011-2013

Among the initial attempts, in 2011, Krizhevsky and Hinton have used a deep autoencoder to map the images to short binary codes for content based image retrieval (CBIR) [64]. Kang et al. (2012) have proposed a deep multi-view hashing to generate the code for CBIR from multiple views of data by modeling the layers with view-specific and shared hidden nodes [65]. In 2013, Wu et al. have considered the multiple pretrained stacked denoising autoencoders over low features of the images [66]. They also fine tune the multiple deep networks on the output of the pretrained autoencoders and integrated to generate the multi-modal similarity function for image retrieval.

3.1.2 2014

In an outstanding work, Babenko et al. (2014) have utilized the activations of the top layers of a large convolutional neural network (CNN) as the descriptors (neural codes) for image retrieval application [67] as depicted in Fig. 6. A very promising performance has been recorded using the neural codes for image retrieval even if the model is trained on un-related data. The retrieval results are further improved by re-training the model over similar data and then extracting the neural codes as the descriptor. They also compress the neural code using principal component analysis (PCA) to generate the compact descriptor. In 2014, Wang et al. have investigated a deep ranking model by learning the similarity metric directly from images [68]. Basically, they have employed the triplets to capture the inter-class and intra-class image differences to improve the discriminative ability of the learnt latent space as the descriptor.

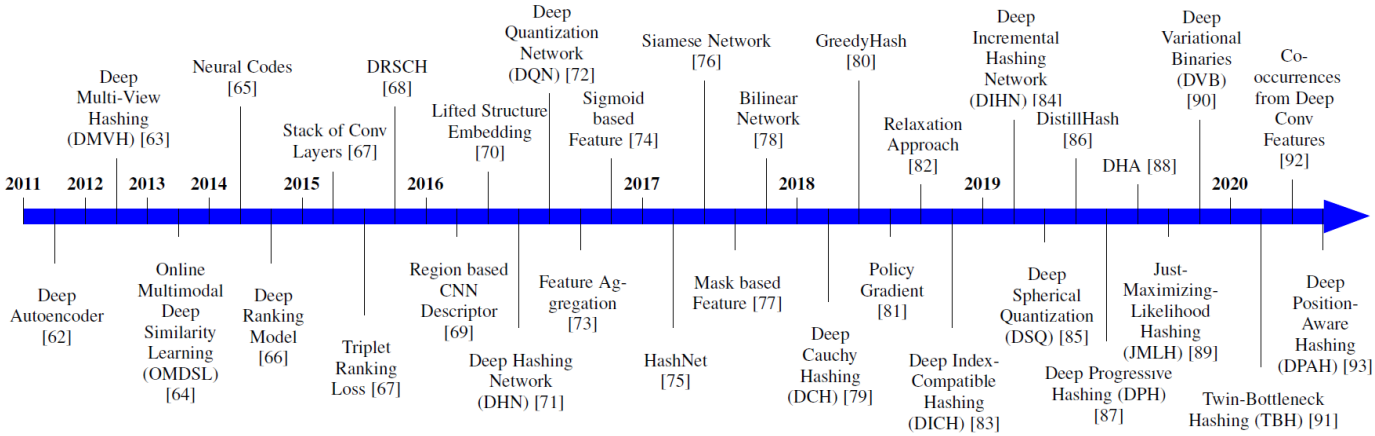


Fig. 5: A chronological view of deep learning based image retrieval methods depicting its evolution from 2011 to 2020.

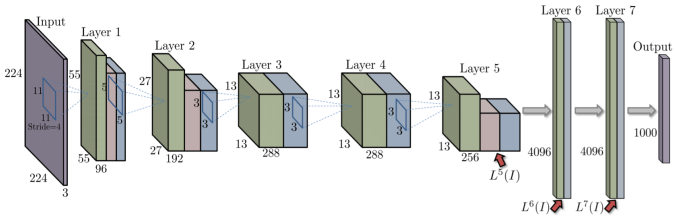


Fig. 6: The illustration of the neural code generation from a convolutional neural network (CNN) [67]. The outputs of layer 5, layer 6 and layer 7 are used to generate the neural code. This figure has been originally shown in [67].

3.1.3 2015

In 2015, Lai et al. have used a deep architecture consisting of a stack of convolution layers to produce the intermediate image features [69]. They have generated the hash bits from the different branches of the intermediate image features. The triplet ranking loss is also utilized to incorporate the inter-class and intra-class differences in [69] for image retrieval. Zhang et al. (2015) have developed a deep regularized similarity comparison hashing (DRSCH) by training a deep CNN model in an end-to-end fashion to simultaneously optimize the discriminative image features and hash functions [70]. They have weighted each bit unequally to make bit-scalable and to prune the redundant bits.

3.2 Chronological Overview: 2016 - 2020

3.2.1 2016

In 2016, Gordo et al. have pooled the relevant regions to form the descriptor with the help of a region proposal network to prioritize the important object regions leading to better retrieval performance [71]. Song et al. (2016) have learnt the lifted structure embedding by computing the lifted structure loss between the CNN and the original features [72]. Zhu et al. have proposed a supervised deep hashing network (DHN) by learning the important image representation for hash codes and controlling the quantization error [73]. At the same time, Cao et al. have introduced a deep quantization network (DQN) which is very similar to the DHN model [74]. The CNN based features are aggregated by Husain and Bober (2016) with the help of rank-aware multi-assignment and direction based combination [75]. Zhong et al. have added a sigmoid layer before the loss layer of a CNN to learn the binary code for CBIR [76].

3.2.2 2017

In 2017, Cao et al. have proposed HashNet deep architecture to generate the hash code by a continuation method [77]. It learns the non-smooth binary activations using the continuation method to generate the binary hash codes from imbalanced similarity data. Gordo et al. (2017) have shown that the noisy training data, inappropriate deep architecture and suboptimal training procedure are the main hurdle to utilize the deep learning for image retrieval [78]. They have performed the cleaning step to improve the dataset and utilized the Siamese network for learning the image representations and reported the mean average precision (%) of 94.7, 96.6, and 94.8 over Oxford 5k, Paris 6k and Holidays datasets, respectively. Different masking schemes such as SUM-mask and MAX-mask are used in [79] to select the prominent CNN features for image retrieval. A bilinear network with two parallel CNNs is also used as the compact feature extractors for CBIR [80] and reported the mean average precision of 95.7% on Oxford5K and 88.6% on Oxford105K datasets with feature vector of 16-length.

3.2.3 2018

In 2018, Cao et al. have investigated a deep cauchy hashing (DCH) model for binary hash code with the help of a pairwise cross-entropy loss based on Cauchy distribution [81]. Su et al. have employed the greedy hash by transmitting the gradient as intact during the backpropagation for hash coding layer which uses the sign function in forward propagation [82]. Thus, it maintains the discrete constraints, while avoiding the vanishing gradient problem. Yuan et al. (2018) have trained the network directly via policy gradient to maximize the reward expectation of similarity preservation using the generated binary codes [83]. A series expansion is used to treat the binary optimization of the hash function as the differentiable optimization which minimizes the objective discrepancy caused by relaxation [84]. Wu et al. (2018) have investigated a deep index-compatible hashing (DICH) method [85] by minimizing the number of similar bits between the binary codes of inter-class images.

3.2.4 2019

In 2019, a deep incremental hashing network (DIHN) is proposed by Wu et al. [127] to directly learn the hash codes corresponding

TABLE 2: A summarization of the state-of-the-art deep learning based approaches for image retrieval in terms of the different supervision mechanism, including supervised, un-supervised, semi-supervised, pseudo-supervised and self-supervised.

Type	Name	Year	Details
Supervised	CNN Hashing (CNNH) [86]	2014	CNN feature learning based hashing
	Supervised Deep Hashing (SDH) [87]	2015	Deep network as the feature extractor with last layer as latent vector
	Binary Hash Codes (BHC) [56]	2015	Learns hash code as CNN features in classification framework
	Deep Regularized Similarity Compar. Hash (DRSCH) [70]	2015	Bit-scalable hash codes with regularized similarity learning using triplet
	Network-In-Network Hashing (NINH) [69]	2015	Generates each bit from a mini-network with triplet loss
	Supervised Discrete Hashing (SDH) [88]	2015	Uses a discrete cyclic coordinate descent (DCC) algorithm
	Deep Hashing Network (DHN) [73]	2016	Jointly learns the feature and quantization using fully connected layer
	Deep Supervised Hashing (DSH) [89]	2016	Similarity-preserving binary code learning
	Very Deep Supervised Hashing (VDSH) [90]	2016	Deep neural networks
	Deep Pairwise-Supervised Hashing (DPSH) [91]	2016	Simultaneous feature and hash-code learning from pairwise labels
	Deep Triplet Supervised Hashing (DTSH) [92]	2016	Extension of DPSH, Triplet label based deep hashing
	Deep Quantization Network (DQN) [74]	2016	Controls the hashing quality with a product quantization loss
	Supervised Deep Hashing (SDH) [93]	2017	Extension of deep hashing with discriminative term and multi-label
	Supervised Semantics-preserving Deep Hash (SSDH) [94]	2017	Classification & retrieval are unified in a model for discriminativeness
	Deep Supervised Discrete Hashing (DSDH) [95]	2017	Uses pairwise label information and the classification information
	HashNet [77]	2017	Hashing by continuation method to learn binary codes
	GreedyHash [82] (also used in unsupervised mode)	2018	Iteratively updates towards a discrete solution in each iteration
	Deep Cauchy Hashing (DCH) [81]	2018	Bayesian learning over Cauchy cross-entropy and quantization losses
	Policy Gradient based Deep Hashing (PGDH) [83]	2018	Maximizes the rewards for similarity preservation in hash code
	GAN based Hashing (HashGAN) [96]	2018	Uses pair conditional wasserstein GAN to generate training images
	Deep Spherical Quantization (DSQ) [97]	2019	Utilizes the L2 normalization based multi-codebook quantization
	Deep Product Quantization (DPQ) [98]	2019	End-to-end learning of product quantization in a supervised manner
	Weighted Multi-Deep Ranking Hashing (WMDRH) [99]	2019	Uses multiple hash tables, ranking pairwise and classification loss
	Deep Hashing using Adaptive Loss (DHA) [100]	2019	Gradient saturation problem is tackled by shifting the loss function
	Just-Maximizing-Likelihood Hashing (JMLH) [100]	2019	Exploits the variational information bottleneck with classification
	Multi-Level Supervised Hashing (MLSH) [101]	2020	Integrates multi-level CNN features using a multiple-hash-table
Un-Supervised	Deep Hashing (DH) [87]	2015	Imposes quantization loss, balanced bits and independent bits
	Discriminative Attributes and Representations (DAR) [102]	2016	Uses clustering on CNN features
	DeepBit [103]	2016	Uses VGGNet architecture and rotation data augmentation
	Unsupervised Hashing Binary DNN (UH-BDNN) [104]	2016	Uses VGG features to learn the hash code in unsupervised manner
	Deep Descriptor with Multi-Quantization (BD-MQ) [105]	2017	Binarization in multiple steps to minimize the quantization loss
	Unsupervised Triplet Hashing (UTH) [106]	2017	Utilizes the quantization, discriminative and entropy loss
	Similarity Adaptive Deep Hashing (SADH) [107]	2018	Uses similarity graph
	Unsupervised Compact Binary Descriptors (UCBD) [108]	2018	Extension of DeepBit with more experiments
	GAN based Hashing (HashGAN) [109]	2018	Generative adversarial network trained in unsupervised manner
	Binary GAN (BGAN) [110]	2018	Binary generative adversarial network with VGG-F features
	Unsupervised ADversarial Hashing (UADH) [111]	2019	The pairs of hash codes are distinguished using discriminative network
	Unsupervised Deep Triplet Hashing (UDTH) [112]	2019	Hashing is performed using autoencoder and binary quantization
	DistillHash [113]	2019	Performs distilling based on the labels generated by the Bayes classifier
	Deep Variational Binaries (DVB) [114]	2019	Learns latent space using conditional variational Bayesian networks
Semi-Supervised	Semi-Supervised Deep Hashing (SSDH) [115]	2017	Jointly learns the embedding error on both labeled and unlabeled data
	Semi-Supervised GAN based Hashing (SSGAH) [116]	2018	Uses triplet-wise information in a semi-supervised way using GAN
	Semi-supervised Self-pace Adversarial Hash (SSAH) [117]	2019	Generates self-paced hard samples to increase the hashing difficulty
	Pairwise Teacher-Student Semi-Super. Hash (PTS3H) [118]	2019	Teacher network produces the pairwise info. to train the student network
Weakly-Supervised	Weakly-supervised Multimodal Hashing (WMH) [119]	2017	Utilizes the local discriminative and geometric structures in visual space
	Tag-based Weakly-supervised Hashing (TWH) [120]	2018	Weakly-supervised pre-training and supervised fine-tuning
	Weakly-supervised Deep Hashing with Tag (WDHT) [121]	2019	Utilizes the information from word2vec semantic embeddings
	Weakly-super. Semantic Guided Hashing (WSGH) [122]	2020	Exploits the binary matrix factorization to learn semantic information
Pseudo-Supervised	Pseudo Label based Deep Hashing (PLDH) [123]	2017	Creates the pseudo labels using K-means clustering
	Deep Self-Taught Graph-embedding Hash (DSTGeH) [124]	2020	Creates the pseudo labels using graph embedding based relationships
Self-Supervised	Self-Supervised Temporal Hashing (SSTH) [125]	2016	Utilizes the binary LSTM (BLSTM) to generate the binary codes
	Self-Supervised Adversarial Hashing (SSAH) [126]	2018	Exploits self-supervised adversarial learning for cross-modal hashing

to the new class coming images, while retaining the hash codes of existing class images. A supervised quantization technique developed for points representation on a unit hypersphere is used in deep spherical quantization (DSQ) model [97]. DistillHash method [113], introduced in 2019, automatically distills data pairs and learns deep hash functions from the distilled data set by employing the Bayesian learning framework. Bai et al. (2019) have developed a deep progressive hashing (DPH) model to generate a sequence of binary codes by utilizing the progressively expanded salient regions [128]. The recurrent deep network is used as the backbone in DPH model. An adaptive loss function based deep hashing model referred as DHA is proposed in [129] to generate the compact and discriminative binary codes. Shen et al. (2019) [100] have introduced a just-maximizing-likelihood hashing (JMLH) model by lower-bounding an information bottleneck between the images and its semantics. The deep variational binaries (DVB) are introduced by Shen et al. (2019) [114] as an unsupervised deep hashing model using conditional auto-encoding variational Bayesian networks.

3.2.5 2020

Recently, in 2020, Shen et al. have come up with a twin-bottleneck hashing (TBH) model between encoder and decoder networks [130]. They have employed the binary and continuous bottlenecks as the latent variables in a collaborative manner. The binary bottleneck uses a code-driven graph to encode the high-level intrinsic information for better hash code learning. Forcen et al.

(2020) have utilized the last convolution layer of CNN representation by modeling the co-occurrences from deep convolutional features [131]. A deep position-aware hashing (DPAH) model is proposed by Wang et al. in 2020 [132] which constraints the distance between data samples and class centers to improve the discriminative ability of the binary codes for image retrieval.

3.3 Summary

Following are the summary and findings from the above mentioned chronological overviews:

- The deep learning based methods have seen a huge progress for image retrieval in a decade from the basic neural network models to advanced neural network models.
- The existing methods can be categorized in different supervision modes, including supervised, unsupervised, etc.
- As the image retrieval application needs feature learning for matching, different type of networks has been utilized to do so, for example, CNN, Autoencoder, Siamese, GAN, etc. based networks.
- Various approaches focus over the binary descriptors/hashcodes for efficient retrieval, however, some methods also generate the real-valued description for higher performance.
- The choice of network and method is also dependent upon the retrieval type, such as object retrieval, semantic retrieval, sketch based retrieval, etc.

4 Different Supervision Categorization

This section is devoted to the discussion over the deep learning based image retrieval methods in terms of the different supervision types. Basically, supervised, unsupervised, semi-supervised, weakly-supervised, pseudo-supervised and self-supervised approaches are included. A high level and chronological overview of such techniques are presented in Table 4.

4.1 Supervised Approaches

The supervised deep learning models are used by researchers very heavily to learn the class specific and discriminative features for image retrieval. In 2014, Xia et al. have used a CNN to learn the representation of images which is used to generate a hash code H and class labels [86]. They have also imposed a criteria as $HH^T = I$, where I is the original image. The promising performance is reported over MNIST, CIFAR-10 and NUS-WIDE datasets. Shen et al. (2015) [88] have proposed the supervised discrete hashing (SDH) based generation of image description with the help of the discrete cyclic coordinate descent for retrieval. Liu et al. (2016) have done the revolutionary work in this area and introduced a deep supervised hashing (DSH) method to learn the binary codes from the similar/dissimilar pairs of images [89]. The DSH imposes the regularization on the real-valued outputs to approximate the desired binary bits. Li et al. have also performed the similar work and proposed a deep pairwise-supervised hashing (DPSH) method for image retrieval [91]. However, the convolutional network model is used in [91] as compared to the multilayer perceptron in [89]. The pair-wise labels are extended to the triplet labels (i.e., query, positive and negative images) by Wang et al. (2016) to train a shared deep CNN model for feature learning [92]. Zhang et al. have utilized the auxiliary variables based independent layer-wise local updates to efficiently train a very deep supervised hashing (VDSH) model to learn the discriminative hash codes for image retrieval [90].

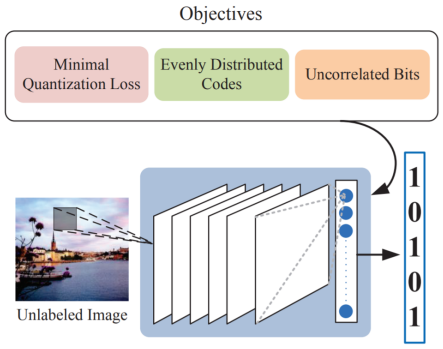


Fig. 7: An illustration of convolutional neural network (CNN) based unsupervised feature learning for image retrieval. This figure is originally shown in DeepBit work [103], [108]. These approaches generally use the different constraints on the abstract features to train the models.

In 2017, Li et al. have used the classification information and the pairwise label information in a single framework for the learning of the deep supervised discrete hashing (DSDH) codes [95]. The DSDH makes the outputs of the last layer to be binary codes directly. Yang et al. (2017) have developed the supervised semantics-preserving deep hashing (SSDH) model by considering the hash functions as a latent layer in addition to the binary codes which are learnt in classification framework [94]. Thus, the SSDH enjoys the integration of retrieval and classification characteristics. Wu et al. have tried to resolve the problem of not keeping the direct constraints on dissimilarity between the descriptors of similar images of triplets [133]. The scalable image search is performed by Lu et al. [93] in 2017 by introducing the following three characteristics: 1) minimizing the loss between the real-valued code and equivalent converted binary code, 2) ensuring the even distribution among each bit in the binary codes, and 3) decreasing the redundancy of a bit in the binary code. The supervised learning is used to increase the discriminating ability of the learnt features.

The supervised training has been also the choice in asymmetric hashing [134]. A deep product quantization (DPQ) model is followed in supervised learning mode for image search and retrieval by Klein et al. (2019) [98]. The supervised deep feature embedding is also used with the hand crafted features [135]. A very recently, a multi-Level hashing of deep features is performed by Ng et al. (2020) [101]. An angular hashing loss function is also used to train the network in the supervised fashion by angular deep supervised hashing (ADSH) method for generating the hash code [136]. A supervised hashing is also used for the multi-deep ranking [99] to improve the retrieval efficiency. Some other supervised approaches are deep binary hash codes [56], deep hashing network [73], deep spherical quantization [97], adaptive loss based supervised deep learning to hash [129], etc.

4.2 Unsupervised Approaches

Though the supervised models have shown promising performance for image retrieval, it is difficult to get the labelled large-scale data always. Thus, several unsupervised models have been also investigated which do not require the class labels to learn the features.

The unsupervised models generally enforce the constraints on hash code and/or generated output to learn the features.

Erin et al. (2015) [87] have used the deep networks in an unsupervised manner to learn the hash code with the help of the constraints like quantization loss, balanced bits and independent bits. Huang et al. (2016) [102] have utilized the CNN coupled with unsupervised discriminative clustering to learn the description in an unsupervised manner. In 2015, Paulin et al. have used an unsupervised convolutional kernel network (CKN) based method for the learning of convolutional features for the image retrieval [137]. They have also applied it to patch retrieval. In an outstanding work, Lin et al. (2016) have imposed the constraints like minimal quantization loss, evenly distributed codes and uncorrelated bits to design an unsupervised deep network based DeepBit model for image retrieval, image matching and object recognition applications [103] as depicted in Fig. 7. A two stage training is performed for DeepBit. In the first stage, the model is trained with respect to above mentioned objectives. Whereas, in order to improve the robustness of DeepBit, a rotation data augmentation based fine tuning is performed in the second stage. The detailed analysis of DeepBit is illustrated in the extended work [108]. However, the DeepBit model suffers with the severe quantization loss due to the rigid binarization of data using sign function without considering its distribution property. In order to tackle the quantization problem of DeepBit, a deep binary descriptor with multiquantization (DBD-MQ) is introduced by Duan et al. [105] in 2017. It is achieved by jointly learning the parameters and the binarization functions using a K-AutoEncoders (KAEs) network.

It is observed by Radenovic et al. [138] that unsupervised CNN can learn more distinctive features if fine tuned with hard positive and hard negative examples. A stacked restricted boltzmann machines (SRBM) based deep neural network is also used to generate the low dimensional features which is fine tuned further to generate the descriptor [139]. Paulin et al. (2017) have worked upon the patch representation and retrieval by developing a patch convolutional kernel network (Patch-CKN) [140]. An anchor image, a rotated image and a random image based triplets are used in unsupervised triplet hashing (UTH) network to learn the binary codes for image retrieval [106]. The UTH objective function uses the combination of discriminative loss, quantization loss and entropy loss. In 2018, an unsupervised similarity-adaptive deep hashing (SADH) model is proposed by Shen et al. [107] by employing the training of the deep hash model, updating the similarity graph and optimizing the binary codes. Xu et al. (2018) [141] have extended the deep CNN layers as part-based detectors by employing its discriminating filters and proposes a semantic-aware part weighted aggregation (PWA) for CBIR systems. The PWA uses an unsupervised way of part selection to suppress the background noise. Unsupervised generative adversarial networks [109], [110], [111] are also investigated for image retrieval. The distill data pairs [113] and deep variational networks [114] are also used for unsupervised image retrieval. The pseudo triplets based unsupervised deep triplet hashing (UDTH) technique [112] is introduced for scalable image retrieval. Very recently unsupervised deep transfer learning has been exploited by Liu et al. (2020) [142] for image retrieval in remote sensing images.

4.3 Semi-supervised Approaches

The semi-supervised approaches generally use a combination of labelled and un-labelled data for feature learning. In 2017, Zhang and Peng [115] have proposed a semi-supervised deep hashing (SSDH) framework for image retrieval from labeled and unlabeled data. The SSDH uses labeled data for the empirical error minimization and both labeled and unlabeled data for embedding error minimization. The generative adversarial learning has been also utilized extensively in semi-supervised deep image retrieval [116], [117]. A teacher-student framework based semi-supervised image retrieval is performed by Zhang et al. (2019) [118] in which the pairwise information learnt by the teacher network is used as the guidance to train the student network.

4.4 Weakly-supervised Approaches

Weakly-supervised approaches have been also explored for the image retrieval task [119], [120], [121], [122]. For example, Tang et al. (2017) have put forward a weakly-supervised multimodal hashing (WMH) by utilizing the local discriminative and geometric structures in the visual space [119]. Guan et al. (2018) [120] have performed the pre-training in weakly-supervised mode and fine-tuning in supervised mode. Gattupalli et al. (2019) [121] have developed the weakly supervised deep hashing using tag embeddings (WDHT) for image retrieval. The WDHT utilizes the word2vec semantic embeddings. Li et al. (2020) [122] have developed a semantic guided hashing (SGH) network for image retrieval by simultaneously employing the weakly-supervised tag information and the inherent data relations.

4.5 Pseudo-supervised Approaches

The pseudo suervised networks have been also developed for image retrieval [123], [112], [124]. A k-means clustering based pseudo labels are generated from the pretrained VGG16 features and used for the training of a deep hashing network with classification loss and quantization loss as the objective functions [123]. An appealing performance has been observed using pseudo labels over CIFAR-10 and Flickr datasets for image retrieval. The pseudo triplets are utilized in [112] for unsupervised image retrieval. Recently, in 2020, pseudo labels are used for deep self-taught graph embedding based hash codes (DSTGeH) [124] for image retrieval.

4.6 Self-supervised Approaches

The self-supervision is another way of supervision used in some research works for image retrieval [125], [126]. For example, Zhang et al. (2016) [125] have introduced a self-supervised temporal hashing (SSTH) for video retrieval. Li et al. (2018) [126] have used the adversarial networks in self-supervision mode for cross-image retrieval by utilizing the multi-label annotations.

4.7 Summary

Following are the summary and take away points from the above discussion on deep learning based image models from the supervision perspective:

- The supervised approaches utilize the class-specific semantic information through the classification error apart from the other objectives related to the hash code generation. Generally, the performance of supervised models is better than other models due to learning of the fine-grained and class specific information. Different type of networks can be exploited for retrieval with classification error.

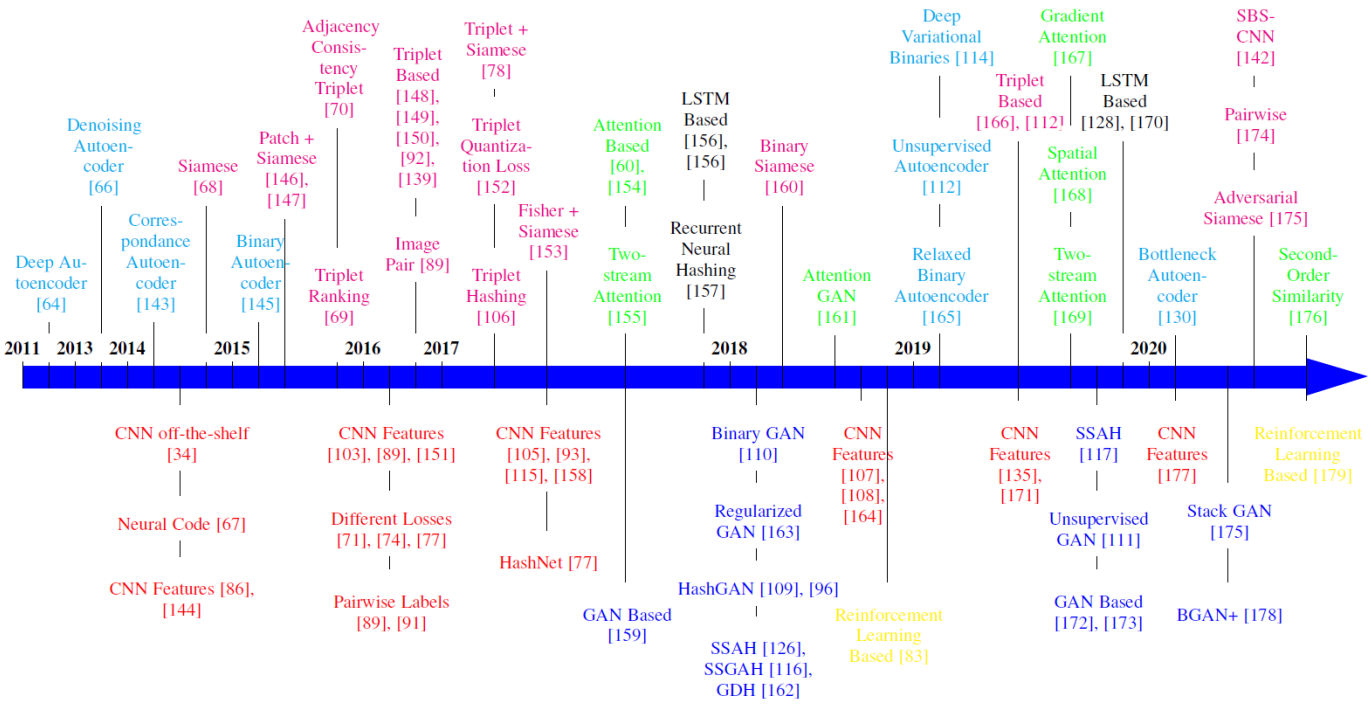


Fig. 8: A chronological view of deep learning based image retrieval methods depicting the different type of neural networks used to learn the features from 2011 to 2020. The convolutional neural network, autoencoder network, siamese & triplet network, recurrent neural network, generative adversarial network, attention network and reinforcement learning network based deep learning approches for image retrieval are depicted in Red, Cyan, Magenta, Black, Blue, Green, and Yellow colors, respectively.

- The unsupervised models make use of the unsupervised constraints on hash code (i.e., quantization loss, independent bits, etc.) and/or data reconstruction (i.e., using an autoencoder type of networks) to learn the features. Different networks such as autoencoder networks, generative adversarial networks, etc. can be used to learn the features in unsupervised mode.
- The semi-supervised approaches exploit the labelled and un-labelled data for the feature learning using deep networks. The weakly-supervised approaches generally utilize the information from different modalities using different networks.
- The pseudo-supervised approaches generate the pseudo labels using some other methods to facilitate the training using generated labels. The self-supervised methods generate the temporal or generative information to learn the models over the training epochs.
- The minimal quantization error, independent bits, low dimensional feature, discriminative code, etc. are the common objectives for most of the image retrieval methods.

5 Network Types For Image Retrieval

In this section, the progress in a decade is presented for deep learning based image retrieval approaches in terms of the different deep learning architectures. The convolutional neural network, autoencoder network, siamese & triplet network, recurrent neural network, generative adversarial network, attention network and reinforcement learning network are included in this paper. A chronological overview from 2011 to 2020 is illustrated in Fig. 8 for different type of networks for image retrieval.

5.1 Convolutional Neural Networks for Image Retrieval

The convolutional neural networks (CNN) based feature learning has been utilized extensively for image retrieval. Some typical examples of CNN based image retrieval are shown in Fig. 6, and 7. The CNN consists of different layers, including convolution, non-linearity, batch normalization, dropout, fully connected layers, etc. Generally, the abstract features learnt through the late fully connected layers are used to generate the hash code and descriptor.

In 2014, the experimental analysis of CNN features off-the-shelf have shown a tremendous performance gain for image recognition and retrieval as compared to the hand-crafted features [34]. At the same time the activations of trained CNN has been also explored as the neural code for retrieval [67]. An image representation learning has been also performed using the CNN model to generate the descriptor for image retrieval [86]. In 2016, pairwise labels are exploited to learn the CNN feature for image retrieval [89], [91]. The CNN activations are heavily used to generate the hash codes for efficient image retrieval by employing the different losses [71], [74], [77]. The abstract features of CNN are learnt for the image retrieval in different modes, such as unsupervised image retrieval [103], [105], [107], [108], supervised image retrieval [86], [89], [93], [135], semi-supervised image retrieval [115], cross-modal retrieval [143], [144], sketch based image retrieval [145], [146], object retrieval [147], [148], etc.

5.2 Autoencoder Networks based Image Retrieval

Autoencoder (AE) is a type of unsupervised neural network [149], [150] which can be used to reconstruct the input image from the latent space. A simple Autoencoder network is portrayed in Fig. 9. Basically, it consists of two networks, namely encoder (En) and decoder (De). The encoder network transforms the input

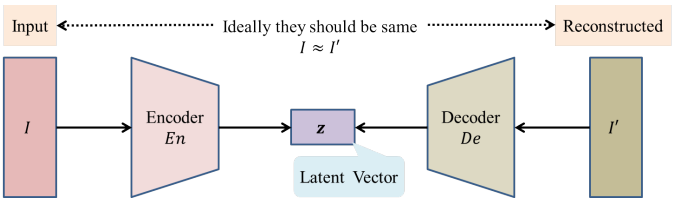


Fig. 9: A typical Autoencoder network consisting of an Encoder network and a Decoder network. Generally, the encoder is a CNN and the decoder is an up-CNN. The output of the encoder is a latent space which is used to generate the hash codes.

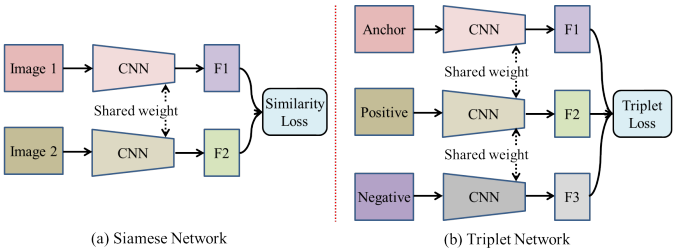


Fig. 10: (a) Siamese network computes the similarity between image pairs. (b) Triplet network minimizes the distance between the anchor and positive and maximizes the distance between the anchor and negative in feature space.

(I) into latent feature space (z) as $En : I \rightarrow z$. Whereas, the decoder network tries to reconstruct the original image (I') from latent feature space as $De : z \rightarrow I'$. The model is trained by minimizing the reconstruction error between original image (I) and reconstructed image (I') using L_1 or L_2 loss function.

The autoencoder based neural networks have also been very intensively used to learn the features as the latent space for image retrieval. In the initial attempts, the deep autoencoder was used for image retrieval in 2011 [64]. A stacked denoising autoencoder is used to train the multiple deep neural networks by Wu et al. (2013) for online multimodal deep similarity learning (OMDSL) for retrieval task [66]. Feng et al. (2014) have utilized the correspondence autoencoder (Corr-AE) for cross-modal retrieval [151]. The Corr-AE captures the correlation between the latent description of two uni-modal autoencoders. In 2015, the binary autoencoder model is used to learn the binary code for fast image retrieval by reconstructing the image from that binary code function [152].

In 2019, a deep variational binaries (DVB) is introduced by Shen et al. by auto-encoding the variational Bayesian networks to learn the latent features [114]. It uses VGG16 as the base network. Gu et al. (2019) have utilized the autoencoder over the triplet in an unsupervised manner to learn the hash code for retrieval [112]. Do et al. (2019) have utilized a relaxed binary autoencoder (RBA) to learn the image description for retrieval along with feature aggregation [153]. In a recent work, Shen et al. (2020) [130] have used the double latent bottlenecks in autoencoder including binary latent variable and continuous latent variable. The latent variable bottleneck exchanges crucial information collaboratively and the binary codes bottleneck uses a code-driven graph to capture the intrinsic data structure.

5.3 Siamese and Triplet Networks for Image Retrieval

5.3.1 Siamese Network

The siamese is type of neural network [154] that aims to minimize the distance between features of similar images and maximize the distance between features of dissimilar images as depicted in Fig. 10(a). The siamese network based learnt features have shown very promising performance for fine-grained image retrieval [68]. A pair of similar or dissimilar images is jointly processed by Liu et al. [89] to produce 1 or -1 output by CNN to learn the feature for image retrieval. Ong et al. (2017) have used the fisher vector computed on top of the CNN feature in autoencoder network to generate the discriminating feature descriptor for image retrieval [155]. In 2018, the siamese network is also used to learn the binary codes using two parallel networks with shared weights for efficient image retrieval [156]. Recently in 2020, the a lightweight similarity-based siamese CNN (SBS-CNN) model is used for remote sensing image retrieval [142]. The SBS-CNN computes the similarity between the features using a multilayer perceptron subnetwork. A pairwise similarity-preserving quantization loss is employed in [157] for learning the features for semantic image retrieval. Pandey et al. (2020) have utilized the siamese network with the stacked adversarial network for zero-shot sketch based image retrieval [158]. The siamese network is extensively used in computer vision such as computing the similarity between the patches for image matching [159], [160].

5.3.2 Triplet Network

A triplet network is a variation of siamese network [161] which utilizes a triplet of images, including an anchor image, a positive image and a negative image. The triplet network minimizes the distance between the features of anchor and positive image and maximizes the distance between the features of anchor and negative image, simultaneously. A typical structure of triplet network is shown in Fig. 10(b). In 2015, a triplet ranking loss function is utilized on top of the shared CNN features to learn the network for computation of binary descriptors for image retrieval [69]. An adjacency consistency based regularization term is introduced by Zhang et al. (2015) in the triplet network to enforce the discriminative ability of the CNN feature description [70]. Zhuang et al. (2016) [162] have used triplet to learn the hash code from the VGG16 abstract features by employing the relation weights matrix and graph cuts optimization for image retrieval. The triplet ranking loss, orthogonality constraint and softmax loss are minimized jointly by Yao et al. (2016) for semantic image retrieval [163]. In 2016, the siamese network based triplet network is introduced to learn the feature descriptor for image retrieval [164]. Similarly, a triplet based siamese network is used by Gordo et al. (2017) [78] and a triplet quantization loss is used by Zhou et al. (2017) [165] to learn the feature descriptor for image retrieval. A deep triplet quantization (DTQ) is performed in [166] for image retrieval. The triplet based feature learning has been also exploited for sketch based image retrieval [167]. Wang et al. (2016) [92] have used the supervised hashing with the help of triplet labels. The unsupervised triplet hashing is also done by Lin et al. (2016) [139], Huang et al. (2017) [106] and Gu et al. (2019) [112] for image retrieval.

5.4 Generative Adversarial Networks based Retrieval

The generative adversarial network (GAN) was developed by Goodfellow et al. [168] in 2014 by employing the two networks,

i.e., generator network and discriminator network. The generator network tries to generate the new samples in the training set from the random vector. Whereas, the discriminator network tries to distinguish between generated image and original image. Thus, the GAN model is trained like a mini-max game where generator wants to fool discriminator by producing more realistic images and discriminator wants not to get fooled due to generator by learning to distinguish between the generated and real images.

In 2017, Wang et al. have generated the common subspace based on adversarial learning for cross-modal retrieval [169]. They use the generator to generate the modality-invariant representation and the discriminator to distinguish between different modalities. In an other work attention based adversarial hashing network is developed for cross-modal retrieval [170]. In 2018, Song et al. have introduced a binary generative adversarial network (BGAN) for generating the representational binary codes for image retrieval [110]. The BGAN network is trained in an unsupervised manner. At the same time, a regularized GAN is used to introduce the BinGAN model [171] to learn the compact binary patterns. The BinGAN uses two regularizers, including a distance matching regularizer and a binarization representation entropy (BRE) regularizer. In 2018, the generative networks are also utilized by Dizaji et al. [109] to develop HashGAN in an unsupervised manner to generate the hash code for image retrieval. At the same time another HashGAN is developed by Cao et al. by employing the paired conditional Wasserstein GAN to generate more samples for learning the hash code for image retrieval [96]. A very supporting results have been observed using HashGAN over NUS-WIDE, CIFAR-10 and MS-COCO datasets. Multiple adversarial networks are also used in self-supervised adversarial hashing (SSAH) [126] to reduce the gap between the presentation of different modalities.

A semi-supervised generative adversarial hashing (SSGAH) model is presented in [116]. The SSGAH learns the distribution of triplet-wise information with the help of the generative as well as discriminative models. A semi-supervised self-pace adversarial hashing (SSAH) method is discovered in [117] by integrating an adversarial network (ANet) with a hashing network (H-Net). The GAN based approach is employed to develop a generative domain-migration hashing (GDH) model to bridge the gap between the sketch and image domains for sketch based image retrieval [172]. A generative model that uses the new class sketch as the condition to generate the images is developed by Verma et al. (2019) [173] for zero-shot sketch based image retrieval. The performance improvement has been observed over Sketchy and TU Berlin datasets in [173]. Gu et al. (2019) have also used adversarial learning for cross-modal retrieval [174]. Very recently, Pandey et al. (2020) [158] have also performed the zero-shot sketch based image retrieval with the help of a stack of adversarial networks by generating the better samples as well as a siamese network by generating a better distance metric. In an another attempt Deng et al. (2019) [111] have also explored the uses of unsupervised adversarial hashing for image retrieval. In 2020, Song et al. [175] have developed a binary generative adversarial networks based unified BGAN+ framework for image retrieval and compression. Basically, the BGAN+ learns one binary representation for image retrieval and another for image compression, simultaneously.

5.5 Attention Networks for Image Retrieval

The utilization of attention has been observed as a very effective way of modelling the saliency information into the feature space to avoid the effect of background noise. In 2017, Noh et al. have used the attention-based keypoints to select the important DEep Local Features (DELF) [60]. The DELF performs the feature selection based on the high scores assigned to relevant features by the attention module sitting on top of the convolutional features. Yang et al. (2017) have introduced a two-stream attentive CNNs by fusing a Main and an Auxiliary CNN (MAC) for image retrieval [176]. The main CNN in MAC focuses over the discriminative visual features for semantic information, whereas the auxiliary CNN focuses over the part of features for attentive information. The detailed analysis of two-stream based attentive MAC network is presented by Wei et al. (2019) in [177]. It uses VGG16 as the main network and DeepFixNet as the auxiliary network.

Two sub-networks are also employed by Ge et al. (2019) [178] with one sub-network to model the spatial attention and another sub-network to extract the global features. Finally, the features from last fully connected layers of both the sub-networks are fused to generate the final descriptor. Recently, Ng et al. in 2020 [179] have computed the second-order similarity (SOS) loss over the selected regions of the input image for image retrieval. The important regions are identified using the attention module learnt automatically from the data. The attention aware hashing in generative framework is used in [170] for cross-modal retrieval. Song et al. [180] have utilized the attention module for fine-grained sketch-based image retrieval. In an extension of attention concept, Huang et al. (2019) [181] have proposed the gradient attention network for deep hashing based image retrieval. Basically, it enforces the CNN binary features of a pair to minimize the distances between them, irrespective of their signs or directions.

5.6 Recurrent Neural Networks for Image Retrieval

The literature has also been witnessed with the exploitation of recurrent neural network (RNN) and long short-term memory (LSTM) to learn the image description for image search. In 2018, Lu et al. have utilized the RNN concept to perform a hierarchical recurrent neural hashing (HRNH) to produce the effective hash codes for image retrieval [182]. In 2017, Shen et al. have used the region-based convolutional networks with LSTM modules for textual-visual cross retrieval [183]. In order to improve the training convergence, a stochastic batch-wise code learning routine is adapted in [183]. Bai et al. (2019) have also employed the LSTM based recurrent deep network in the triplet hashing framework to naturally inherit the useful information for image retrieval [128]. Chen et al. (2019) [184] have used the LSTM module as a bottleneck between the convolution and fully-connected blocks in a siamese network framework to learn the discriminative description.

5.7 Reinforcement Learning Networks based Retrieval

Few works have been also reported in the literature by utilizing the concept of reinforcement learning for image retrieval. In 2018, Yuan et al. have exploited the reinforcement learning for image retrieval [83]. They have used a relaxation free method through policy gradient to generate the hash codes for image retrieval. The similarity preservation via the generated binary codes is used as the reward function. In 2020, recently, Yang et al. [185] have utilized the deep reinforcement learning to perform the de-redundancy in hash bits to get rid of redundant and/or harmful bits, which reduces the ambiguity in the similarity computation for image retrieval.

5.8 Summary

The summary of the different network driven deep learning based image retrieval approaches is as follows:

- The convolutional neural network features are exploited for the hash code and descriptor learning by employing the various constraints like classification error, quantization error, independent bits, etc.
- In order to make the features more representative of the image, the autoencoder networks are used which enforces the learning based on the reconstruction loss.
- The discriminative power of descriptive hash code is enhanced by exploiting the siamese and triplet networks. Different constraints are used on the hash code to make it discriminative and compact.
- The generative adversarial network based approaches have been highly utilized to improve the discriminative ability and robustness of the learnt features by encoder network guided through the discriminator network.
- The automatic important feature selection is performed using attention module to control the redundancy in the feature space. The recurrent neural network and reinforcement learning network have been also shown very effective for the image retrieval.

6 Type of Descriptors for Image Retrieval

The literature has witnessed with binary hash codes for efficient image retrieval, real-valued descriptors and feature aggregation for discriminative image retrieval as depicted in Fig. 11 using chronological overview.

6.1 Binary Descriptors

Most of the methods focus over the learning of the binary descriptors in terms of the hash codes. Different types of networks are used to learn the binary description such as deep neural networks [87], convolutional neural networks [56], autoencoder networks [64], [152], siamese networks [156], triplet networks [112], generative adversarial networks [110], [171], [175], and deep variational networks [114].

In 2015, Liong et al. [87] have introduced a supervised deep hashing (SDH) approach by using a deep neural network to produce the binary hash codes. The SDH method uses three constraints, including the quantization loss, balanced bits and independent bits. Lin et al. (2015) have used a CNN to produce the binary hash code through a latent layer in a supervised manner [56] and demonstrated outstanding retrieval precision over MNIST and CIFAR-10 datasets. A binary autoencoder consisting of an encoder and a decoder is also used in 2011 [64] and 2015 [152] to learn the binary features for efficient image retrieval. In 2017, a siamese neural network is utilized by Jose et al. [156] to learn the binary representation for image retrieval application.

In 2016, Do et al. have proposed a binary deep neural network (BDNN) by converting a hidden layer output to binary code [104]. The BDNN is tested in both supervised and unsupervised frameworks. The extended analysis of this work is further presented in [186]. Do et al. have also employed a network to learn the binary code jointly with feature aggregation [153]. In 2019, Do et al. have applied a masking technique using different masks, such as SIFT-mask, SUM-mask, and MAX-mask, over the convolutional features to generate the aggregate features used to produce the binary description for image retrieval [187]. A ranking optimization discrete hashing (RODH) approach is used by Lu et al. (2019) [188] by generating the discrete hash codes as either +1 or -1 for image retrieval by employing the ranking information. The binary hash code learnt using deep learning is also used for clothing image retrieval [189]. A cauchy quantization loss is used by Cao et al. (2018) [81] to improve the discriminative power of binary descriptors for image retrieval. Su et al. (2018) [82] have used an iterative quantization approach to convert the features into binary codes to avoid the quantization loss.

The binary description is learnt through the supervised [88], [89], [95], unsupervised [103], [108], [109] and self-supervised [125] deep learning techniques. The deep supervised hashing [89] and deep supervised discrete hashing [88], [95] approaches generate the supervised binary hash code for image retrieval. Among unsupervised binary hashing approaches, Lin et al. (2016) [103] have utilized the minimal quantization loss, evenly distributed codes and uncorrelated bits properties to generate the discriminative and compact code for image retrieval. Lin et al. (2018) [108] have added the transformation invariant bit apart from the minimal quantization loss and evenly distributed codes constraints to increase the robustness of the binary descriptors for retrieval. Among the generative approaches, a binary generative adversarial network (BGAN) is used to learn the binary code in 2018 [110]. At the same time a regularized GAN is used by maximizing the entropy of binarized layer for image retrieval [171]. The GAN is trained in unsupervised mode by Ghasedi et al. (2018) [109] to learn the binary codes for image retrieval. In 2020, the binary GAN [175] is used for image retrieval and compression jointly.

6.2 Real-Valued Descriptors

The hashing has shown the promising performance for large-scale image retrieval. Most of the hashing approaches generate the binary codes. However, the binary hashing approaches have the obvious shortcomings. First, it is difficult to represent the fine-grained similarity using binary code. Second, the generation of similar binary codes is common even for different images. Thus, researchers have also used the real-valued features to represent the images for the retrieval task.

In 2016, Kumar et al. [164] have used a deep siamese and triplet convolutional networks to learn the non-binary description for image retrieval. Ong et al. (2017) [155] have also used siamese Network to generate the deep fisher-vector descriptors for image retrieval. Gordo et al. (2017) [78] have used a siamese architecture to learn the real-valued code using a deep network for image retrieval. In 2018, a non-binary hash code based image retrieval system is developed by Xu et al. (2018) [141] by aggregating the part-based CNN features. The real-valued descriptors generated using CNNs are used for medical image retrieval as well [190], [191]. Ji et al. (2019) [192] have used the real-valued descriptors for cross-modal retrieval. Chen et al. (2020) [193] have developed a pairwise correlation discrete hashing (PCDH) by exploiting the pairwise correlation of deep features for image retrieval. Shen et al. (2020) [130] have also used the real-valued descriptors with the help of double bottleneck hashing approach for retrieval.

6.3 Aggregation of Descriptors

Several researchers have also tried to combine/fuse the feature at different stages of the network or multiple networks to generate the

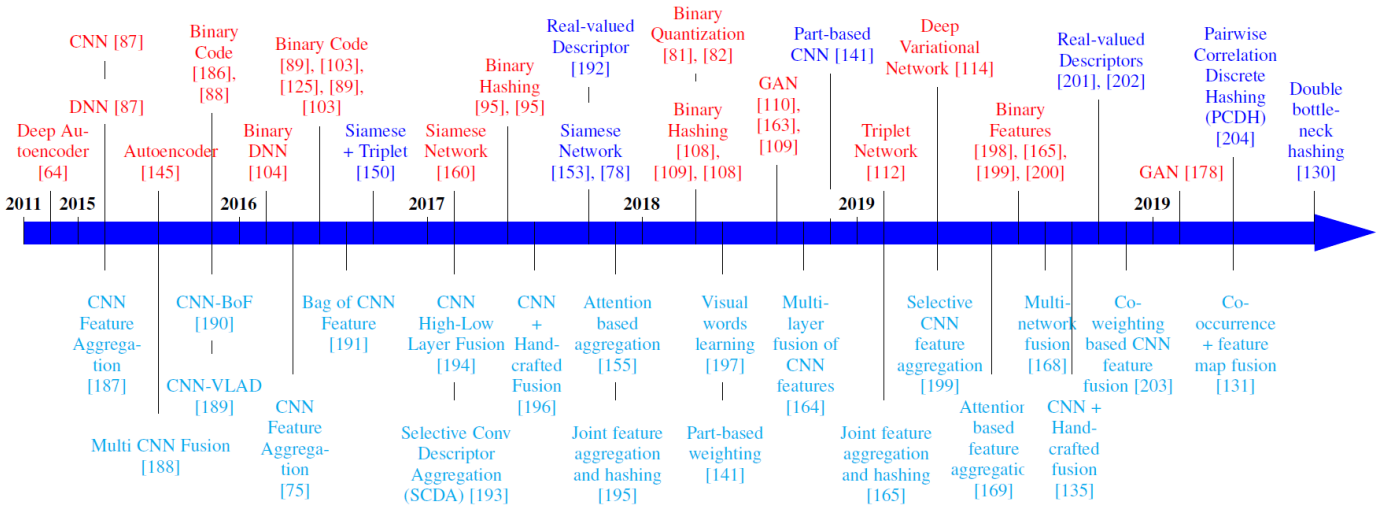


Fig. 11: A chronological view of deep learning based image retrieval methods depicting the different type of descriptors. The binary and real-valued feature vector based models are presented in Red and Blue colors, respectively. The feature aggregation based models are presented in Cyan color.

aggregation of descriptors for image retrieval. In 2015, Babenko et al. have applied the aggregation of deep convolutional features to form the discriminative descriptors for image retrieval [194]. They have identified that the sum-pooling based aggregation is more suitable for CNN features. Similarly, Husain et al. (2016) [75] have also aggregated the CNN features for image retrieval. Liu et al. (2015) [195] have integrated the mid-level deep feature from one CNN and high-level deep feature from another CNN to generate more discriminative descriptor for image retrieval. At the same time, fisher encoded convolutional bag-of-features are also used for image retrieval [196]. The vector locally aggregated descriptors (VLAD) encoding is applied by Yue et al. (2015) [197] on the features extracted from different layers. In 2016, a bag of local convolutional features is used by Mohedano et al. [198] for scalable instance search. Wei et al. (2017) [199] have proposed a selective convolutional descriptor aggregation (SCDA) to generate the descriptor for image retrieval. The SCDA firstly localizes the fine-grained regions and then aggregate its CNN features. The higher-layer features are fused with the lower-layer features by Yu et al. (2017) [200] to make the descriptor rich of details and abstract information. Xu et al. (2018) [141] have also performed the part-based weighting aggregation of deep convolutional features and trained in an unsupervised manner for image retrieval. Liu et al. (2019) [201] have introduced an end- to-end BoWs model using deep CNN by exploiting the visual words learning for image retrieval.

In 2017, Do et al. have performed the joint training of feature aggregation and hashing simultaneously [202]. Yu et al. (2018) have employed the multi-layer fusion of deep CNN features for sketch based image retrieval [145]. In 2019, Do et al. have introduced a simultaneous feature aggregating and supervised hashing (SASH) [153] approach that learns the feature aggregation and hash function in a joint manner for image retrieval. Do et al. have also aggregated the masked features for hash code learning in [187]. The VGG16 network is aggregated with a DeepFixNet network to model the attention mechanism in [176] and [177]. One main sub-network and other attention-based sub-network are also fused at the last fully connected layer in [178]. In 2019, Kan et al. [135] have integrated the hand crafted features with deep feature embeddings and trained in a supervised manner for image retrieval. The hand-designed features are fused with CNN in [203] also. A co-weighting based CNN feature fusion is applied by Zhu et al. (2019) [204] for object retrieval. Recently, Forcen et al. (2020) [131] have generated the image representation by combining a co-occurrence map with the feature map for image retrieval.

6.4 Summary

The followings are the summary of deep learning based approaches from the perspective of the type of feature descriptor:

- In order to facilitate the large-scale image retrieval, the compact and binary hash codes are generated using different networks. Different methods try to improve the discriminative ability, lower redundancy among bits, generalization of the binary hash code, etc. in different supervision modes.
- The real-valued descriptors concentrate over the discriminative ability of the learnt features for image retrieval at the cost of increased computational complexity for feature matching. Such approaches also try to increase the robustness and reduce the dimensionality of the descriptors.
- Feature aggregation approaches try to utilize the complementary information between the features of different networks, the features of different sub-network, and the features of different layers of same network to improve the image retrieval performance.

7 Retrieval Type

Various retrieval types have been explored using deep learning approaches based on the nature of the problem and data. In this section, a detailed overview and survey is presented for different retrieval type, including cross-modal retrieval, sketch based retrieval, multi-label retrieval, instance retrieval, object retrieval, semantic retrieval, fine-grained retrieval and asymmetric retrieval.

7.1 Cross-modal Retrieval

The cross-modal retrieval refers to the image retrieval involving more than one modality by measuring the similarity between heterogeneous data objects. The extensive work has been conducted in recent years by utilizing the deep learning approaches for cross-modal retrieval. Feng et al. (2014) have introduced a correspondence autoencoder (Corr-AE) network for cross-modal retrieval [151]. In 2016 [205], a deep visual-semantic hashing (DVSH) network is developed for sentence and image based cross-modal retrieval. The DVSH jointly learns the embeddings for images and sentences using a hybrid deep model. Shen et al. (2017) have proposed a textual-visual deep binaries (TVDB) based model to represent the long descriptive sentences along with its corresponding informative images [183]. In another work, the pairwise constraints are incorporated to learn the deep hashing network for cross-modal retrieval [144]. The CNN visual features have been also exploited for cross-modal retrieval. For example, in 2017, Wei et al. [143] have utilized the CNN off-the-shelf features for labeled annotation based cross-modal retrieval. The CNN features are used in online learning mode with bi-directional hinge loss for cross-modal retrieval by Wu et al. (2017) [206]. The adversarial neural network is also employed for cross-modal retrieval such as adversarial cross-modal retrieval (ACMR) [169], self-supervised adversarial hashing (SSAH) [126], attention-aware deep adversarial hashing (ADAH) [170], adversary guided asymmetric hashing (AGAH) [174], etc. A deep multi-level semantic hashing (DMSH) [192] is developed in 2019 for cross-modal retrieval.

7.2 Sketch Based Image Retrieval

Sketch based image retrieval (SBIR) is a special case of cross-modal retrieval where the query image is in the sketch domain the retrieval has to be performed in the image domain. With the advancements in the deep learning approaches in the last decade, several deep learning models have been also investigated for sketch based image retrieval.

In 2017, Song et al. [180] have developed a fine-grained sketch-based image retrieval (FG-SBIR) model with the help of attention module and higher-order learnable energy function loss. Liu et al. (2017) [207] have introduced a semi-heterogeneous deep sketch hashing (DSH) model for sketch based image retrieval by utilizing the representation of free-hand sketches. The sketches and natural photos are mapped in multiple layers in a deep CNN framework by Yu et al. (2018) [145] for sketch based image retrieval. A zero-shot sketch-based image retrieval (ZS-SBIR) is proposed for retrieval of photos from unseen categories [167]. Wang et al. (2019) [146] have proposed a CNN based SBIR re-ranking approach to refine the retrieval results. Recently, the sketch augmentation has been used by Zhou et al. (2020) [208] for shape retrieval.

The generative adversarial networks have been also exploited extensively for SBIR. For example, a generative domain-migration hashing (GDH) model is developed using cycle consistency loss for SBIR [172]. A new class sketch conditioned generative model is also introduced for zero-shot SBIR [173]. A semantically aligned paired cycle-consistent generative (SEM-PCYC) model is developed by Dutta and Akata (2019) for zero-shot SBIR [209]. A stacked adversarial network (SAN) [158] is the recent attempt in the development of the zero-shot SBIR model.

7.3 Multi-label Image Retrieval

Multi-label retrieval involves multiple categorical labels while generating the image representations for image retrieval. In 2015, Zhao et al. have utilized the multilevel similarity information using deep semantic ranking for image retrieval [210]. Lai et al. (2016) [211] have extended the multi-label hashing by incorporating the object instances in the representation. A deep multilevel semantic similarity preserving hashing (DMSSPH) [133] method is used by Wu et al. for multi-label image retrieval. Multi-label annotations are also utilized for cross-modal retrieval [126]. Deep network based Multi-label search is also applied for video search [184]. Multiple category-aware object based deep hashing is performed in [212] for multi-label image retrieval. Recently, Qin et al. (2020) [213] have utilized the fine-grained features of a deep multilevel similarity hashing for multi-label Image Retrieval. Readers may refer to the survey of multi-label image retrieval [48] published recently in 2020 for wider aspects and developments.

7.4 Instance Retrieval

In 2015, Razavian et al. have developed a baseline for deep CNN based visual instance retrieval [214]. They have evaluated different CNN models over different instance retrieval datasets. Another study for image instance retrieval is performed in [215] using CNNs. An instance-aware image representations for multi-label image data by modeling the features of one category in a group is proposed in [211]. Other approaches for image instance retrieval includes bags of local convolutional features [198], learning global representations [71], group invariant deep representation [216], DeepHash [217], and nested invariance pooling and RBM hashing [218]. In 2020, Chen et al. have proposed a deep multiple-instance ranking based hashing (DMIRH) model for multi-label image retrieval by employing the category-aware bag of feature [212]. The DMIRH uses region proposal network on top of VGG16 intermediate layer features. More details about image instance retrieval can be found in the survey compiled by Zheng et al. (2017) [47].

7.5 Object Retrieval

The object retrieval aims to perform the retrieval based on the features derived from the specific objects in the image. In 2014, Sun et al. have extracted the CNN features from the region of interest detected through object detection technique for object based retrieval [147]. Integral image driven max-pooling on CNN activations is used by Tolias et al. (2016) to detect the regions for object retrieval [219]. Gordo et al. (2016) have pooled the relevant features based on the region proposal network to construct the final global descriptor for image retrieval [71]. Pang et al. (2018) [220] have incorporated the replicator equation to simultaneously select and weight the primitive deep CNN features for object retrieval. A co-weighting scheme is used by Zhu et al. for aggregating the semantic CNN features for object retrieval [204]. In order to facilitate the object based image retrieval, Shi and Qian (2019) [221] have employed the spatial and channel contribution to improve the region detection. Recently, Gao et al. (2020) [148] have performed the 3D object retrieval with the help of a multi-view discrimination and pairwise CNN (MDPCNN) network.

7.6 Semantic Retrieval

In 2016, Yao et al. [163] have introduced a deep semantic preserving and ranking-based hashing (DSRH) method by exploiting

the hash and classification losses. Similar losses are also used in [222] for semantic image retrieval. Cao et al. (2017) [223] have developed a deep visual-semantic quantization (DVSQ) network by jointly learning the visual-semantic embeddings and quantizers. In order to incorporate the semantic features, an adaptive Gaussian filter based aggregation of CNN features is used in [204]. An asymmetric deep semantic quantization (ADSQ) approach uses three streams including one LabelNet and two ImgNets for semantic image retrieval [224]. Semantic hashing has been also extensively performed for sketch based image retrieval [180], [167], [209], [146], cross-modal retrieval [192], [205], [183]. Other notable deep learning based works that model the semantic information include Multi-label retrieval [210], unsupervised image retrieval [111], supervised image retrieval [94], and semi-supervised image retrieval [115]. In a recent work, Wang et al. (2020) [132] have employed the continuous semantic similarity in Hamming space to develop the deep position-aware hashing (DPAH) for semantic image retrieval. In 2020, a semantic affinity based learning of codes is utilized by the deep semantic reconstruction hashing (DSRH) method for semantic retrieval [157].

7.7 Fine-Grained Image Retrieval

In order to increase the discriminative ability of the deep learnt descriptors, many researchers have utilized the fine-grained constraints in deep networks. In 2014, Wang et al. [68] have modeled the fine-grained image search by capturing the inter-class and intra-class image similarities using a siamese network. Song et al. (2017) have utilized the attention modules to model the spatial-semantic information for fine-grained sketch-based image retrieval [180]. The selective CNN features are used in [199] for fine-grained image retrieval. In 2018, the fine-grained ranking is performed using the weighted Hamming distance w.r.t. the different queries with the help of query-adaptive deep weighted hashing (QaDWH) approach [225]. Recently in 2020, Qin et al. [213] have preserved the multilevel semantic similarity between multi-label image pairs using the deep hashing with fine-grained feature learning (DH-FFL). At the same time, fine-grained image retrieval is performed by Zeng et al. (2020) [226] using a piecewise cross entropy loss function.

7.8 Asymmetric Quantization based Retrieval

In 2017, Wu et al. have performed the online asymmetric similarity learning to preserve the similarity between heterogeneous data to facilitate the cross-modal retrieval [206]. In 2018, Jiang et al. have proposed an asymmetric deep supervised hashing (ADSH) by learning the deep hash function only for query images, while the hash codes for gallery images are directly learned [134]. In 2019, Yang et al. have investigated an asymmetric deep semantic quantization (ADSQ) method using three stream networks to model the heterogeneous data [224]. Chen et al. (2019) have proposed a similarity preserving deep asymmetric quantization (SPDAQ) based image retrieval model by exploiting the image subset and the label information of all the database items [227]. An adversary guided asymmetric hashing (AGAH) is introduced by Gu et al. (2019) [174] with the help of adversarial learning guided multi-label attention module for cross-modal image retrieval.

7.9 Summary

Based on the progress in image retrieval using deep learning methods for different retrieval types, following are the outlines drawn from this section:

- The cross-modal retrieval approaches learn the joint features for multiple modality using different networks. The recent methods utilize of the adversarial network for cross-modal retrieval. The similar observation and trend has been also witnessed for sketch based image retrieval.
- The multi-label and instance retrieval approaches are generally useful where more than one type of visual scenarios is present in the image. The deep learning based approaches are able to handle such retrieval by facilitating the feature learning through different type of networks.
- The region proposal network based feature selection has been employed by the existing deep learning methods for the object retrieval.
- The semantic information of the image has been used by different networks through abstract features to enhance the semantic image retrieval. The reconstruction based network is more suitable for semantic preserving hashing.
- Different feature selection and aggregation based networks have been utilized for fine-grained image retrieval due to improved discriminative & robustness of such approaches.
- The asymmetric hashing has also shown the suitability of deep learning models by processing the query and gallery images with different networks.

8 MISCELLANEOUS

This section covers the deep learning models for image retrieval in terms of the different losses, applications and other aspects.

8.1 Progress in Retrieval Loss

A siamese based loss function is used in [164] by Kumar et al. (2016) for minimizing the global loss leading to discriminative feature learning. Zhou et al. (2017) have used the triplet quantization loss for deep hashing, which is based on the similarity between the anchor-positive pairs and anchor-negative pairs [165]. A listwise loss has been employed by Revaud et al. in 2019 [228] to directly optimize the global mean average precision in end-to-end deep learning. Recently in 2020, a piecewise cross entropy loss function is used by Zeng et al. [226] fine-grained image retrieval. Several innovative loss functions have been used by the different feature learning approaches such as a lifted structured loss [72], higher-order learnable energy function (HOLEF) based loss [180], ranking loss [116], scaling and shifting based adaptive loss function [129], holographic composition layer based loss [184], second-order loss [179], and kurtosis loss (KT loss) to handle the distribution of real-valued features [132], etc.

8.2 Applications

The deep learning based approaches have been utilized for image retrieval pertaining to different applications such as cloth retrieval [189], biomedical image retrieval [190], [191], face retrieval [230], [231], remote sensing image retrieval [142], landmark retrieval [232], shape retrieval [208], social image retrieval [122], and video retrieval [125], [184], etc.

8.3 Others

In 2015, Wang et al. have identified the nuances in terms of the what works and what not for deep leart features based image retrieval [233]. The hashing difficulty is also increased by generating the harder samples in a self-paced manner [117] to make

TABLE 3: Mean Average Precision (mAP) with 5000 retrieved images (mAP@5000) in % for different deep learning based image retrieval approaches over NUS-WIDE, MS COCO and CIFAR-10 datasets. Note that 2^{nd} column list the reference from where the results of corresponding approach are considered.

Method	Result	NUS-WIDE			MS COCO		
Name	Source	16 Bits	32 Bits	64 Bits	16 Bits	32 Bits	64 Bits
CNNH'14 [86]	[77]	57.0	58.3	60.0	56.4	57.4	56.7
SDH'15 [88]	[77]	47.6	55.5	58.1	55.5	56.4	58.0
DNNH'15 [69]	[77]	59.8	61.6	63.9	59.3	60.3	61.0
DHN'16 [73]	[77]	63.7	66.4	67.1	67.7	70.1	69.4
HashNet'17 [77]	[77]	66.2	69.9	71.6	68.7	71.8	73.6
DeepBit'16 [103]	[130]	39.2	40.3	42.9	40.7	41.9	43.0
BGAN'18 [110]	[130]	68.4	71.4	73.0	64.5	68.2	70.7
GreedyHash'18 [82]	[130]	63.3	69.1	73.1	58.2	66.8	71.0
BinGAN'18 [171]	[130]	65.4	70.9	71.3	65.1	67.3	69.6
DVB'19 [114]	[130]	60.4	63.2	66.5	57.0	62.9	62.3
DistillHash'19 [113]	[130]	66.7	67.5	67.7	-	-	-
TBH'20 [130]	[130]	71.7	72.5	73.5	70.6	73.5	72.2
CNNH'14 [86]	[129]	57.0	58.3	60.0	56.4	57.4	56.7
DNNH'15 [69]	[129]	59.8	61.6	63.9	59.3	60.3	61.0
SDH'15 [88]	[129]	47.6	55.5	58.1	55.5	56.4	58.0
DHN'16 [73]	[129]	63.7	66.4	67.1	67.7	70.1	69.4
HashNet'17 [77]	[129]	66.3	69.9	71.6	68.7	71.8	73.6
DHA'19 [129]	[129]	66.9	70.6	72.7	70.8	73.1	75.2
HashGAN'18 [109]	[109]	71.5	73.7	74.8	69.7	72.5	74.4
UH-BDNN'16 [104]	[112]	59.2	59.0	61.0	-	-	-
UTH'17 [106]	[112]	54.3	53.7	54.7	-	-	-
UDTH'19 [112]	[112]	64.4	67.7	69.6	-	-	-
SSDH'17 [94]	[132]	-	-	-	69.7	72.5	74.4
DPAH'20 [132]	[132]	-	-	-	73.3	76.8	**78.2**
DRDH'20 [185]	[185]	80.5	81.7	**81.8**	71.5	74.8	76.1
DVSQ'17 [223]	[227]	79.0	79.7	-	71.2	72.0	-
DTQ'18 [166]	[227]	79.8	80.1	-	76.0	76.7	-
SPDAQ'19 [227]	[227]	**84.2**	85.1	-	**84.4**	**84.7**	-
DSQ'19 [97]	[97]	77.9	79.0	79.9	-	-	-
		CIFAR-10 Dataset					
		-	12 Bits	24 Bits	32 Bits	48 Bits	-
SDH'15 [88]	[229]	-	45.4	63.3	65.1	66.0	-
DSH'16 [89]	[229]	-	64.4	74.2	77.0	79.9	-
DHN'16 [73]	[229]	-	68.1	72.1	72.3	73.3	-
DPSH'16 [91]	[229]	-	68.2	72.0	73.4	74.6	-
DQN'16 [74]	[229]	-	55.4	55.8	56.4	58.0	-
DSDH'17 [95]	[229]	-	74.0	78.6	80.1	82.0	-
ADSH'18 [134]	[229]	-	89.0	92.8	93.1	93.9	-
DIHN2+ADSH'19 [127]	[229]	-	89.8	92.9	92.9	93.9	-
DTH'20 [229]	[229]	-	**92.1**	**93.3**	**93.7**	**94.9**	-

the network training as reasoning oriented. In the initial work, the pre-trained CNN features have also very promising retrieval performance [67]. The fine tuning without human annotation is also performed by Radenovic et al. [234] for CNN image retrieval. Recently, the transfer learning has been also utilized in [229] for deep transfer hashing based image retrieval.

8.4 Summary

Researchers have come up with various loss functions to facilitate the discriminative learning of features by the networks for image retrieval. The losses constraint and guide the training of the deep learning models. The image retrieval has shown a great utilization with its application to solve the real-life problems. Researchers have also tried to understand what works and what not for deep learning based image retrieval. The transfer learning has been also utilized for the image retrieval.

TABLE 4: Mean Average Precision (mAP) with 1000 retrieved images (mAP@1000) in % for different deep learning based image retrieval methods over ImageNet, CIFAR-10 and MNIST datasets.

Method	Result Source	16 Bits	32 Bits	48 Bits	64 Bits
		ImageNet Dataset			
CNNH'14 [86]	[77]	28.1	45.0	52.5	55.4
SDH'15 [88]	[77]	29.9	45.5	55.5	58.5
DNNH'15 [69]	[77]	29.0	46.1	53.0	56.5
DHN'16 [73]	[77]	31.1	47.2	54.2	57.3
HashNet'17 [77]	[77]	50.6	63.1	66.3	68.4
SSDH'17 [94]	[132]	63.4	69.2	70.1	70.7
DSQ'19 [97]	[97]	57.8	65.4	68.0	69.4
DPAH'20 [132]	[132]	**65.2**	**70.0**	**71.5**	**71.4**
		CIFAR-10 Dataset			
BGAN'18 [110]	[130]	52.5	53.1	-	56.2
GreedyHash'18 [82]	[130]	44.8	47.3	-	50.1
BinGAN'18 [171]	[130]	47.6	51.2	-	52.0
HashGAN'18 [109]	[130]	44.7	46.3	-	48.1
DVB'19 [114]	[130]	40.3	42.2	-	44.6
DistillHash'19 [113]	[130]	28.4	28.5	-	28.8
TBH'20 [130]	[130]	**53.2**	**57.3**	-	**57.8**
SDH'15 [87]	[87]	18.8	20.8	-	22.5
DAR'16 [102]	[102]	16.8	17.0	-	17.2
DH'15 [87]	[106]	16.2	16.6	-	17.0
DeepBit'16 [103]	[106]	19.4	24.9	-	27.7
UTH'17 [106]	[106]	28.7	30.7	-	32.4
DBD-MQ'17 [105]	[105]	21.5	26.5	-	31.9
UCBD'18 [108]	[108]	26.4	27.9	-	34.1
UH-BDNN'16 [104]	[112]	30.1	30.9	-	31.2
UDTH'19 [112]	[112]	46.1	50.4	-	54.3
		MNIST Dataset			
SDH'15 [87]	[87]	**46.8**	**51.0**	-	**52.5**
DH'15 [87]	[106]	43.1	45.0	-	46.7
DeepBit'16 [103]	[106]	28.2	32.0	-	44.5
UTH'17 [106]	[106]	43.2	46.6	-	49.9

TABLE 5: mAP@54000 and mAP@All in % for state-of-the-art and recent image retrieval methods over the CIFAR-10 dataset.

Methods	Result	CIFAR-10 Dataset				
Name	Source	16 Bits	24 Bits	32 Bits	48 Bits	64 Bits
		mAP@54000				
CNNH'14 [86]	[129]	47.6	-	47.2	48.9	50.1
DNNH'15 [69]	[129]	55.9	-	55.8	58.1	58.3
SDH'15 [88]	[129]	46.1	-	52.0	55.3	56.8
DHN'16 [73]	[129]	56.8		60.3	62.1	63.5
HashNet'17 [77]	[129]	64.3	-	66.7	67.5	68.7
DHA'14 [129]	[129]	65.2		68.1	69.0	69.9
HashGAN'18 [109]	[109]	66.8	-	73.1	73.5	74.9
DTQ'18 [166]	[166]	**78.9**	-	79.2	-	-
DRDH'20 [185]	[185]	78.7	-	**80.5**	**80.6**	**80.3**
		mAP@All				
DQN'16 [74]	[227]	-	55.8	56.4	58.0	-
DPSH'16 [91]	[227]	-	72.7	74.4	75.7	-
DSDH'17 [95]	[227]	-	78.6	80.1	82.0	-
DTQ'18 [166]	[227]	-	79.0	79.2	-	-
DVSQ'17 [223]	[227]	-	80.3	80.8	81.1	-
SPDAQ'19 [227]	[227]	-	**88.4**	**89.1**	**89.3**	-
SSAH'19 [117]	[117]	-	87.8	-	88.6	-
DeepBit'19 [103]	[111]	22.0	-	24.1	-	29.0
BGAN'19 [110]	[111]	49.7	-	47.0	-	50.7
UADH'19 [111]	[111]	**67.7**	-	68.9	-	**69.6**
DSAH'19 [178]	[178]	-	84.1	84.5	84.9	-

9 Performance Comparison

This survey also presents a performance analysis for the state-of-the-art deep learning based image retrieval approaches. The Mean Average Precision (mAP) reported for the different image retrieval

approaches is summarized in Table 3, 4, and 5. The mAP@5000 (i.e., 5000 retrieved images) using various existing deep learning approaches is summarized in Table 3 over CIFAR-10, NUS-WIDE and MS COCO datasets. The results over CIFAR-10, ImageNet and MNIST datasets using different state-of-the-art deep learning based image retrieval methods are compiled in Table 4 in terms of the mAP@1000. The mAP@54000 using few methods is reported in Table 5 over the CIFAR-10 dataset. The standard mAP is also depicted in Table 5 by considering all the retrieved images for CIFAR-10 dataset using some of the available literature. Note that 2nd column in Table 3, 4, and 5 list the source reference of the corresponding method reported results. Following are the observations out of these results by deep learning methods:

- Recently proposed Deep Transfer Hashing (DTH) by Zhai et al. (2020) [229] have shown outstanding performance over CIFAR-10 and NUS-WIDE datasets in terms of the mAP@5000. Other promising methods include Deep Spatial Attention Hashing (DSAH) by Ge et al. (2019) [178], Similarity Preserving Deep Asymmetric Quantization (SPDAQ) by Chen et al. (2019) [227], Deep Position-Aware Hashing (DPAH) by Wang et al. (2020) [132] and Deep Reinforcement De-Redundancy Hashing (DRDH) by Yang et al. (2020) [185].
- The Twin-Bottleneck Hashing (TBH) introduced by Shen et al. (2020) [130] is also observed as an appealing method using autoencoder having a double bottleneck over the CIFAR-10 dataset in terms of the mAP@1000. However, the Deep Position-Aware Hashing (DPAH) investigated by Wang et al. (2020) [132] have outperformed the other approaches over ImageNet dataset. Supervised Deep Hashing (SDH) by Erin et al. (2015) [87] has depicted appealing performance over the MNIST dataset.
- The deep reinforcement learning based image retrieval model, namely Deep Reinforcement De-Redundancy Hashing (DRDH) by Yang et al. (2020) [185], is one of recent breakthrough as supported by superlative mAP@54000 over the CIFAR-10 dataset. The Deep Triplet Quantization [166] is also one of the favourable model for feature learning.
- The Similarity Preserving Deep Asymmetric Quantization (SPDAQ) by Chen et al. (2019) [227] and Unsupervised ADversarial Hashing (UADH) by Deng et al. (2019) [111] methods have been also identified as very encouraging based on the mAP by considering all the retrieved images over the CIFAR-10 dataset.

10 Conclusion and Future Directives

This paper presents a comprehensive survey of deep learning methods for content based image retrieval. As most of the deep learning based developments are recent, this survey majorly focuses over the image retrieval methods using deep learning in a decade from 2011 to 2020. A detailed taxonomy is presented in terms of different supervision type, different networks used, different data type of descriptors, different retrieval type and other aspects. The detailed discussion under each section is also presented with the further categorization. A chronological summarization is presented to show the evolution of the deep learning models for image retrieval. Moreover, the chronological overview is also portrayed under each category to showcase the growth of image retrieval approaches. A summary of large-scale common datasets used for image retrieval is also compiled in this survey. A performance analysis of the state-of-the-art deep learning based image retrieval methods is also conducted in terms of the mean average precision for different no. of retrieved images.

The research trend in image retrieval suggests that the deep learning based models are driving the progress. The recently developed models such as generative adversarial networks, autoencoder networks and reinforcement learning networks have shown the superior performance for image retrieval. The discovery of better objective functions has been also the trend in order to constrain the learning of the hash code for discriminative, robust and efficient image retrieval. The semantic preserving class-specific feature learning using different networks and different quantization techniques is also the recent trend for image retrieval. Other trends include utilization of attention module, transfer learning, etc.

The future directions in the image retrieval include exploration of improved deep learning models, more relevant objective functions, minimum loss based quantization techniques, semantic preserving feature learning, and attention focused feature learning.

Acknowledgement

This work is supported by Global Innovation & Technology Alliance (GITA) on behalf of Department of Science and Technology (DST), Govt. of India through project no. GITA/DST/TWN/P-83/2019.

References

[1] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic *et al.*, "Query by image and video content: The qbic system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[2] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE TPAMI*, vol. 22, no. 12, pp. 1349–1380, 2000.

[3] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: overview and proposals," *Pattern Recog. Letters*, vol. 22, no. 5, pp. 593–601, 2001.

[4] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE TPAMI*, vol. 23, no. 9, pp. 947–963, 2001.

[5] S. R. Dubey, "Robust image feature description, matching & applications," Ph.D. dissertation, IIIT Allahabad, India, 2016.

[6] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," *Information Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[7] S. R. Dubey, S. K. Singh, and R. K. Singh, "Rotation and scale invariant hybrid image descriptor and retrieval," *Computers & Electrical Engineering*, vol. 46, pp. 288–302, 2015.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE TPAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[10] S. R. Dubey, S. K. Singh, and R. K. Singh, "Rotation and illumination invariant interleaved intensity order-based local descriptor," *IEEE TIP*, vol. 23, no. 12, pp. 5323–5333, 2014.

[11] I. J. Jacob, K. Srinivasagan, and K. Jayapriya, "Local oppugnant color texture pattern for image retrieval system," *Pattern Recog. Letters*, vol. 42, pp. 72–78, 2014.

[12] S. R. Dubey, S. K. Singh, and R. K. Singh, "Local diagonal extrema pattern: a new and efficient feature descriptor for ct image retrieval," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1215–1219, 2015.

[13] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE TPAMI*, vol. 34, no. 9, pp. 1704–1716, 2011.

[14] S. Murala, R. Maheshwari, and R. Balasubramanian, "Local tetra patterns: a new feature descriptor for content-based image retrieval," *IEEE TIP*, vol. 21, no. 5, pp. 2874–2886, 2012.

[15] S. R. Dubey, S. K. Singh, and R. K. Singh, "Local wavelet pattern: a new feature descriptor for image retrieval in medical ct databases," *IEEE TIP*, vol. 24, no. 12, pp. 5892–5903, 2015.

[16] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE TPAMI*, vol. 35, no. 12, pp. 2916–2929, 2012.

[17] S. R. Dubey, S. K. Singh, and R. K. Singh, "Local bit-plane decoded pattern: a novel feature descriptor for biomedical image retrieval," *IEEE JBHI*, vol. 20, no. 4, pp. 1139–1147, 2015.

[18] K.-C. Fan and T.-Y. Hung, "A novel local pattern descriptor—local vector pattern in high-order derivative space for face recognition," *IEEE TIP*, vol. 23, no. 7, pp. 2877–2891, 2014.

[19] S. R. Dubey, S. K. Singh, and R. K. Singh, "Multichannel decoded local binary patterns for content-based image retrieval," *IEEE TIP*, vol. 25, no. 9, pp. 4018–4032, 2016.

[20] R. C. Veltkamp and M. Tanase, "A survey of content-based image retrieval systems," in *Content-based image and video retrieval*. Springer, 2002, pp. 47–101.

[21] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applications—clinical benefits and future directions," *Int. J. of Medical Informatics*, vol. 73, no. 1, pp. 1–23, 2004.

[22] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM TOMM*, vol. 2, no. 1, pp. 1–19, 2006.

[23] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[24] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 1–60, 2008.

[25] T. Mei, Y. Rui, S. Li, and Q. Tian, "Multimedia search reranking: A literature survey," *ACM Computing Surveys*, vol. 46, no. 3, pp. 1–38, 2014.

[26] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *arXiv preprint arXiv:1706.06064*, 2017.

[27] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *arXiv preprint arXiv:1306.6709*, 2013.

[28] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE TPAMI*, vol. 40, no. 4, pp. 769–790, 2017.

[29] S. C. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *CVPR*, vol. 2, 2006, pp. 2072–2078.

[30] H. Chang and D.-Y. Yeung, "Kernel-based distance metric learning for content-based image retrieval," *Image and Vision Computing*, vol. 25, no. 5, pp. 695–703, 2007.

[31] L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. Hoi, and M. Satyanarayanan, "A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval," *IEEE TPAMI*, vol. 32, no. 1, pp. 30–44, 2008.

[32] J.-E. Lee, R. Jin, and A. K. Jain, "Rank-based distance metric learning: An application to image retrieval," in *CVPR*, 2008, pp. 1–8.

[33] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM TOMM*, vol. 6, no. 3, pp. 1–26, 2010.

[34] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR workshops*, 2014, pp. 806–813.

[35] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE TPAMI*, 2020.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[38] C. H. Dagli, *Artificial neural networks for intelligent manufacturing*. Springer Science & Business Media, 2012.

[39] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel, "Artificial neural networks in medical diagnosis," *J Appl Biomed*, vol. 11, pp. 47–58, 2013.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[43] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *ICML*, 2015, pp. 2067–2075.

[44] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," *arXiv preprint arXiv:1609.01704*, 2016.

[45] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Conf. of the Int. Speech Communication Association*, 2012.

[46] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACMMM*, 2014, pp. 157–166.

[47] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE TPAMI*, vol. 40, no. 5, pp. 1224–1244, 2017.

[48] J. Rodrigues, M. Cristo, and J. G. Colonna, "Deep hashing for multi-label image retrieval: a survey," *Artificial Intel. Review*, pp. 1–47, 2020.

[49] X. Luo, C. Chen, H. Zhong, H. Zhang, M. Deng, J. Huang, and X. Hua, "A survey on deep hashing methods," *arXiv preprint arXiv:2003.03369*, 2020.

[50] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Tront*, 2009.

[51] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *ACM Int. Conf. on Image and Video Retrieval*, 2009, pp. 1–9.

[52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[53] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS*, 2011.

[54] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[55] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *CVPR*, 2014, pp. 192–199.

[56] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *CVPR workshops*, 2015, pp. 27–35.

[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[58] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[59] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative," in *ICMIR*, 2010, pp. 527–536.

[60] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *ICCV*, 2017, pp. 3456–3465.

[61] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *CVPR*, 2020, pp. 2575–2584.

[62] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *ACMMM Information Retrieval*, 2008, pp. 39–43.

[63] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "Sun database: Exploring a large collection of scene categories," *IJCV*, vol. 119, no. 1, pp. 3–22, 2016.

[64] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval." in *ESANN*, vol. 1, 2011, p. 2.

[65] Y. Kang, S. Kim, and S. Choi, "Deep learning to hash with multiple representations," in *ICDM*, 2012, pp. 930–935.

[66] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao, "Online multimodal deep similarity learning with application to image retrieval," in *ACMMM*, 2013, pp. 153–162.

[67] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *ECCV*, 2014, pp. 584–599.

[68] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *CVPR*, 2014, pp. 1386–1393.

[69] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *CVPR*, 2015, pp. 3270–3278.

[70] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and

person re-identification," *IEEE TIP*, vol. 24, no. 12, pp. 4766–4779, 2015.

[71] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *ECCV*, 2016, pp. 241–257.

[72] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016, pp. 4004–4012.

[73] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *AAAI*, 2016.

[74] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *AAAI*, 2016.

[75] S. S. Husain and M. Bober, "Improving large-scale image retrieval through robust aggregation of local descriptors," *IEEE TPAMI*, vol. 39, no. 9, pp. 1783–1796, 2016.

[76] G. Zhong, H. Xu, P. Yang, S. Wang, and J. Dong, "Deep hashing learning networks," in *IJCNN*, 2016, pp. 2236–2243.

[77] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *ICCV*, 2017, pp. 5608–5617.

[78] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *IJCV*, vol. 124, no. 2, pp. 237–254, 2017.

[79] T. Hoang, T.-T. Do, D.-K. Le Tan, and N.-M. Cheung, "Selective deep convolutional features for image retrieval," in *ACMMM*, 2017, pp. 1600–1608.

[80] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105, 2017.

[81] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for hamming space retrieval," in *CVPR*, 2018, pp. 1229–1237.

[82] S. Su, C. Zhang, K. Han, and Y. Tian, "Greedy hash: Towards fast optimization for accurate hash coding in cnn," in *NIPS*, 2018, pp. 798–807.

[83] X. Yuan, L. Ren, J. Lu, and J. Zhou, "Relaxation-free deep hashing via policy gradient," in *ECCV*, 2018, pp. 134–150.

[84] Z. Chen, X. Yuan, J. Lu, Q. Tian, and J. Zhou, "Deep hashing via discrepancy minimization," in *CVPR*, 2018, pp. 6838–6847.

[85] D. Wu, J. Liu, B. Li, and W. Wang, "Deep index-compatible hashing for fast image retrieval," in *ICME*, 2018, pp. 1–6.

[86] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, 2014.

[87] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *CVPR*, 2015, pp. 2475–2483.

[88] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *CVPR*, 2015, pp. 37–45.

[89] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *CVPR*, 2016, pp. 2064–2072.

[90] Z. Zhang, Y. Chen, and V. Saligrama, "Efficient training of very deep neural networks for supervised hashing," in *CVPR*, 2016, pp. 1487–1495.

[91] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *IJCAI*, 2016, pp. 1711–1717.

[92] X. Wang, Y. Shi, and K. M. Kitani, "Deep supervised hashing with triplet labels," in *ACCV*, 2016, pp. 70–84.

[93] J. Lu, V. E. Liong, and J. Zhou, "Deep hashing for scalable image search," *IEEE TIP*, vol. 26, no. 5, pp. 2352–2367, 2017.

[94] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE TPAMI*, vol. 40, no. 2, pp. 437–451, 2017.

[95] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *NIPS*, 2017, pp. 2482–2491.

[96] Y. Cao, B. Liu, M. Long, and J. Wang, "Hashgan: Deep learning to hash with pair conditional wasserstein gan," in *CVPR*, 2018, pp. 1287–1296.

[97] S. Eghbali and L. Tahvildari, "Deep spherical quantization for image search," in *CVPR*, 2019, pp. 11 690–11 699.

[98] B. Klein and L. Wolf, "End-to-end supervised product quantization for image search and retrieval," in *CVPR*, 2019, pp. 5041–5050.

[99] J. Li, W. W. Ng, X. Tian, S. Kwong, and H. Wang, "Weighted multi-deep ranking supervised hashing for efficient image retrieval," *Int. Journal of Machine Learning and Cybernetics*, pp. 1–15, 2019.

[100] Y. Shen, J. Qin, J. Chen, L. Liu, F. Zhu, and Z. Shen, "Embarrassingly simple binary representation learning," in *ICCV Workshops*, 2019.

[101] W. W. Ng, J. Li, X. Tian, H. Wang, S. Kwong, and J. Wallace, "Multi-level supervised hashing with deep features for efficient image retrieval," *Neurocomputing*, 2020.

[102] C. Huang, C. Change Loy, and X. Tang, "Unsupervised learning of discriminative attributes and visual representations," in *CVPR*, 2016, pp. 5175–5184.

[103] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *CVPR*, 2016, pp. 1183–1192.

[104] T.-T. Do, A.-D. Doan, and N.-M. Cheung, "Learning to hash with binary deep neural network," in *ECCV*, 2016, pp. 219–234.

[105] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptor with multi-quantization," in *CVPR*, 2017, pp. 1183–1192.

[106] S. Huang, Y. Xiong, Y. Zhang, and J. Wang, "Unsupervised triplet hashing for fast image retrieval," in *Thematic Workshops of ACM Multimedia*, 2017, pp. 84–92.

[107] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE TPAMI*, vol. 40, no. 12, pp. 3034–3044, 2018.

[108] K. Lin, J. Lu, C.-S. Chen, J. Zhou, and M.-T. Sun, "Unsupervised deep learning of compact binary descriptors," *IEEE TPAMI*, vol. 41, no. 6, pp. 1501–1514, 2018.

[109] K. Ghasedi Dizaji, F. Zheng, N. Sadoughi, Y. Yang, C. Deng, and H. Huang, "Unsupervised deep generative adversarial hashing network," in *CVPR*, 2018, pp. 3664–3673.

[110] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Binary generative adversarial networks for image retrieval," in *AAAI*, 2018.

[111] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE TIP*, vol. 28, no. 8, pp. 4032–4044, 2019.

[112] Y. Gu, H. Zhang, Z. Zhang, and Q. Ye, "Unsupervised deep triplet hashing with pseudo triplets for scalable image retrieval," *Multimedia Tools and Applications*, pp. 1–22, 2019.

[113] E. Yang, T. Liu, C. Deng, W. Liu, and D. Tao, "Distillhash: Unsupervised deep hashing by distilling data pairs," in *CVPR*, 2019, pp. 2946–2955.

[114] Y. Shen, L. Liu, and L. Shao, "Unsupervised binary representation learning with deep variational networks," *IJCV*, vol. 127, no. 11-12, pp. 1614–1628, 2019.

[115] J. Zhang and Y. Peng, "Ssdh: semi-supervised deep hashing for large scale image retrieval," *IEEE TCSVT*, vol. 29, no. 1, pp. 212–225, 2017.

[116] G. Wang, Q. Hu, J. Cheng, and Z. Hou, "Semi-supervised generative adversarial hashing for image retrieval," in *ECCV*, 2018, pp. 469–485.

[117] S. Jin, S. Zhou, Y. Liu, C. Chen, X. Sun, H. Yao, and X. Hua, "Ssah: Semi-supervised adversarial deep hashing with self-paced hard sample generation," *arXiv preprint arXiv:1911.08688*, 2019.

[118] S. Zhang, J. Li, and B. Zhang, "Pairwise teacher-student network for semi-supervised hashing," in *CVPR Workshops*, 2019.

[119] J. Tang and Z. Li, "Weakly supervised multimodal hashing for scalable social image retrieval," *IEEE TCSVT*, vol. 28, no. 10, pp. 2730–2741, 2017.

[120] Z. Guan, F. Xie, W. Zhao, X. Wang, L. Chen, W. Zhao, and J. Peng, "Tag-based weakly-supervised hashing for image retrieval." in *IJCAI*, 2018, pp. 3776–3782.

[121] V. Gattupalli, Y. Zhuo, and B. Li, "Weakly supervised deep image hashing through tag embeddings," in *CVPR*, 2019, pp. 10 375–10 384.

[122] Z. Li, J. Tang, L. Zhang, and J. Yang, "Weakly-supervised semantic guided hashing for social image retrieval," *IJCV*, 2020.

[123] Q. Hu, J. Wu, J. Cheng, L. Wu, and H. Lu, "Pseudo label based unsupervised deep discriminative hashing for image retrieval," in *ACMMM*, 2017, pp. 1584–1590.

[124] Y. Liu, Y. Wang, J. Song, C. Guo, K. Zhou, and Z. Xiao, "Deep self-taught graph embedding hashing with pseudo labels for image retrieval," in *ICME*, 2020, pp. 1–6.

[125] H. Zhang, M. Wang, R. Hong, and T.-S. Chua, "Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing," in *ACMMM*, 2016, pp. 781–790.

[126] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *CVPR*, 2018, pp. 4242–4251.

[127] D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang, "Deep incremental hashing network for efficient image retrieval," in *CVPR*, 2019, pp. 9069–9077.

[128] J. Bai, B. Ni, M. Wang, Z. Li, S. Cheng, X. Yang, C. Hu, and W. Gao, "Deep progressive hashing for image retrieval," *IEEE TMM*, vol. 21, no. 12, pp. 3178–3193, 2019.

[129] J. Xu, C. Guo, Q. Liu, J. Qin, Y. Wang, and L. Liu, "Dha: Supervised deep learning to hash with an adaptive loss function," in *ICCV Workshops*, 2019.

[130] Y. Shen, J. Qin, J. Chen, M. Yu, L. Liu, F. Zhu, F. Shen, and L. Shao, "Auto-encoding twin-bottleneck hashing," in *CVPR*, 2020, pp. 2818–2827.

[131] J. I. Forcen, M. Pagola, E. Barrenechea, and H. Bustince, "Co-occurrence of deep convolutional features for image search," *Image and Vision Computing*, p. 103909, 2020.

[132] R. Wang, R. Wang, S. Qiao, S. Shan, and X. Chen, "Deep position-aware hashing for semantic continuous image retrieval," in *WACV*, 2020, pp. 2493–2502.

[133] D. Wu, Z. Lin, B. Li, M. Ye, and W. Wang, "Deep supervised hashing for multi-label and large-scale image retrieval," in *ICMR*, 2017, pp. 150–158.

[134] Q.-Y. Jiang and W.-J. Li, "Asymmetric deep supervised hashing," in *AAAI*, 2018.

[135] S. Kan, Y. Cen, Z. He, Z. Zhang, L. Zhang, and Y. Wang, "Supervised deep feature embedding with handcrafted feature," *IEEE TIP*, vol. 28, no. 12, pp. 5809–5823, 2019.

[136] C. Zhou, L.-M. Po, W. Y. Yuen, K. W. Cheung, X. Xu, K. W. Lau, Y. Zhao, M. Liu, and P. H. Wong, "Angular deep supervised hashing for image retrieval," *IEEE Access*, vol. 7, pp. 127 521–127 532, 2019.

[137] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *ICCV*, 2015, pp. 91–99.

[138] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *ECCV*, 2016, pp. 3–20.

[139] J. Lin, O. Morere, J. Petta, V. Chandrasekhar, and A. Veillard, "Tiny descriptors for image retrieval with unsupervised triplet hashing," in *DCC*, 2016, pp. 397–406.

[140] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: an unsupervised approach," *IJCV*, vol. 121, no. 1, pp. 149–168, 2017.

[141] J. Xu, C. Shi, C. Qi, C. Wang, and B. Xiao, "Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval," in *AAAI*, 2018.

[142] Y. Liu, L. Ding, C. Chen, and Y. Liu, "Similarity-based unsupervised deep transfer learning for remote sensing image retrieval," *IEEE TGRS*, 2020.

[143] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan, "Cross-modal retrieval with cnn visual features: A new baseline," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 449–460, 2016.

[144] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *AAAI*, 2017.

[145] D. Yu, Y. Liu, Y. Pang, Z. Li, and H. Li, "A multi-layer deep fusion convolutional neural network for sketch based image retrieval," *Neurocomputing*, vol. 296, pp. 23–32, 2018.

[146] L. Wang, X. Qian, Y. Zhang, J. Shen, and X. Cao, "Enhancing sketch-based image retrieval by cnn semantic re-ranking," *IEEE Transactions on Cybernetics*, 2019.

[147] S. Sun, W. Zhou, H. Li, and Q. Tian, "Search by detection: Object-level feature for image retrieval," in *Int. conf. on Internet Multimedia Computing and Service*, 2014, pp. 46–49.

[148] Z. Gao, H. Xue, and S. Wan, "Multiple discrimination and pairwise cnn for view-based 3d object retrieval," *Neural Networks*, 2020.

[149] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *NIPS*, 2007, pp. 153–160.

[150] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008, pp. 1096–1103.

[151] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *ACMMM*, 2014, pp. 7–16.

[152] M. A. Carreira-Perpinán and R. Raziperchikolaei, "Hashing with binary autoencoders," in *CVPR*, 2015, pp. 557–566.

[153] T.-T. Do, K. Le, T. Hoang, H. Le, T. V. Nguyen, and N.-M. Cheung, "Simultaneous feature aggregating and hashing for compact binary code learning," *IEEE TIP*, vol. 28, no. 10, pp. 4954–4969, 2019.

[154] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *ICPR*, 2016, pp. 378–383.

[155] E.-J. Ong, S. Husain, and M. Bober, "Siamese network of deep fisher-vector descriptors for image retrieval," *arXiv preprint arXiv:1702.00338*, 2017.

[156] A. Jose, S. Yan, and I. Heisterklaus, "Binary hashing using siamese neural networks," in *ICIP*, 2017, pp. 2916–2920.

[157] Y. Wang, X. Ou, J. Liang, and Z. Sun, "Deep semantic reconstruction hashing for similarity retrieval," *IEEE TCSVT*, 2020.

[158] A. Pandey, A. Mishra, V. K. Verma, A. Mittal, and H. Murthy, "Stacked adversarial network for zero-shot sketch based image retrieval," in *WACV*, 2020, pp. 2540–2549.

[159] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015, pp. 4353–4361.

[160] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *CVPR*, 2015, pp. 3279–3286.

[161] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Int. W. on Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[162] B. Zhuang, G. Lin, C. Shen, and I. Reid, "Fast training of triplet-based deep binary embedding networks," in *CVPR*, 2016, pp. 5955–5964.

[163] T. Yao, F. Long, T. Mei, and Y. Rui, "Deep semantic-preserving and ranking-based hashing for image retrieval," in *IJCAI*, vol. 1, 2016, p. 4.

[164] V. Kumar BG, G. Carneiro, and I. Reid, "Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions," in *CVPR*, 2016, pp. 5385–5394.

[165] Y. Zhou, S. Huang, Y. Zhang, and Y. Wang, "Deep hashing with triplet quantization loss," in *VCIP*, 2017, pp. 1–4.

[166] B. Liu, Y. Cao, M. Long, J. Wang, and J. Wang, "Deep triplet quantization," in *ACMMM*, 2018, pp. 755–763.

[167] S. Dey, P. Riba, A. Dutta, J. Llados, and Y.-Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," in *CVPR*, 2019, pp. 2179–2188.

[168] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[169] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *ACMMM*, 2017, pp. 154–162.

[170] X. Zhang, H. Lai, and J. Feng, "Attention-aware deep adversarial hashing for cross-modal retrieval," in *ECCV*, 2018, pp. 591–606.

[171] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski, "Bingan: Learning compact binary descriptors with a regularized gan," in *NIPS*, 2018, pp. 3608–3618.

[172] J. Zhang, F. Shen, L. Liu, F. Zhu, M. Yu, L. Shao, H. Tao Shen, and L. Van Gool, "Generative domain-migration hashing for sketch-to-image retrieval," in *ECCV*, 2018, pp. 297–314.

[173] V. Kumar Verma, A. Mishra, A. Mishra, and P. Rai, "Generative model for zero-shot sketch-based image retrieval," in *CVPR Workshops*, 2019.

[174] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Adversary guided asymmetric hashing for cross-modal retrieval," in *ICMR*, 2019, pp. 159–167.

[175] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Unified binary generative adversarial network for image retrieval and compression," *IJCV*, pp. 1–22, 2020.

[176] F. Yang, J. Li, S. Wei, Q. Zheng, T. Liu, and Y. Zhao, "Two-stream attentive cnns for image retrieval," in *ACMMM*, 2017, pp. 1513–1521.

[177] S. Wei, L. Liao, J. Li, Q. Zheng, F. Yang, and Y. Zhao, "Saliency inside: Learning attentive cnns for content-based image retrieval," *IEEE TIP*, vol. 28, no. 9, pp. 4580–4593, 2019.

[178] L.-W. Ge, J. Zhang, Y. Xia, P. Chen, B. Wang, and C.-H. Zheng, "Deep spatial attention hashing network for image retrieval," *JVCIR*, vol. 63, p. 102577, 2019.

[179] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk, "Solar: Second-order loss and attention for image retrieval," *arXiv preprint arXiv:2001.08972*, 2020.

[180] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *ICCV*, 2017, pp. 5551–5560.

[181] L.-K. Huang, J. Chen, and S. J. Pan, "Accelerate learning of deep hashing with gradient attention," in *ICCV*, 2019, pp. 5271–5280.

[182] X. Lu, Y. Chen, and X. Li, "Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features," *IEEE TIP*, vol. 27, no. 1, pp. 106–120, 2017.

[183] Y. Shen, L. Liu, L. Shao, and J. Song, "Deep binaries: Encoding semantic-rich cues for efficient textual-visual cross retrieval," in *ICCV*, 2017, pp. 4097–4106.

[184] Z. Chen, J. Lin, Z. Wang, V. Chandrasekhar, and W. Lin, "Beyond ranking loss: Deep holographic networks for multi-label video search," in *ICIP*, 2019, pp. 879–883.

[185] J. Yang, Y. Zhang, R. Feng, T. Zhang, and W. Fan, "Deep reinforcement hashing with redundancy elimination for effective image retrieval," *Pattern Recognition*, vol. 100, p. 107116, 2020.

[186] T.-T. Do, T. Hoang, D.-K. Le Tan, A.-D. Doan, and N.-M. Cheung, "Compact hash code learning with binary deep neural network," *IEEE TMM*, vol. 22, no. 4, pp. 992–1004, 2019.

[187] T.-T. Do, T. Hoang, D.-K. L. Tan, H. Le, T. V. Nguyen, and N.-M. Cheung, "From selective deep convolutional features to compact binary representations for image retrieval," *ACM TOMM*, vol. 15, no. 2, pp. 1–22, 2019.
[188] X. Lu, Y. Chen, and X. Li, "Discrete deep hashing with ranking optimization for image retrieval," *IEEE TNNLS*, 2019.
[189] K. Lin, H.-F. Yang, K.-H. Liu, J.-H. Hsiao, and C.-S. Chen, "Rapid clothing retrieval via deep learning of binary codes and hierarchical search," in *ICMR*, 2015, pp. 499–502.
[190] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, 2017.
[191] S. R. Dubey, S. K. Roy, S. Chakraborty, S. Mukherjee, and B. B. Chaudhuri, "Local bit-plane decoded convolutional neural network features for biomedical image retrieval," *NCAA*, pp. 1–13, 2019.
[192] Z. Ji, W. Yao, W. Wei, H. Song, and H. Pi, "Deep multi-level semantic hashing for cross-modal retrieval," *IEEE Access*, vol. 7, pp. 23 667–23 674, 2019.
[193] Y. Chen and X. Lu, "Deep discrete hashing with pairwise correlation learning," *Neurocomputing*, vol. 385, pp. 111–121, 2020.
[194] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *ICCV*, 2015, pp. 1269–1277.
[195] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "Deepindex for accurate and efficient image retrieval," in *ICMR*, 2015, pp. 43–50.
[196] T. Uricchio, M. Bertini, L. Seidenari, and A. Bimbo, "Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging," in *ICCV Workshops*, 2015, pp. 9–15.
[197] J. Yue-Hei Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *CVPR workshops*, 2015, pp. 53–61.
[198] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto, "Bags of local convolutional features for scalable instance search," in *ICMR*, 2016, pp. 327–331.
[199] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE TIP*, vol. 26, no. 6, pp. 2868–2881, 2017.
[200] W. Yu, K. Yang, H. Yao, X. Sun, and P. Xu, "Exploiting the complementary strengths of multi-layer cnn features for image retrieval," *Neurocomputing*, vol. 237, pp. 235–241, 2017.
[201] X. Liu, S. Zhang, T. Huang, and Q. Tian, "E2bows: An end-to-end bag-of-words model via deep convolutional neural network for image retrieval," *Neurocomputing*, 2019.
[202] T.-T. Do, D.-K. Le Tan, T. T. Pham, and N.-M. Cheung, "Simultaneous feature aggregating and hashing for large-scale image search," in *CVPR*, 2017, pp. 6618–6627.
[203] W. Zhou, H. Li, J. Sun, and Q. Tian, "Collaborative index embedding for image retrieval," *IEEE TPAMI*, vol. 40, no. 5, pp. 1154–1166, 2017.
[204] J. Zhu, J. Wang, S. Pang, W. Guan, Z. Li, Y. Li, and X. Qian, "Co-weighting semantic convolutional features for object retrieval," *JVCIR*, vol. 62, pp. 368–380, 2019.
[205] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, "Deep visual-semantic hashing for cross-modal retrieval," in *ACM ICKDDM*, 2016, pp. 1445–1454.
[206] Y. Wu, S. Wang, and Q. Huang, "Online asymmetric similarity learning for cross-modal retrieval," in *CVPR*, 2017, pp. 4269–4278.
[207] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," in *CVPR*, 2017, pp. 2862–2871.
[208] W. Zhou, J. Jia, W. Jiang, and C. Huang, "Sketch augmentation-driven shape retrieval learning framework based on convolutional neural networks," *IEEE TVCG*, 2020.
[209] A. Dutta and Z. Akata, "Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval," in *CVPR*, 2019, pp. 5089–5098.
[210] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *CVPR*, 2015, pp. 1556–1564.
[211] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, "Instance-aware hashing for multi-label image retrieval," *IEEE TIP*, vol. 25, no. 6, pp. 2469–2479, 2016.
[212] G. Chen, X. Cheng, S. Su, and C. Tang, "Multiple-instance ranking based deep hashing for multi-label image retrieval," *Neurocomp.*, 2020.
[213] Q. Qin, L. Huang, and Z. Wei, "Deep multilevel similarity hashing with fine-grained features for multi-label image retrieval," *Neurocomp.*, 2020.
[214] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," in *ICLR*, 2015.
[215] V. Chandrasekhar, J. Lin, O. Morere, H. Goh, and A. Veillard, "A practical guide to cnns and fisher vectors for image instance retrieval," *Signal Processing*, vol. 128, pp. 426–439, 2016.
[216] O. Morère, A. Veillard, L. Jie, J. Petta, V. Chandrasekhar, and T. Poggio, "Group invariant deep representations for image instance retrieval," in *AAAI Spring Symposium Series*, 2017.
[217] J. Lin, O. Morère, A. Veillard, L.-Y. Duan, H. Goh, and V. Chandrasekhar, "Deephash for image instance retrieval: Getting regularization, depth and fine-tuning right," in *ICMR*, 2017, pp. 133–141.
[218] O. Morère, J. Lin, A. Veillard, L.-Y. Duan, V. Chandrasekhar, and T. Poggio, "Nested invariance pooling and rbm hashing for image instance retrieval," in *ICMR*, 2017, pp. 260–268.
[219] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.
[220] S. Pang, J. Zhu, J. Wang, V. Ordonez, and J. Xue, "Building discriminative cnn image representations for object retrieval using the replicator equation," *Pattern Recognition*, vol. 83, pp. 150–160, 2018.
[221] X. Shi and X. Qian, "Exploring spatial and channel contribution for object based image retrieval," *Knowledge-Based Systems*, vol. 186, p. 104955, 2019.
[222] J. Guo, S. Zhang, and J. Li, "Hash learning with convolutional neural networks for semantic based image retrieval," in *KDDM*, 2016, pp. 227–238.
[223] Y. Cao, M. Long, J. Wang, and S. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *CVPR*, 2017, pp. 1328–1337.
[224] Z. Yang, O. I. Raymond, W. Sun, and J. Long, "Asymmetric deep semantic quantization for image retrieval," *IEEE Access*, vol. 7, pp. 72 684–72 695, 2019.
[225] J. Zhang and Y. Peng, "Query-adaptive image retrieval by deep-weighted hashing," *IEEE TMM*, vol. 20, no. 9, pp. 2400–2414, 2018.
[226] X. Zeng, Y. Zhang, X. Wang, K. Chen, D. Li, and W. Yang, "Fine-grained image retrieval via piecewise cross entropy loss," *Image and Vision Computing*, vol. 93, p. 103820, 2020.
[227] J. Chen and W. K. Cheung, "Similarity preserving deep asymmetric quantization for image retrieval," in *AAAI*, vol. 33, 2019, pp. 8183–8190.
[228] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *ICCV*, 2019, pp. 5107–5116.
[229] H. Zhai, S. Lai, H. Jin, X. Qian, and T. Mei, "Deep transfer hashing for image retrieval," *IEEE TCSVT*, 2020.
[230] Z. Dong, C. Jing, M. Pei, and Y. Jia, "Deep cnn based binary hash video representations for face retrieval," *Pattern Recognition*, vol. 81, pp. 357–369, 2018.
[231] S. R. Dubey and S. Chakraborty, "Average biased relu based cnn descriptor for improved face retrieval," *arXiv preprint arXiv:1804.02051*, 2018.
[232] T.-Y. Yang, D. Kien Nguyen, H. Heijnen, and V. Balntas, "Dame web: Dynamic mean with whitening ensemble binarization for landmark retrieval without human annotation," in *ICCV Workshops*, 2019.
[233] H. Wang, Y. Cai, Y. Zhang, H. Pan, W. Lv, and H. Han, "Deep learning for image retrieval: What works and what doesn't," in *ICDM Workshop*, 2015, pp. 1576–1583.
[234] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *TPAMI*, vol. 41, no. 7, pp. 1655–1668, 2018.

Shiv Ram Dubey has been with the Indian Institute of Information Technology (IIIT), Sri City since June 2016, where he is currently the Assistant Professor of Computer Science and Engineering. He received the Ph.D. degree in Computer Vision and Image Processing from Indian Institute of Information Technology, Allahabad (IIIT Allahabad) in 2016. Before that, from August 2012-Feb 2013, he was a Project Officer in the Computer Science and Engineering Department at Indian Institute of Technology, Madras (IIT Madras). He was a recipient of several awards, including the Best PhD Award in PhD Symposium, IEEE-CICT2017 at IIITM Gwalior and NVIDIA GPU Grant Award Twice from NVIDIA. His research interest includes Computer Vision, Deep Learning, Image Feature Description, and Content Based Image Retrieval.

Robust Image Retrieval-based Visual Localization using Kapture

Martin Humenberger Yohann Cabon Nicolas Guerin Julien Morat Jérôme Revaud
Philippe Rerole Noé Pion Cesar de Souza Vincent Leroy
Gabriela Csurka
NAVER LABS Europe, 38240 Meylan, France
https://europe.naverlabs.com
firstname.lastname@naverlabs.com

Abstract

In this paper, we present a versatile method for visual localization. It is based on robust image retrieval for coarse camera pose estimation and robust local features for accurate pose refinement. Our method is top ranked on various public datasets showing its ability of generalization and its great variety of applications. To facilitate experiments, we introduce kapture, a flexible data format and processing pipeline for structure from motion and visual localization that is released open source. We furthermore provide all datasets used in this paper in the kapture format to facilitate research and data processing. Code and datasets can be found at `https://github.com/naver/kapture`*, more information, updates, and news can be found at* `https://europe.naverlabs.com/research/3d-vision/kapture`*.*

1. Introduction

Visual localization The goal of visual localization is to estimate the accurate position and orientation of a camera using its images. In detail, correspondences between a representation of the environment (map) and query images are utilized to estimate the camera pose in 6 degrees of freedom (DOF). The representation of the environment can be a structure from motion (SFM) reconstruction [43, 52, 19, 49, 34], a database of images [56, 53, 41], or even a CNN [24, 28, 7, 48]. Structure-based methods [43, 32, 44, 30, 53, 49] use local features to establish correspondences between 2D query images and 3D reconstructions. These correspondences are then used to compute the camera pose using perspective-n-point (PNP) solvers [25] within a RANSAC loop [17, 10, 29]. To reduce the search range in large 3D reconstructions, image retrieval methods can be used to first retrieve most relevant images from the SFM model. Second, local correspondences are established in the area defined by those images. Scene point regression methods [51, 8] establish the 2D-3D correspondences using a deep neural network (DNN) and absolute pose regression methods [24, 28, 7, 48] directly estimate the camera pose with a DNN. Furthermore, also objects can be used for visual localization, such as proposed in [58, 40, 11, 4].

Challenges Since in visual localization correspondences between the map and the query image need to be established, environmental changes present critical challenges. Such changes could be caused by time of day or season of the year, but also structural changes on house facades or store fronts are possible. Furthermore, the query images can be taken under significantly different viewpoints than the images used to create the map.

Long term visual localization To overcome these challenges, researchers proposed various ways to increase robustness of visual localization methods. Most relevant to our work are data-driven local [33, 15, 16, 38, 13] and global [1, 36, 37] features. Instead of manually describing how keypoints or image descriptions should look like, a large amount of data is used to train an algorithm to make this decision by itself. Recent advances in the field showed great results on tasks like image matching [35] and visual localization [41, 16, 39]. [45] provide an online benchmark which consists of several datasets covering a variety of the mentioned challenges.

In this paper, we present a robust image retrieval-based visual localization method. Extensive evaluations show that it reports top results on various public datasets which highlights its versatile application. We implemented our algorithm using our newly proposed data format and toolbox named *kapture*. The code is open source and all datasets from the website mentioned above are provided in this format.

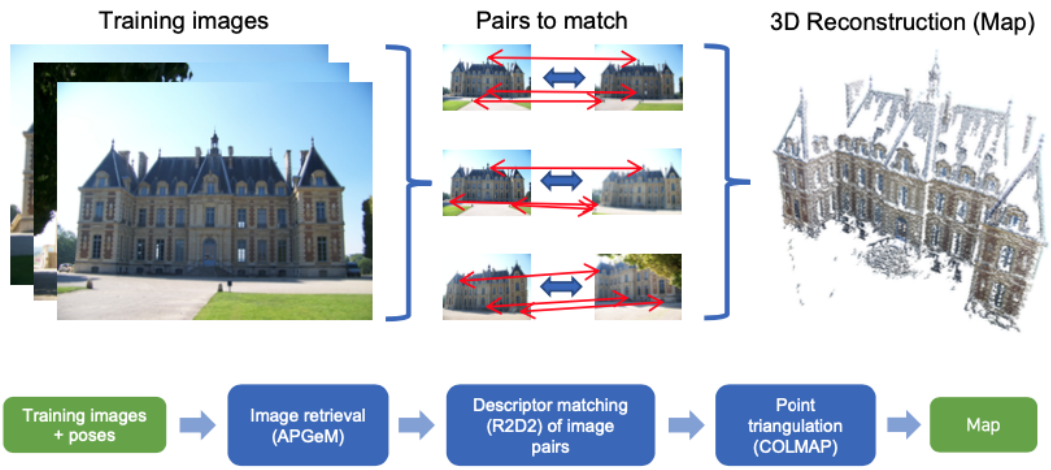


Figure 1. Overview of the structure from motion (SFM) reconstruction of the map from a set of training (mapping) images. Photos: Sceaux Castle image dataset[2]

2. Visual Localization Method

As a reminder, visual localization is the problem of estimating the 6DOF pose of a camera within a known 3D space representation using query images. There are several ways to tackle this problem including structure-based methods [43, 32, 44, 30, 53, 49], pose [24, 28, 7, 48] and scene point regression-based [51, 8] methods or image retrieval-based methods [55, 60, 56]. Our approach follows the workflow of image retrieval as well as structure-based methods and combines functionalities provided by the COLMAP SFM library[3] [49] as well as our local features R2D2 [38] and our global image representation AP-GeM [37]. The method consists of two main components: the SFM-based mapping pipeline (shown in Figure 1) and the localization (image registration) pipeline (shown in Figure 2).

Mapping SFM is one of the most popular strategies for reconstruction of a 3D scene from un-ordered photo collections [52, 19, 49, 34]. The main idea is to establish 2D-2D correspondences between local image features (keypoints) of mapping[4] image pairs, followed by geometric verification to remove outliers. By exploiting transitivity, observations of a keypoint can be found in several images allowing to apply relative pose estimation for initialization of the reconstruction followed by 3D point triangulation [23] and image registration for accurate 6DOF camera pose estimation. RANSAC [17, 10, 29] can be used to increase robustness of several steps in this pipeline and bundle adjustment [57] can be used for global (and local) optimization of the model (3D points and camera poses). Since the camera poses of the training images for all datasets used in this paper are known, our mapping pipeline can skip the camera pose estimation step of SFM. For geometric verification of the matches and triangulation of the 3D points, we use COLMAP. Figure 1 illustrates our mapping workflow.

Localization Similarly to the reconstruction step, 2D-2D local feature correspondences are established between a query image and the database images used to generate the map. In order to only match relevant images, we use image retrieval to obtain the most similar images (e.g. 20 or 50) from the database. Since many keypoints from the database images correspond to 3D points of the map, 2D-3D correspondences between query image and map can be established. These 2D-3D matches are then used to compute the 6DOF camera pose by solving a PNP problem [25, 26, 27] robustly inside a RANSAC loop [17, 10, 29]. We again use COLMAP for geometric verification and image registration.

Local descriptors We can see that both pipelines (mapping and localization) heavily rely on local image descriptors and matches. Early methods used handcrafted local feature extractors, notably the popular SIFT descriptor[5] [31]. However, those keypoint extractors and descriptors have several limitations including the fact that they are not necessary tailored to the target task. Therefore, several data-driven learned representations were proposed recently (see the evolution of local features in [13, 50]).

Our method uses R2D2 [38], which is a sparse keypoint extractor that jointly performs detection and description but separately estimates keypoint reliability and keypoint repeatability. Keypoints with high likelihoods on both aspects are chosen which improves the overall feature match-

[3] https://colmap.github.io

[3] https://github.com/openMVG/ImageDataset_SceauxCastle

[4] Also referred to as training images.

[5] as used in COLMAP

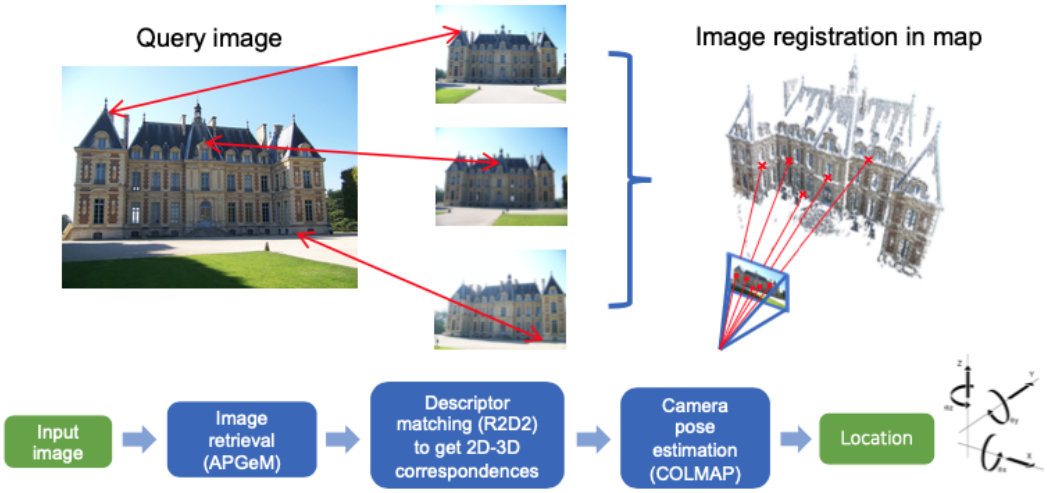


Figure 2. Overview of the localization pipeline which registers query images in the SFM map. Photos: Sceaux Castle image dataset[3]

ing pipeline. R2D2 uses a list-wise loss that directly maximises the average precision to learn reliability. Since a very large amount of image patches (only one is correct) is used per batch, the resulting reliability is well suited for the task of matching. Since reliability and patch descriptor are related, the R2D2 descriptor is extracted from the reliability network. The R2D2 model was trained with synthetic image pairs generated by known transformations (homographies) providing exact pixel matches as well as optical flow data from real image pairs. See Section 4 for details about the model.

Image retrieval In principle, mapping and localization can be done by considering all possible image pairs. However, this approach does not scale well to visual localization in real-world applications where localization might need to be done in large scale environments such as big buildings or even cities. To make visual localization scaleable, image retrieval plays an important role. On the one hand, it makes the mapping more efficient, on the other hand, it increases robustness and efficiency of the localization step [20, 42, 53]. This is achieved in two steps: First, the global descriptors are matched in order to find the most similar images which form image pairs (e.g. reference-reference for mapping and query-reference for localization). Second, these image pairs are used to establish the local keypoint matches.

Localization approaches based on image retrieval typically use retrieval representations designed for geo-localization [1, 54, 2]. However, our initial experiments have not shown superiority of these features compared to our off-the-shelf deep visual representations Resnet101-AP-GeM [37]. Note that our model was trained for the landmark retrieval task on the Google Landmarks (GLD) dataset [33]. The model considers a generalized mean-pooling (GeM) layer [36] to aggregate the feature maps into a compact, fixed-length representation which is learned by directly optimizing the mean average precision (mAP). Section 5 contains more details about AP-GeM as well as comparisons of various global image representations and combinations of them.

3. Kapture

3.1. Kapture format and toolbox

When running a visual localization pipeline on several datasets, one of the operational difficulties is to convert those datasets into a format that the algorithm and all the tools used can handle. Many formats already exist, notably the ones from Bundler[6], VisualSFM[7], OpenMVG[8], OpenSfM[9], and COLMAP[10], but none meets all our requirements. In particular we need a format that can handle timestamps, shared camera parameters, multi-camera rigs, but also reconstruction data (keypoints, descriptors, global features, 3D points, matches...) and that would be flexible and easy to use for localization experiments. Furthermore, it should be easy to convert data into other formats supported by major open source projects such as OpenMVG and COLMAP.

Inspired by the mentioned open source libraries, kapture started as pure data format that provids a good representation of all the information we needed. It then grew into a Python toolbox and library for data manipulation (conversion between various popular formats, dataset merging/s-

[6] https://www.cs.cornell.edu/~snavely/bundler/bundler-v0.4-manual.html#S6
[7] http://ccwu.me/vsfm/doc.html#nvm
[8] https://openmvg.readthedocs.io/en/latest/software/SfM/SfM_OutputFormat/
[9] https://www.opensfm.org/docs/dataset.html/
[10] https://colmap.github.io/format.html

plitting, trajectory visualization, etc.), and finally it became the basis for our mapping and localization pipeline. More precisely, the kapture format can be used to store sensor data: images, camera parameters, camera rigs, trajectories, but also other sensor data like lidar or wifi scans. It can also be used to store reconstruction data, in particular local descriptors, keypoints, global features, 3D points, observations, and matches.

We believe that the kapture format and tools are useful for the community, so we release them as open-source at `https://github.com/naver/kapture`. We also provide major public datasets of the domain in this format to facilitate future experiments for everybody.

3.2. Kapture pipeline

We implemented our visual localization method, described in Section 2, on top of the kapture tools and libraries. In particular, the mapping pipeline consists of the following steps:

1. Extraction of local descriptors and keypoints (e.g. R2D2) of training images
2. Extraction of global features (e.g. AP-GeM) of training images
3. Computation of training image pairs using image retrieval
4. Computation of local descriptor matches between these image pairs
5. Geometric verification of the matches and point triangulation with COLMAP

The localization steps are similar:

1. Extraction of local and global features of query images
2. Retrieval of similar images from the training images
3. Local descriptor matching
4. Geometric verification of the matches and camera pose estimation with COLMAP

4. Evaluation

For evaluation of our method, we use the datasets provided by the online visual localization benchmark[11] [45]. Each of these datasets is split into a training (mapping) and a test set. The training data, which consists of images, corresponding poses in the world frame as well as intrinsic camera parameters, is used to construct the map, the test data is used to evaluate the precision of the localization method. Intrinsic parameters of the test images are not always provided.

We converted all datasets to kapture and we used the publicly available models for R2D2[12] and AP-GeM[13] for all datasets and evaluations. If not indicated differently, we used the top 20k keypoints extracted with R2D2.

Parameters We experimented with three COLMAP parameter settings which are presented in Table 1. For map generation we always used *config1*.

Metrics All datasets used are divided into different conditions. These conditions could be different times of day, differences in weather such as snow, or even different buildings or locations within the dataset. In order to report localization results, we used the online benchmark[11] which computes the percentage of query images which where localized within three pairs of translation and rotation thresholds.

4.1. Aachen Day-Night

The Aachen Day-Night dataset [45, 47] represents an outdoor handheld camera localization scenario where all query images are taken individually with large changes in viewpoint and scale, but also between daytime and nighttime. In detail, the query images are divided into the classes *day* and *night* and the two classes are evaluated separately. We evaluated our method in two settings: (i) we used the *full* dataset to construct a single map using the provided reference poses and localized all query images within this map, and (ii) we used the *pairs*[14] provided for the local features evaluation task on the online benchmark[11], which cover nighttime images only. Recently, an updated version Aachen Day-Night v.1.1 [61], which contains more training images and more accurate poses of the query images (not public), was released. Table 2 presents the results for both versions of the dataset.

4.2. InLoc

InLoc [53, 59] is a large indoor dataset for visual localization. It also represents a handheld camera scenario with large viewpoint changes, occlusions, people and even changes in furniture. Contrary to the other datasets, InLoc also provides 3D scan data, i.e. 3D point clouds for each training image. However, since the overlap between the training images is quite small, the resulting structure from motion models are sparse and, according to our experience, not suitable for visual localization. Furthermore, the InLoc

[11] `http://visuallocalization.net`

[12] r2d2_WASF_N8_big from `https://github.com/naver/r2d2`

[13] Resnet101-AP-GeM-LM18 from `https://github.com/almazan/deep-image-retrieval`

[14] `https://github.com/tsattler/visuallocalizationbenchmark`

Table 1. Parameter configurations.

COLMAP image_registrator	config1	config2	config3
−−Mapper. ba_refine_focal_length	0	0	1
−−Mapper. ba_refine_principal_point	0	0	0
−−Mapper.ba_refine_extra_params	0	0	1
−−Mapper.min_num_matches	15	4	4
−−Mapper.init_min_num_inliers	100	4	4
−−Mapper.abs_pose_min_num_inliers	30	4	4
−−Mapper. abs_pose_min_inlier_ratio	0.25	0.05	0.05
−−Mapper.ba_local_max_num_iterations	25	50	50
−−Mapper.abs_pose_max_error	12	20	20
−−Mapper. filter_max_reproj_error	4	12	12

Table 2. Results on Aachen Day-Night. In *pairs* we used the top 40k R2D2 keypoints. Day: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°), Night: (0.5m, 2°) / (1m, 5°) / (5m, 10°)

v1 setting	day	night
full (config2)	88.7 / 95.8 / 98.8	81.6 / 88.8 / 96.9
pairs (config1, R2D2 40k)	-	76.5 / 90.8 / 100.0
v1.1 setting	**day**	**night**
full (config2)	90.0 / 96.2 / 99.5	72.3 / 86.4 / 97.9
pairs (config1, R2D2 40k)	-	71.2 / 86.9 / 97.9

Table 3. Results on InLoc using different 3D point distance thresholds for mapping. (0.25m, 10°) / (0.5m, 10°) / (5m, 10°)

setting	DUC1	DUC2
config2, 0.5mm	36.4 / 52.0 / 64.1	30.5 / 53.4 / 58.0
config2, 0.1mm	36.9 / 53.0 / 65.7	34.4 / 52.7 / 59.5

environment is very challenging for global and local features because it contains large textureless areas and many repetitive, non-unique areas. To overcome these problems, the original InLoc localization method [53] introduced various dense matching and pose verification methods which make use of the provided 3D data.

Mapping We constructed our SFM map using the provided 3D data and the camera poses, which differs from the mapping described in Section 2. We first assign a 3D point to each local feature in the training images. Second, we generate matches based on 3D points. In detail, we look for local features which are the projection of the same 3D point in different images. To decide whether or not a 3D point is the same for different keypoints, we use an Euclidean distance threshold (0.5mm and 0.1mm). This results in a very dense 3D map (Figure 3) where each 3D point is associated with a local descriptor and can, thus, be used in our method.

Localization We ran the localization pipeline (Figure 2) for all provided query images. Table 3 presents the results.

4.3. RobotCar Seasons

RobotCar Seasons [45] is an outdoor dataset captured in the city of Oxford at various periods of a year and in different conditions (rain, night, dusk, etc.). The images are taken from a car with a synchronized three-camera rig pointing in three directions (rear, left and right). The data was captured at 49 different non-overlapping locations and several 3D models are provided. Training images were captured with a reference condition (overcast-reference), while test images were captured in different conditions. For each test image, the dataset provides its condition, the location where it was captured (one of the 49 location used in the training data), its timestamp, and the camera name.

Mapping Since the different locations are not overlapping, there is no benefit in building a single map. For our experiments, we used the individual models for each 49 locations that are provided in the COLMAP format. We converted the COLMAP files into the kapture format to recover trajectories (poses and timestamps) and created 49 individual maps using our mapping pipeline (Figure 1). For this step, we used the provided camera parameters (pinhole model) and considered each camera independently without using the rig information.

Localization Since the location within the dataset is given for each query image, we can directly use it during localization. Otherwise, we would have first selected the correct map, e.g. by using image retrieval. We tested both, COLMAP config1 and config2.

For the images that could not be localized we ran two additional steps. First, we leverage the fact that images are captured synchronously with a rig of three cameras for which the calibration parameters are provided. Hence, if one image taken at a specific timestamp is localized, using the provided extrinsic camera parameters we can compute

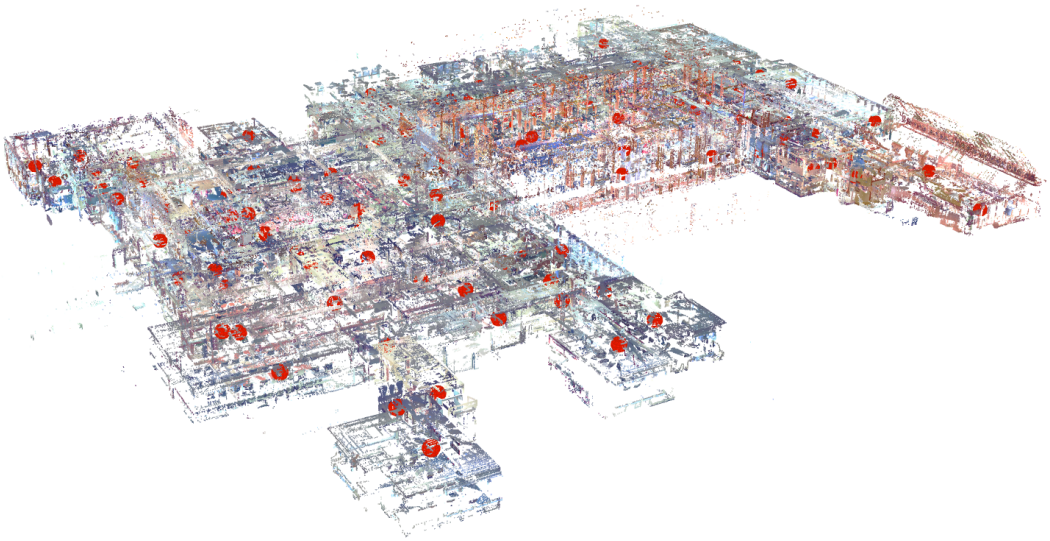


Figure 3. InLoc map generated by assigning a 3D point to each R2D2 feature in the training images (viewed in COLMAP).

the pose for all images of the rig (even if they were not successfully localized). We used this technique to find the missing poses for all images for which this can be applied.

However, there are still timestamps for which no pose was found for any of the three cameras. In this case, we leverage the fact that query images are given in sequences (e.g. 6 to 12 images in most cases). Sequences can be found using image timestamps. When the gap between two successive timestamps is too large (i.e. above a certain threshold), we start a new sequence. Once the sequences are defined, we look for non-localized image triplets in these sequences and estimate their poses by linear interpolation between the two closest successfully localized images. If this is not possible, we use the closest available pose. Note that for real-world applications, we could either only consider images of the past or introduce a small latency if images from both directions (before and after) are used. These steps increase the percentage of localized images to 97.2%. Table 4 presents the results of the configurations tested. Interestingly, even if config2 could localize all images and config1 only 90%, applying the rig and sequence information on config1 led to overall better results.

Table 4. Results on RobotCar Seasons. Thresholds: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°)

setting	day	night
config2	55.2 / 82.0 / 97.1	28.1 / 59.0 / 82.7
config1	55.1 / 82.1 / 96.9	26.9 / 55.6 / 78.4
config1 + rig	55.1 / 82.1 / 97.2	28.7 / 58.3 / 83.4
config1 + rig + seq	55.1 / 82.1 / 97.3	28.8 / 58.8 / 89.4

4.4. Extended CMU-Seasons

The Extended CMU-Seasons dataset [45, 5] is an autonomous driving dataset that contains sequences from urban, suburban, and park environments. The images were recorded in the area of Pittsburgh, USA over a period of one year and thus contain different conditions (foliage/mixed-foliage/no foliage, overcast, sunny, low sun, cloudy, snow). The training and query images were captured by two front-facing cameras mounted on a car, pointing to the left and right of the vehicle at approximately 45 degrees with respect to the longitudinal axis. The cameras are not synchronized. This dataset is also split into multiple locations. Unlike RobotCar Seasons, there is some overlap between the locations. However, we did not leverage this in our experiments.

Mapping For our experiments, we used the individual models for each location. We converted the ground-truth-database-images-sliceX.txt files into the kapture format to recover trajectories (poses and timestamps). We then created 14 individual maps (the slices that were provided with queries 2-6/13-21) using the pipeline described above. For this step, we used the provided camera parameters (OpenCV[15] pinhole camera), and considered each camera independently, i.e. without using the rig information.

Localization We ran the localization pipeline described above on all images listed in the test-images-sliceX.txt files with config1. We then ran two post-processing steps: rig and sequence. For rig, we first estimated a rig configuration from the slice2 training poses. For all images that

[15] https://opencv.org

failed to localize, we computed the position using this rig if the image from the other camera with the closest timestamp was successfully localized. Finally, we applied the same sequence post-processing as we did for RobotCar Seasons (see Section 4.3). Table 5 presents the results on this dataset and the improvements we get from each of the post-processing steps.

Table 5. Results on Extended CMU-Seasons. All conditions: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°)

setting	urban	suburban	park
config2	95.9 / 98.1 / 98.9	89.5 / 92.1 / 95.2	78.3 / 82.0 / 86.4
config1	95.8 / 98.1 / 98.8	88.9 / 91.1 / 93.4	75.5 / 78.4 / 82.0
config1 + rig	96.5 / 98.8 / 99.5	94.3 / 96.7 / 99.1	83.1 / 87.9 / 92.8
config1 + rig + seq	96.7 / 98.9 / 99.7	94.4 / 96.8 / 99.2	83.6 / 89.0 / 95.5

4.5. SILDa Weather and Time of Day

SILDa Weather and Time of Day [6] is an outdoor dataset captured over a period of 12 months (clear, snow, rain, noon, dusk, night) which covers 1.2km of streets around Imperial College in London. It was captured using a camera rig composed of two back-to-back wide-angle fisheye lenses. The geometry of the rig as well as the hardware synchronization of the acquisition could be leveraged, e.g. to reconstruct spherical images.

Mapping The dataset provides camera parameters corresponding to a fisheye model that is not available in COLMAP. For the sake of simplicity, we chose to estimate the parameters of both cameras using a camera model supported by COLMAP, namely the FOV model (we still use the provided estimation of the principal point).

Localization Similarly to the RobotCar Seasons dataset, we applied the image sequences and camera rig configuration to estimate camera poses of images which could not be localized. As the rig geometry is not given for SILDa, we estimated an approximation. Table 6 presents the results of the configurations used. As can be seen, leveraging the sequence did not improve the results.

Table 6. Results on SILDa. Thresholds: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°)

setting	evening	snow	night
config1	31.8 / 66.3 / 89.4	0.3 / 3.9 / 64.9	30.0 / 53.4 / 77.5
config1 + rig	31.9 / 66.6 / 92.5	0.5 / 5.8 / 89.2	30.5 / 54.2 / 78.5
config1 + rig + seq	31.9 / 66.6 / 92.5	0.5 / 5.8 / 89.2	30.5 / 54.2 / 78.5

5. Experiments with late fusion of multiple image representations

The motivation of these experiments is finding a late fusion strategy where the combination of different global descriptors outperforms each descriptor individually. To do so, we replaced the global descriptor AP-GeM, we used for image retrieval described in Section 3.2, with combinations of 4 different global descriptors:

DenseVLAD [54] To obtain the DenseVLAD representation for an image, first RootSIFT [3] descriptors are extracted on a multi-scale, regular, densely sampled grid, and then, aggregated into an intra-normalized VLAD [22] descriptor followed by PCA compression, whitening, and L2 normalization [21]. DenseVLAD is often used in structure-based visual localization methods to scale them up to large scenes [54, 46].

NetVLAD [1] The main component of the NetVLAD architecture is a generalized VLAD layer that aggregates mid-level convolutional features extracted from the entire image into a compact single vector representation for efficient indexing similarly to VLAD [22]. The resulting aggregated representation is then compressed using PCA to obtain a final compact descriptor for the image. NetVLAD is trained with geo-tagged image sets consisting of groups of images taken from the same locations at different times and seasons, allowing the network to discover which features are useful or distracting and what changes should the image representation be robust to. This makes NetVLAD very interesting for the visual localization pipeline. Furthermore, NetVLAD has already been used in state-of-the-art localization pipelines [41, 18] and in combination with D2-Net [16].

AP-GeM [37] This model, similarly to [36], uses a generalized-mean pooling layer (GeM) to aggregate CNN-based descriptors of several image regions at different scales. Instead of contrastive loss, it directly optimizes the Average Precision (AP) approximated by histogram binning to make it differentiable. It is currently one of the best performing methods of image representation on popular landmark retrieval benchmarks such as $\mathcal{R}$Oxford and $\mathcal{R}$Paris [35].

DELG [9] DELG is a 2-in-1 local and global features extraction CNN. After a common backbone, the model is split into two parts (heads), one to detect relevant local features and one to describe the global content of the image as a compact descriptor. The two networks are jointly trained on Google Landmark v1 [33] in an end-to-end manner using the ArcFace [14] loss for the compact descriptor. The method is originally designed for image search where the local features enable geometric verification and re-ranking.

5.1. Fusion

Late fusion means that we first compute the similarities for each descriptor individually and then apply a fusion operator, such as:

Generalized harmonic mean (GHarm) [12] GHarm is a generalization of the weighted harmonic mean (WHarm) [16]. It can be obtained using the generalized f-mean:

$$M_f(x_1, ..., x_n) = f^{-1}(\sum_{i=1}^{n} \frac{f(x_i)}{n}) \quad (1)$$

with $f = \frac{1}{\gamma + x}, x_i = \alpha_i sim_i, \sum_{i=1}^{n} \alpha_i = 1$.

round_robin[17] Most similar images are picked from each individual descriptor in equal portions and in circular order.

mean_and_power (WMP) [12]

$$M_{WMP} = \gamma \cdot \sum_{i=1}^{n} \alpha_i sim_i + (1 - \gamma) \cdot \prod_{i=1}^{n} sim_i^{\alpha_i} \quad (2)$$

min_and_max [12]

$$M_{Min\&Max} = (1 - \alpha) \max_{i=1}^{n}(sim_i) + \alpha \min_{i=1}^{n}(sim_i) \quad (3)$$

min [12]

$$M_{Min} = \min_{i=1}^{n}(sim_i) \quad (4)$$

max [12]

$$M_{Max} = \max_{i=1}^{n}(sim_i) \quad (5)$$

5.2. Parameters

For AP-GeM[18], we used the Resnet101-AP-GeM model trained on Google Landmarks v1 [33]. For DELG[19], the model is also trained on Google Landmarks v1. [33]. For NetVLAD[20], we used the VGG-16-based NetVLAD model trained on Pitts30k. DenseVLAD is available at `http://www.ok.ctrl.titech.ac.jp/~torii/project/247/`. We used GHarm [12] with $\alpha_i = \frac{1}{n}$ $\gamma = 0.5$ and for all other operators we used equal weights.

[16] `https://en.wikipedia.org/wiki/Harmonic_mean`

[17] `https://en.wikipedia.org/wiki/Round-robin_scheduling`

[18] AP-GeM code at `https://github.com/almazan/deep-image-retrieval`

[19] DELG code at `https://github.com/tensorflow/models/tree/master/research/delf/delf/python/delg`

[20] NetVLAD code at `https://github.com/Relja/netvlad`

5.3. Aachen Day-Night v1.1

As we already did for the Aachen Day-Night experiments in Section 4.1, we evaluated our method with two settings: (i) we used the *full* dataset to construct a single map using the provided reference poses and localized all query images within this map, and (ii) we used the *pairs*[21] provided for the local features evaluation task on the online benchmark[11], which cover nighttime images only. Table 7 presents the results.

Table 7. Results on Aachen Day-Night v1.1. In all experiments but *pairs* and *gharm top50*, we used the top 20k R2D2 keypoints. All conditions: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°)

setting	day	night
pairs (r2d2 40k) config1	-	71.2 / 86.9 / 97.9
gharm top50 (r2d2 40k) config2	**90.9** / 96.7 / **99.5**	**78.5** / **91.1** / 98
gharm top20 config2	90.5 / **96.8** / 99.4	74.9 / 90.1 / **98.4**
AP-GeM-LM18 top20 config2	89.9 / 96.5 / **99.5**	71.2 / 86.9 / 97.9
DELG top20 config2	90.0 / 96.0 / 99.2	73.3 / 87.4 / 97.4
netvlad_vd16pitts top20 config2	88.7 / 95.1 / 98.1	74.3 / 89.5 / 97.9
densevlad top20 config2	88.2 / 94.2 / 97.3	62.8 / 79.1 / 88.0
round_robin top20 config2	89.6 / 96.6 / 99.4	72.8 / 87.4 / **98.4**
mean_and_power top20 config2	90.7 / 96.8 / 99.3	73.3 / 89.5 / **98.4**
min_and_max top20 config2	89.9 / 96.4 / 99.2	73.3 / 88.0 / 97.4
min top20 config2	89.1 / 95.0 / 98.2	71.2 / 84.8 / 92.1
max top20 config2	89.2 / 96.0 / 99.0	73.8 / 88.0 / 97.9

5.4. InLoc

We ran the localization pipeline described in Section 4.2 for all provided query images. Table 8 presents the results.

Table 8. Results on InLoc using 0.1mm 3D point distance threshold for mapping. (0.25m, 10°) / (0.5m, 10°) / (5m, 10°). In all experiments we used the top 40k R2D2 keypoints.

setting	DUC1	DUC2
gharm top50 config2	**41.4** / **60.1** / **73.7**	47.3 / **67.2** / **73.3**
AP-GeM-LM18 top50 config2	37.4 / 55.6 / 70.2	36.6 / 51.9 / 61.1
DELG top50 config2	38.4 / 56.1 / 71.7	37.4 / 59.5 / 67.9
netvlad_vd16pitts top50 config2	36.9 / 58.1 / 70.2	38.2 / 62.6 / 70.2
densevlad top50 config2	33.8 / 51.5 / 67.7	45.0 / 65.6 / 72.5
round_robin top50 config2	33.8 / 51.5 / 66.2	29.8 / 48.1 / 53.4
mean_and_power top50 config2	39.9 / 57.6 / 71.7	**48.9** / **67.2** / **73.3**
min_and_max top50 config2	37.9 / 56.6 / 70.7	44.3 / 62.6 / 69.5
min top50 config2	32.8 / 54.5 / 67.7	42.7 / 65.6 / 71.8
max top50 config2	39.9 / 55.1 / 69.2	38.9 / 58.0 / 64.1

5.5. RobotCar Seasons v2

For this experiment, we used the newly released v2 of the RobotCar Seasons dataset. The new version is based on the same data, but uses a different split for training and testing. In addition, the training data (images and poses) of v2 contains images from various conditions (contrary to the orig-

[21] `https://github.com/tsattler/visuallocalizationbenchmark`

inal dataset where only condition overcast-reference was available). We created 22 individual maps (the locations that were provided with queries 4-6/23-36,44,45,47/49) using the pipeline described above. Table 9 presents the results.

Table 9. Results on RobotCar Seasons v2. Thresholds: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°). In all experiments we used the top 20k R2D2 keypoints.

setting	day	night
gharm top20 config1 + rig + seq	**66.0 / 95.1 / 100.0**	46.2 / 76.5 / 91.4
AP-GeM-LM18 top20 config1 + rig + seq	65.7 / **95.1 / 100.0**	43.6 / 76.7 / 93.9
DELG top20 config1 + rig + seq	65.6 / 94.6 / 99.6	37.8 / 64.6 / 78.8
netvlad_vd16pitts top20 config1 + rig + seq	65.6 / **95.1 / 100.0**	35.7 / 70.4 / 90.9
densevlad top20 config1 + rig + seq	65.7 / **95.1 / 100.0**	41.3 / 74.1 / 92.8
round_robin top20 config1 + rig + seq	65.9 / **95.1 / 100.0**	42.4 / 75.1 / **94.2**
mean_and_power top20 config1 + rig + seq	65.9 / **95.1 / 100.0**	**46.4 / 79.7** / 92.5
min_and_max top20 config1 + rig + seq	65.7 / **95.1 / 100.0**	41.3 / 72.5 / 86.2
min top20 config1 + rig + seq	65.8 / **95.1 / 100.0**	40.8 / 72.0 / 91.6
max top20 config1 + rig + seq	65.6 / 94.7 / 99.7	36.6 / 63.2 / 80.0

5.6. Extended CMU-Seasons

We created 14 individual maps (the slices that were provided with queries 2-6/13-21) using the pipeline described above. Table 10 presents the results on this dataset.

Table 10. Results on Extended CMU-Seasons. All conditions: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°). In all experiments we used the top 20k R2D2 keypoints.

setting	urban	suburban	park
gharm top20 config1 + rig + seq	97.0 / 99.1 / **99.8**	**95.0** / 97.0 / 99.4	**89.2 / 93.4** / 97.5
AP-GeM-LM18 top20 config1 + rig + seq	96.7 / 98.9 / 99.7	94.4 / 96.8 / 99.2	83.6 / 89.0 / 95.5
DELG top20 config1 + rig + seq	96.6 / 98.8 / 99.7	94.1 / 96.7 / 99.1	84.7 / 89.6 / 95.7
netvlad_vd16pitts top20 config1 + rig + seq	**97.1** / 99.1 / **99.8**	93.8 / 96.3 / 99.1	88.1 / 92.7 / 97.5
densevlad top20 config1 + rig + seq	96.1 / 98.4 / 99.4	94.2 / 96.7 / 99.1	87.9 / 92.3 / 96.8
round_robin top20 config1 + rig + seq	96.9 / 99.1 / 99.7	94.8 / 97.0 / 99.4	88.8 / 93.1 / **97.6**
mean_and_power top20 config1 + rig + seq	97.0 / **99.2** / 99.7	94.7 / 97.0 / **99.5**	**89.2 / 93.4** / 97.5
min_and_max top20 config1 + rig + seq	96.7 / 98.9 / 99.6	**95.0 / 97.1 / 99.5**	88.6 / 93.2 / **97.6**
min top20 config1 + rig + seq	96.3 / 98.5 / 99.4	94.2 / 96.6 / 99.2	88.0 / 92.4 / 97.3
max top20 config1 + rig + seq	96.8 / 98.9 / 99.7	94.4 / 97.0 / 99.3	84.1 / 89.2 / 96.0

5.7. SILDa Weather and Time of Day

Table 11 presents the results of the configurations used.

Table 11. Results on SILDa. Thresholds: (0.25m, 2°) / (0.5m, 5°) / (5m, 10°). In all experiments we used the top 20k R2D2 keypoints

setting	evening	snow	night
gharm top20 config1 + rig + seq	**32.4** / 67.4 / 93.3	0.2 / 4.1 / 88.9	30.4 / 54.2 / 81.1
AP-GeM-LM18 top20 config1 + rig + seq	31.9 / 66.6 / 92.5	**0.5** / 5.8 / 89.2	**30.5** / 54.2 / 78.5
DELG top20 config1 + rig + seq	31.3 / 66.4 / 92.1	0.2 / **7.5** / 85.6	30.2 / 54.1 / 77.4
netvlad_vd16pitts top20 config1 + rig + seq	31.6 / 66.5 / 91.0	0.0 / 2.7 / **89.6**	28.9 / 52.1 / 78.5
densevlad top20 config1 + rig + seq	27.9 / 59.7 / 75.6	0.0 / 1.9 / 59.1	29.3 / 52.9 / 79.7
round_robin top20 config1 + rig + seq	31.8 / 66.9 / 92.0	0.0 / 3.4 / **89.6**	29.3 / 54.0 / 79.5
mean_and_power top20 config1 + rig + seq	31.7 / 67.0 / **93.5**	0.0 / 2.7 / 88.5	30.4 / **54.4** / 80.7
min_and_max top20 config1 + rig + seq	32.1 / **67.6** / 93.4	0.2 / 3.3 / 88.7	**30.5** / 54.0 / 80.2
min top20 config1 + rig + seq	30.1 / 63.9 / 82.5	0.0 / 1.9 / 76.2	28.1 / 54.1 / **81.4**
max top20 config1 + rig + seq	31.8 / 65.9 / 92.4	0.3 / 2.7 / 84.6	30.4 / 54.1 / 77.8

5.8. Discussion

In summary, the improvement over the individual features, especially AP-GeM, is not too large. The reason is that AP-GeM already performs quite well and it is sufficient for our localization pipeline to retrieve just a few good images. However, we observe a consistent improvement for all of our datasets using the GHarm operator. As expected, the improvement is more significant for cases where the performance with individual descriptors is lower, such as nighttime or indoor images.

6. Conclusion and Future Work

We presented a versatile method for visual localization based on robust global features for coarse localization using image retrieval and robust local features for accurate pose computation. We evaluated our method on multiple datasets covering a large variety of application scenarios and challenging situations. Our method ranks among the best methods on the online visual localization benchmark[11]. We implemented our method in Python and ran the experiments using kapture, a unified SFM and localization data format which we released open source. Since all datasets are available in this format, we hope to facilitate future large scale visual localization and structure from motion experiments using a multitude of datasets. Finally, we showed that late fusion of global image descriptors is a promising direction to improve our method.

References

[1] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomáš Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *CVPR*, 2016.

[2] Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomáš Pajdla. Dislocation: Scalable Descriptor Distinctiveness for Location Recognition. In *ACCV*, 2014.

[3] Relja Arandjelović and Andrew Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. In *CVPR*, 2012.

[4] Shervin Ardeshir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. Gis-assisted object detection and geospatial localization. In *European Conference on Computer Vision*, pages 602–617. Springer, 2014.

[5] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU Visual Localization Data Set. `http://3dvis.ri.cmu.edu/data-sets/localization`, 2011.

[6] Vasileios Balntas, Duncan Frost, Rigas Kouskouridas, Axel Barroso-Laguna, Arjang Talattof, Huub Heijnen, and Krystian Mikolajczyk. Silda: Scape imperial localisation dataset. `https://www.visuallocalization.net/`, 2019.

[7] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation Using Neural Nets. In *ECCV*, 2018.

[8] Eric Brachmann and Carsten Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[9] Bingyi Cao, André Araujo, and Jack Sim. Unifying Deep Local and Global Features for Efficient Image Search. *arXiv*, 2001.05027, 2020.

[10] Ondřej Chum and Jiří Matas. Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008.

[11] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-outdoor 3d reconstruction alignment. In *European Conference on Computer Vision*, pages 285–300. Springer, 2016.

[12] Gabriela Csurka and Stephane Clinchant. An empirical study of fusion operators for multimodal image retrieval. In *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*.

[13] Gabriela Csurka, Christopher R. Dance, and Martin Humenberger. From Handcrafted to Deep Local Invariant Features. *arXiv*, 1807.10254, 2018.

[14] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. 01 2018.

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised Interest Point Detection and Description. In *CVPR*, 2018.

[16] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: a Trainable CNN for Joint Description and Detection of Local Features. In *CVPR*, 2019.

[17] M. Fischler and R. Bolles. Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of the ACM*, 24:381–395, 1981.

[18] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization. In *International Conference on 3D Vision (3DV)*, 2019.

[19] J. Heinly, J. L. Schönberger, E. Dunn, and J. M. Frahm. Reconstructing the world* in six days. In *CVPR*, 2015.

[20] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009.

[21] Hervé Jégou and Ondřej Chum. Negative Evidences and Co-occurrences in Image Retrieval: the Benefit of PCA and Whitening. In *ECCV*, 2012.

[22] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating Local Descriptors Into a Compact Image Representation. In *CVPR*, 2010.

[23] Lai Kang, Lingda Wu, , and Yee-Hong Yang. Robust multiview L2 triangulation via optimal inlier selection and 3D structure refinement. *PR*, 47(9):2974–2992, 2014.

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: a Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*.

[25] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A Novel Parametrization of the Perspective-three-point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *CVPR*, 2011.

[26] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *ICCV*, 2013.

[27] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making Minimal Solvers for Absolute Pose Estimation Compact and Robust. In *ICCV*, 2017.

[28] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. 2017.

[29] Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012.

[30] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map. In *ICCV*, 2017.

[31] David G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[32] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. In *ICCV*, 2013.

[33] Hyeonwoo Noh, André Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *ICCV*, 2017.

[34] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305364, 2017.

[35] Filip Radenović, Asmet Iscen, Giorgos Tolias, and Ondřej Avrithis, Yannis Chum. Revisiting Oxford and Paris: Large-scale Image Retrieval Benchmarking. In *CVPR*, 2018.

[36] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-Tuning CNN Image Retrieval with no Human Annotation. *PAMI*, 41(7):1655–1668, 2019.

[37] Jérôme Revaud, Jon Almazan, Rafael Sampaio de Rezende, and Cesar Roberto de Souza. Learning with Average Precision: Training Image Retrieval with a Listwise Loss. In *ICCV*, 2019.

[38] Jérôme Revaud, Philippe Weinzaepfel, César De Souza, and Martin Humenberger. R2D2: Reliable and Repeatable Detectors and Descriptors. In *NeurIPS*, 2019.

[39] Jérôme Revaud, Philippe Weinzaepfel, César De Souza, Noé Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Reliable and Repeatable Detectors and Descriptors for Joint Sparse Keypoint Detection and Local Feature Extraction. *CoRR*, (arXiv:1906.06195), 2019.

[40] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[41] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019.

[42] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *ICCV*, 2015.

[43] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011.

[44] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 39(9):1744–1756, 2017.

[45] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DoF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018.

[46] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *CVPR*, 2017.

[47] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference (BMCV)*, 2012.

[48] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, 2019.

[49] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *CVPR*, 2016.

[50] Johannes L. Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *CVPR*, 2017.

[51] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *CVPR*, 2013.

[52] N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 80(2):189–210, 2008.

[53] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *PAMI*, pages 1–1, 2019.

[54] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomáš Pajdla. 24/7 Place Recognition by View Synthesis. In *CVPR*, 2015.

[55] Akihiko Torii, Josef Sivic, and Tomáš Pajdla. Visual Localization by Linear Combination of Image Descriptors. In *ICCV-W*, 2011.

[56] A. Torii, H. Taira, J. Sivic, M. Pollefeys, M. Okutomi, T. Pajdla, and T. Sattler. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? *PAMI*, pages 1–1, 2019.

[57] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustmenta modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[58] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[59] Erik Wijmans and Yasutaka Furukawa. Exploiting 2d floorplan for building-scale panorama rgbd alignment. In *Computer Vision and Pattern Recognition, CVPR*, 2017.

[60] Amir Roshan Zamir and Mubarak Shah. Accurate Image Localization Based on Google Maps Street View. In *ECCV*, 2010.

[61] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis. *arXiv*, 2005.05179, 2020.

Which is Plagiarism: Fashion Image Retrieval based on Regional Representation for Design Protection

Yining Lang[1], Yuan He[1], Fan Yang[1], Jianfeng Dong[2,3], Hui Xue[1] *
Alibaba Group[1], Zhejiang Gongshang Univresity[2],
Alibaba-Zhejiang University Joint Institute of Frontier Technologies[3]

Abstract

With the rapid growth of e-commerce and the popularity of online shopping, fashion retrieval has received considerable attention in the computer vision community. Different from the existing works that mainly focus on identical or similar fashion item retrieval, in this paper, we aim to study the plagiarized clothes retrieval which is somewhat ignored in the academic community while itself has great application value. One of the key challenges is that plagiarized clothes are usually modified in a certain region on the original design to escape the supervision by traditional retrieval methods. To relieve it, we propose a novel network named Plagiarized-Search-Net (PS-Net) based on regional representation, where we utilize the landmarks to guide the learning of regional representations and compare fashion items region by region. Besides, we propose a new dataset named Plagiarized Fashion for plagiarized clothes retrieval, which provides a meaningful complement to the existing fashion retrieval field. Experiments on Plagiarized Fashion dataset verify that our approach is superior to other instance-level counterparts for plagiarized clothes retrieval, showing a promising result for original design protection. Moreover, our PS-Net can also be adapted to traditional fashion retrieval and landmark estimation tasks and achieves the state-of-the-art performance on the DeepFashion and DeepFashion2 datasets.

1. Introduction

Fashion-related works have attracted increasing attention, due to the boom of online shopping in these years. The rapid growth of deep learning-based approaches further enhances the ability of fashion image classification [30, 34], fashion landmark detection [39, 27], and fashion retrieval [45, 4, 49, 31]. The traditional clothes retrieval methods [26, 16] typically perform similarity learning in the entire instance of clothes without any focus, which is easily interfered by irrelevant features. The recent clothes retrieval methods [2, 4, 20, 49] learn the attribute representations to guide the retrieval, thus improve the performance.

Figure 1. Examples for identical, similar, and plagiarized clothes with respect to the original item.

Different from the exiting methods [26, 45, 2, 4, 49] typically aim to retrieve visually similar or identical clothes, we focus on a novel problem of *plagiarized clothes* retrieval. The plagiarized clothes retrieval is somewhat ignored in the academic community, while it has great application value in the industry. The similar clothes retrieval task is somewhat similar to the plagiarized clothes retrieval task, as some retrieved similar items may be plagiarized one. However, plagiarized items are not always very similar to the original fashion items. As shown in Figure 1, the plagiarized item is relatively more dissimilar than the similar item with the original one. Hence, the retrieved target of both tasks is different. Moreover, in the plagiarized clothes retrieval task, the ground-truth images may be in a different category with the original item (a long-sleeved T-shirt and a short-sleeved T-shirt in the example). But in the similar or identical clothes retrieval task, they are usually in the same category. It also shows that the plagiarized clothes retrieval task is more challenging.

Actually, plagiarized clothes are very complicated and appear in a wide variety of forms. For example, an item which only plagiarizes the design of a certain part can be considered as plagiarism or an item which completely copies another item without any authorization, etc. Moreover, the form of plagiarized clothes is dynamic, as illegal businesses continue to update their plagiarized ways. Therefore, it is difficult to use a uniform definition to include all plagiarized types. As the first work for plagiarized clothes retrieval task, we initially define the plagia-

*Corresponding Author: Hui Xue (hui.xueh@alibaba-inc.com).

2575-7075/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPR42600.2020.00267

Figure 2. Hard cases for attribute-driven retrieval method: Indistinguishable sleeves (a & b); Unrecognizable collars (c & d).

rized clothes as samples that are modified in less than or equal to two regions on the original design (e.g., change the shape of the collar, modify the pattern within the chest region). This kind of plagiarized clothes occupies the high proportion in e-commerce platforms. Besides, the defined plagiarized clothes are relatively easy for evaluation, thus helpful for the study of the plagiarized clothes retrieval task.

In the fashion-related works [1, 2], clothes attributes are common used. However, clothes attribute is somewhat subjective, which is not very suitable for the plagiarized clothes retrieval task. For instance, it is difficult to judge the length of the sleeves or the style of the collar in some hard cases, as Figure 2 shows. Besides, for some clothes with deformations and occlusions, the retrieval performance also decreases obviously. On the contrary, the geometric properties of the clothes are highly deterministic and can maintain stability for deformed and occluded samples. Hence, we propose a novel PS-Net based on regional representation, where clothes landmarks are employed to guide the learning of regional representations and clothes are compared region by region. Besides, we find that different categories of plagiarized clothes are easy to be modified in different regions. Therefore, we would like to learn different groups of region weights for each category of clothes in order to manipulate the region weights automatically during similarity learning. By doing so, a plagiarized clothes image with a modified region could be recalled more easily. Additionally, there is no available dataset for the plagiarized clothes retrieval task. Hence, we collect a new dataset named "Plagiarized Fashion", where clothes images are annotated by experts who majored in intellectual property protection.

In summary, the major contributions of our paper are:

- We introduce a novel problem of plagiarized clothes retrieval and a new dataset named "Plagiarized Fashion" for plagiarized clothes retrieval, which provides a meaningful complement to the fashion retrieval field.
- A multi-task network named PS-Net based on the regional representation is proposed, which is superior to other instance-level counterparts for plagiarized clothes retrieval.
- Besides the plagiarized clothes retrieval, our proposed PS-Net can also be used for traditional fashion retrieval and landmark estimation tasks, achieving the state-of-the-art performance on both DeepFashion [27] and DeepFashion2 [14] datasets.

2. Related Work

Visual Fashion Analysis. Visual fashion works have attracted lots of attention due to the boom of e-commerce and online shopping in these years. With the development of large-scale fashion datasets [27, 14], deep learning-based techniques further boosted the interest in fashion-related tasks, like clothes recognition [6, 17, 19], retrieval [16, 26, 45, 2, 49], recommendation [23, 18], clothes synthesis [5, 24] and fashion landmark detection[28, 39]. Recently, some multi-task neural network, such as FashionNet [27] and Match-RCNN [14] can even perform the above tasks simultaneously. Earlier works [40, 12] on clothes recognition mostly relied on hand-crafted features, such as SIFT [29], HOG [11]. The performance of these methods was limited by their ability of feature representation. Recently, plenty of deep learning-based models have been introduced to learn more discriminative representation [49, 22], which can even handle cross-domain scenarios [16] and near-duplicate detection task [33]. Moreover, some related works have performed clothes retrieval using parsing [45, 44], or achieved the search by attribute-driven methods [12, 1, 2, 49]. However, we found in practice that, for retrieving the images of plagiarized clothes, the existing methods are not effective enough due to the characteristic of plagiarized clothes: modified less than or equal to two regions on the original design.

Different from the above works, in this paper, we focus on the new task of plagiarized clothes retrieval. To the best of our knowledge, this paper is the first work for plagiarized clothes retrieval. Besides, the task is aimed to retrieve the plagiarized clothes with regional manipulation, which to some extent has a similar idea with Deepfake detection tasks [7, 15].

Landmark Guided Attention. Landmark detection technique is widely used in many tasks nowadays, like face alignment [42] and human pose estimation [36]. To obtain much stronger feature representations of clothes, fashion landmark estimation task is proposed in recent years [28, 46, 39]. On the other hand, attention technique is also an effective way to obtain stronger feature representations. Previous works [43, 47, 38] have proved that attention mechanism is helpful due to it enables the network to focus on the critical features and filter out the irrelevant ones.

Given an image, the typical attention model learns to obtain one whole image feature vector by weighted summing with attention weights. However, in this work, we go a step further by dividing fashion items into several regions under the guidance of predicted landmarks and learning to obtain several weighted region feature vectors. With the proposed regional attention, we compare images region by region and find it is better than the typical attention for the plagiarized clothes retrieval task.

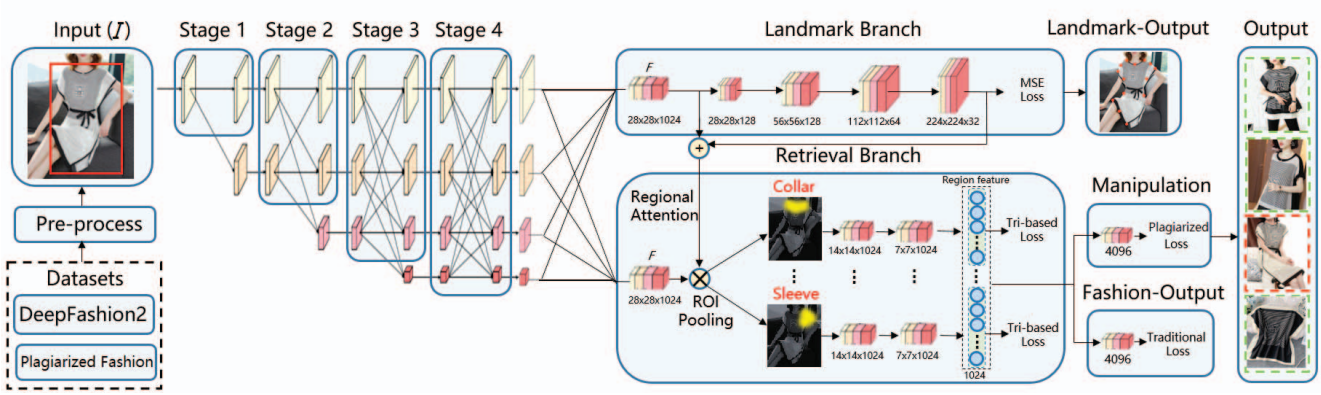


Figure 3. Structure of the proposed PS-Net which consists of a landmark branch and a retrieval branch, based on the HR-Net backbone (some convolution layers are hidden). Two output feature maps $F \in \mathcal{R}^{28\times28\times1024}$ of the backbone are identical for demonstration. The landmark guided regional attention is introduced to the retrieval branch during the ROI pooling. The retrieval branch is also split into two parts for output, one for traditional fashion retrieval and the other for plagiarized clothes retrieval. The images bounded by green and red boxes indicate plagiarized clothes and identical clothes, respectively.

3. Our Approach

Our work aims to retrieve the images of plagiarized clothes which are modified in less than or equal to two regions on the original design. Hence the key is to compute the similarity between two images of clothes. To this end, we propose a Plagiarized-Search-Net (PS-Net), which obtain the regional representation of images and compute the similarity region by region. Specially, given an image of clothes I, we propose to represent the image by multiple regional features $f_1(I), f_2(I), ...f_R(I)$, where R is the number of image regions. Besides, we find from practice that different categories of clothes are easy to be plagiarized in different regions. Therefore, we would like to learn the region weights $(\lambda_1, \lambda_2, ...\lambda_R)$ for different categories of clothes in order to manipulate the region weights automatically for plagiarized clothes retrieval. Finally, the similarity between images of clothes I and I' is:

$$\sum_{r=1}^{R} \lambda_r \cos(f_r(I), f_r(I')), \tag{1}$$

where cos indicates cosine similarity between two feature vectors. Figure 3 illustrates the structure of our proposed PS-Net, it is composed of a backbone, a landmark branch and a retrieval branch. As our PS-Net has a landmark branch, so it can also be used for fashion landmarks detection task.

In what follows, we firstly describe the detailed structure of our proposed PS-Net, followed by the description of its optimization.

3.1. Network Architecture

Network Backbone. In the proposed PS-Net, we choose the HR-Net [36] as our backbone. With its multi-stage parallel structure, the HR-Net can maintain high resolution in deep networks, which is especially important for landmark estimation task. Note that the choice of the backbone is not mandatory, which can be replaced by any backbone with a similar effect (e.g., ResNet [19], VGG-Net [35]). Besides, as shown in Figure 3, the landmark branch and the retrieval branch in PS-Net share the same type of backbone (but not identical one). Before feeding an image of clothes to the backbone, we first detect the clothes in the image. Hence we trained a Faster R-CNN [32] (Res50-FPN) model on the DeepFashion2 [14] dataset as a detector to obtain the clothes and their category labels. The cropped images are resized to 224×224 pixels as the input I.

Landmark Branch. We design a landmark branch to predict landmarks on each image of clothes. More specifically, we transform the fashion landmark estimation task to predicting k heatmaps, where each the i-th heatmap indicates the location confidence of the i-th landmark. Given the output feature map F of the backbone, we use one 1×1 convolution to convert it to $28 \times 28 \times 128$. Then, several groups of transposed convolution are utilized to produce a high-resolution landmark heatmap with the same scale as the input. Finally, we use a regressor to estimate the heatmaps where the landmark positions are chosen.

Regional Attention-based Retrieval Branch. On the other hand, the output feature map F of the backbone is fed to the retrieval branch. In our experiment, we first train the model on the Deepfashion2 [14] dataset to obtain the ability for identical clothes image retrieval. After that, we get a pretrained model for further step training on plagiarized clothes retrieval task. Utilizing the regional representation achieved by the landmark branch, we fine-tune the retrieval model by manipulating the region weights. Finally, we can

Figure 4. The visualization of the landmark guided region division. Five bounding boxes are estimated which covers the largest segmented region, respectively.

Category	Sleeves	Collar	Chest	Waist	Sum
T-shirts	12%	6%	72%	10%	15,300
Tops	13%	34%	38%	15%	15,500
Outwear	25%	21%	33%	21%	14,200
Dress	6%	17%	21%	56%	15,000

Table 1. The distribution of modified regions among different categories of plagiarized clothes in our proposed Plagiarized Fashion Dataset.

obtain one retrieval model with two types of output form, which are "Fashion Output" and our target plagiarized "Output", as indicates in Figure 3.

The attention generated by the landmark branch is introduced to the retrieval branch by the following process: Firstly, we take the concatenation of the representations output $F \in \mathcal{R}^{28\times28\times1024}$ by the backbone and the bilinear downsampled landmark information $M_{ij} \in \mathcal{R}^{28\times28\times32}$ as the input. Second, we reshape the input attention map A to $28 \times 28 \times 1024$, which has the targeted scale of the retrieval branch. Then, inspired by previous fashion analysis work [25], the attention is introduced to the retrieval branch by making $F^{'} = F \circ (1/2 + A)$, where $\circ$ stands for Hadamard product. By adding $1/2$ to the attention feature map, the range of the element becomes $(1/2, 3/2)$. The critical features are strengthened by elements greater than 1, while irrelevant features are filtered out via elements less than 1. For instance, the landmarks around critical areas like cuff and collar can guide the extraction of features, which makes these key features have more possibility to retain.

To learn regional representations for the plagiarized retrieval task, we go a step further by dividing fashion items into several regions, as shown in Figure 4, under the guidance of predicted landmarks. Five bounding boxes are estimated as regions of proposal, which covers the largest segmented area, respectively. Then, we achieve an ROI pooling based the proposed regions on the feature map of the Hadamard product. By this way, the landmark guided regional attention is introduced to the retrieval branch and the input image I is represented by multiple regional features.

Different from the previous work [50] in the field of person re-ID, which generates regions by an RPN [32] network for feature decomposition, we directly divide the regions by the distribution of landmark outputs. In this way, the regions generated by our approach are explicit rather than implicit, which is more controllable with the high accuracy of landmark estimation.

3.2. Optimization

The optimization process of our approach can be divided into two phases: a pre-trained phase and a fine-tune phase.

Pre-trained Phase. For the landmark branch, we choose the mean squared error (MSE) as our loss function. The ground-truth heatmaps are generated by applying 2D Gaussian with a standard deviation of 1 pixel centred on the location of each landmark.

For the regional attention based retrieval branch, we utilize triplet (tri) ranking loss which are commonly used in the retrieval tasks [13, 48]. Formally, the loss is defined as

$$\mathcal{L}_{tri}(I, I^+, I^-) = \sum_{r=1}^{R} \max(D_r^{I,I^+} - D_r^{I,I^-} + m, 0) \quad (2)$$

$$\mathcal{L}_{tra} = \sum_{n=1}^{N} \mathcal{L}_{tri}(I, I^+, I^-) \quad (3)$$

where I corresponds to the input image, N is the number of training examples, R is the number of the regions and m represents the margin. The loss aims to minimize $D_r^{I,I^+} = ||f_r(I) - f_r(I^+)||_2$ and maximizing $D_r^{I,I^-} = ||f_r(I) - f_r(I^-)||_2$. $f_r(I^+)$ and $f_r(I^-)$ represent the feature maps of image I^+ and I^- corresponded to region r, respectively. Note that the triplets are chosen from identical mini-batch. For each triplet: I and I^+ must share the same label while I^- is chosen randomly from others. By doing so, images of identical clothes are made to be close to each other in the feature space. After that, we also combine region representations $(f_1, f_2, ...f_5)$ into a global one. The concatenated feature is used to obtain the "Fashion Output" mentioned in Figure 3.

In general, our approach is able to learn critical features representations by leveraging landmark guided regional attention, which can increase the focus on specific regions during the training process. Also, the geometric properties of clothes are highly stable with few false predictions compared to the attribute-driven ones, which can enhance the retrieval performance for some hard samples (e.g., samples with deformations and occlusions).

Fine-tune Phase. The most challenging problem for plagiarized clothes retrieval is that plagiarists typically modify the clothes in a certain region on the original design to escape the supervision by traditional retrieval methods. We find from practice that different categories of clothes are easy to be plagiarized in different regions, as shown in Table 1. Therefore, we would like to learn the region weights for different categories of clothes in order to manipulate the region weights automatically for plagiarized clothes retrieval.

During the training, each image of clothes is divided into 5 regions (2 sleeves included) automatically, guided by the geometric distribution of landmarks. On the output features of the last convolution layer, plagiarized retrieval loss $\mathcal{L}_{pla}$ as shown below is imposed to enable region weights learning, which shares the same network framework with traditional fashion retrieval:

$$\mathcal{L}_{tri}^{'}(I, I^+, I^-) = \sum_{r=1}^{R} \max(D_r^{I,I^+} - D_r^{I,I^-} + m, 0) \cdot \lambda_r, \tag{4}$$

$$\alpha_{tri} = \frac{\text{avg}\{||f_r(I) - f_r(I^+)||_2; r = 1, 2, ...R\}}{\max\{||f_r(I) - f_r(I^+)||_2; r = 1, 2, ...R\}}, \tag{5}$$

$$\mathcal{L}_{pla} = \sum_{n=1}^{N} [\mathcal{L}_{tri}^{'}(I, I^+, I^-) \cdot \alpha_{tri}]. \tag{6}$$

The loss $\mathcal{L}_{pla}$ is only used to update the weight λ_r of each region, which is decoupled with the parameter update of traditional retrieval task. $\mathcal{L}_{tri}^{'}$ is a triplet-based loss which contains region weights λ_r. α_{tri} is the weight for loss functions $\mathcal{L}_{tri}^{'}$, which is updated during the training. Through the adjustment of α_{tri}, the loss $\mathcal{L}_{tri}^{'}$ of samples with large feature difference in a single region and small difference in other regions will be lower.

We utilize Coordinate Ascent as our optimization method. The λ_r of each region is set to 1 with a step size $\Delta\lambda$ of 0.1 at the beginning. The step size drops to 0.05 after 40 epochs, and 0.01 after 60 epochs. The weights of each region are sampled with a step size before each iteration ($\lambda_r \pm \Delta\lambda \rightarrow \lambda_r'$). After each iteration, if the loss decreased, the current weight λ_r' is accepted; otherwise, the weight turn back to λ_r. Note that the weights of the five regions (2 sleeves included) are always normalized in proportion to ensure a sum of 1. The weights of each region are updated iteratively to reduce the loss until the last epoch.

Finally, we also combine region representations into a global one to complete the plagiarized clothes search. The retrieval branch can recall more partially modified samples by manipulating the region weights of the features. Note that the region weights for four categories of clothes are trained separately.

4. Plagiarized Fashion Dataset

Fashion datasets (e.g., DeepFashion [27], Shopping 100K [3]) provide a variety of data for the training of clothes retrieval model. But there is not yet a benchmark dataset for the retrieval of plagiarized clothes. Hence, in this paper, we propose a new dataset named Plagiarized Fashion for plagiarized clothes retrieval. The dataset contains 60,000 images in total, where 40,000 images for training, and 20,000 images for testing. Among them, 1500 are query images, and the others are gallery images. The dataset consists of four categories of clothes: short-sleeved T-shirts, long-sleeved tops, outwears and dresses. The numbers of samples for them are approximately balanced. Table 1 shows the distribution of modified regions among different categories of plagiarized clothes. Since the design of shorts, trousers and skirts are not recognizable enough, we do not include these three types of clothes. We consider expanding the category of clothes in future work to enable more powerful design protection ability.

We collect the dataset by crawling from Taobao, the biggest e-commerce website in Asia. Given an original clothes image, we can obtain a set of images (top-100) of similar clothes on the website by the traditional retrieval method. Then, we invite three experts who majored in intellectual property protection to achieve the annotation. The experts need to annotate the clothes images in each set by identical, plagiarized, or irrelevant. If it is plagiarized clothes, they also need to label the modified region. The challenge in constructing the dataset is to mark out the clothes with minor variations in style from a large number of identical outfits.

5. Experiment

In order to verify the effectiveness of our proposed PS-Net for the plagiarized fashion task, we evaluate it on the Plagiarized Fashion dataset. Additionally, as mentioned that PS-Net can also be adapted to traditional fashion retrieval and landmark estimation tasks, so we also conduct experiments on both DeepFashion and DeepFashion2 datasets.

Implementations. Our proposed multi-task network requires training on two datasets: 1) learn landmark estimation and retrieval abilities on 13 categories of clothes in DeepFashion2 [14] dataset; 2) obtain "reasonable" region weights for plagiarized retrieval on four categories of clothes in Plagiarized Fashion dataset. The training is carried out in sequence and finally, combined to achieve the goal of plagiarized clothes retrieval. For the landmark branch, the initial learning rate is set as 0.001. It decreases at the 9th and 12th epochs with a factor of 0.1. The training is completed after 12 epochs. For the retrieval branch, the initial learning rate is set as 0.001 and decreases at the 61st and 71st epochs with a factor of 0.1. The training is completed after 80 epochs. Specifically, given a query, it takes approximately 0.75 seconds to retrieve images from the Plagiarized Fashion dataset. The performance is tested on a computer with 64G RAM and a GTX 1080TI GPU.

5.1. Plagiarized Clothes Retrieval

Experimental Setup. We conduct the plagiarized clothes retrieval on the Plagiarized Fashion dataset. We compare our approach with the traditional method without landmark guided regional attention, manual manipulation

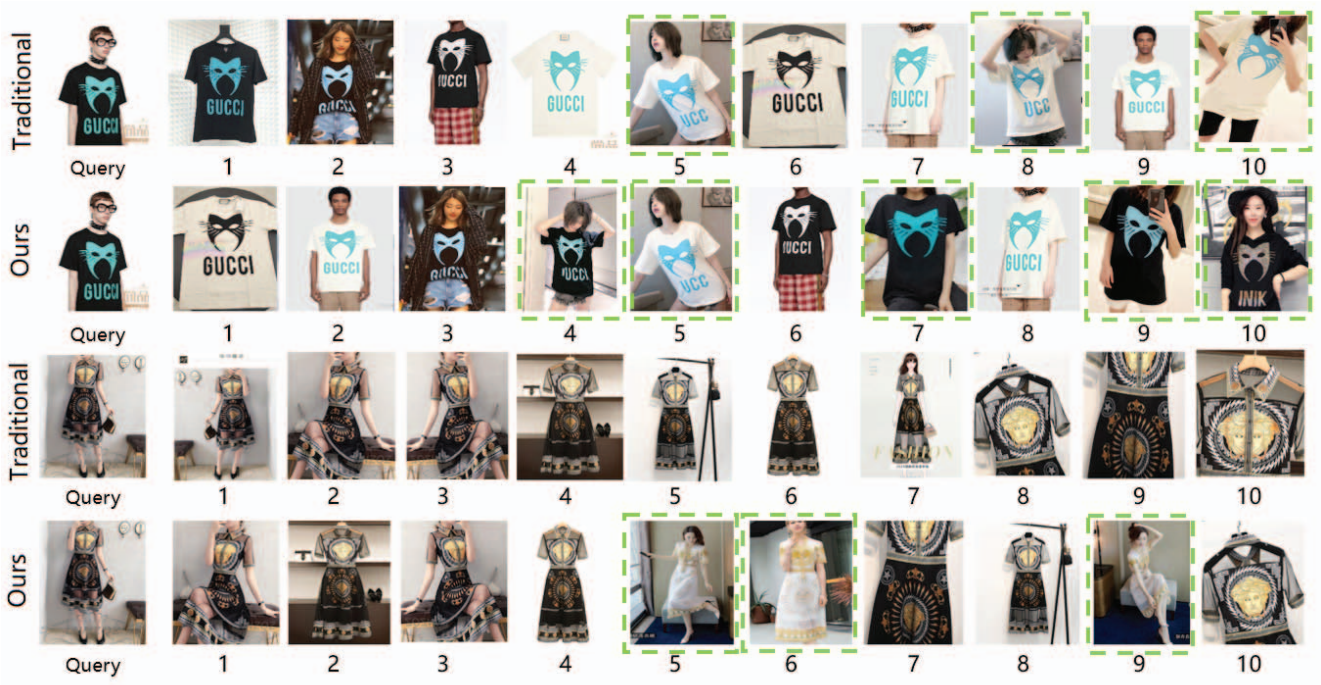


Figure 5. Example results of plagiarized clothes retrieval. The query is the clothes image with the original design, and the target recalls with green boxes are the plagiarized clothes in the gallery. For each query, the results of the traditional method are shown above, and our results are shown below.

	T-shirts			Outwears			Tops			Dress			Overall		
	Top-10	Top-20	mAP	Top-10	Top-20	mAP	Top-10	Top-20	mAP	Top-10	Top-20	mAP	Top-10	Top-20	mAP
PCB [37]	0.388	0.610	0.306	0.390	0.632	0.321	0.383	0.667	0.349	0.401	0.645	0.334	0.391	0.640	0.328
Traditional	0.395	0.622	0.313	0.398	0.640	0.325	0.390	0.672	0.353	0.406	0.650	0.338	0.397	0.645	0.332
Manual	0.564	0.803	0.465	0.532	0.772	0.401	0.556	0.793	0.451	0.532	0.768	0.429	0.542	0.783	0.443
Ours	**0.627**	**0.862**	**0.513**	**0.587**	**0.834**	**0.482**	**0.613**	**0.854**	**0.505**	**0.577**	**0.827**	**0.474**	**0.597**	**0.842**	**0.493**

Table 2. Quantitative results for plagiarized retrieval evaluated by Top-K recall and mAP. We compare our approach with the traditional method without landmark guided regional attention, manual manipulation method without learned region weights, and the PCB [37] method, which is widely used in near-duplicate retrieval task. The other settings of the model are identical.

method without learned region weights, and the PCB [37] method, which is widely used in near-duplicate retrieval task. For the traditional method, we set the five region weights as 1 by default. For the manual method, we collect the manual manipulation results from 25 participants, and use the average weight values to complete the plagiarized retrieval. Specifically, on the interactive interface we provide, the user can lower or raise the weight of each region by dragging the slider. The other settings of the model are identical (e.g., the backbone). The results of the three methods are evaluated by the metrics of Top-K recall and mAP.

Evaluation Resutls. Quantitative results for plagiarized clothes retrieval are shown in Table 2. The traditional retrieval method obtains the top-20 recall of 0.645 and an overall mAP of 0.332 on four categories of clothes, which is similar to the PCB [37] method. The manual method obtains over 10 percent improvement on recall rate and an overall mAP of 0.443, which is better than the traditional one. Then, we utilize the learned weights from the training to complete the retrieval. Our approach obtains 0.852 top-20 recall and an overall mAP of 0.493 on four categories of clothes, which improves the performance by a large margin, compared to other counterparts. Especially, for the categories of T-shirts and long-sleeves tops, our approach gets an mAP of 0.513 and 0.505, which are obviously higher than the manual method (0.465 & 0.451) and traditional method (0.313 & 0.353).

Results Visualization. Figure 5 shows two groups of plagiarized retrieval results of our approach and the traditional method. The images bounded by green boxes are correct recalls, which indicate the plagiarized clothes.

For the T-shirt and long-sleeved top categories, the tricks which commonly used by plagiarisers are replacing logo

Method	Top-10	Top-20	Top-30	mAP
No Attention	0.567	0.811	0.854	0.466
No Manipulation	0.413	0.682	0.743	0.361
None	0.397	0.645	0.698	0.332
Ours (HR-Net)	0.597	0.842	0.887	0.493
Ours (Ensemble)	**0.602**	**0.852**	**0.893**	**0.501**

Table 3. Quantitative results for the ablation study, evaluated by the Top-K recall rate and mAP.

text, adding mosaics or patterns and flipping clothes prints. Taking the first query as an example, after using the region manipulation, we successfully recall five plagiarized samples within the top-10 results. By contrast, the counterpart method only completes three plagiarized recalls within the top-10.

For the plagiarized sample of dresses, it usually not only has a small local modification but an imitation of the overall style. Therefore, the modification of this magnitude makes it difficult for traditional retrieval methods to complete the recall. For the second group of the query, the traditional method fails to recall any plagiarized clothes in the top-10 results. By manipulating the region weights, we can recall three plagiarized samples within the top-10 results.

The results show that our approach has significantly improved the ability of retrieval plagiarized clothes and alleviates the difficulty of recalling samples with partial modification. In conclusion, the region weights we learned through training are reasonable, and the region manipulation mechanism is effective for plagiarized clothes retrieval.

5.2. Ablation Study

Experimental Setup. We conduct an ablation study on the Plagiarized Fashion dataset. The factors we consider are: attention mechanism, region manipulation, and model ensemble. The results are evaluated by the Top-K recall rate and mAP.

Evaluation Results. The quantitative results of the ablation study are shown in table 3. The complete model of our approach achieved a 0.842 top-20 recall rate and a 0.493 mAP on the Plagiarized Fashion dataset. When we ensemble five models together (with different initial learning rates from 0.0005 to 0.01), the top-20 recall is increased to 0.852, and the mAP becomes 0.501. When the attention mechanism is removed from the complete model, it achieved a top-20 recall rate of 0.811 and the mAP drops to 0.466, which proves that the attention mechanism is vital for the retrieval task. To verify the effect of region manipulation, we adjust the learned region weights to the default ones. The top-20 recall rate drops significantly for more than 15 percent. Finally, we test the model without any component mentioned above, the top-20 recall rate drops about 20 percent on the Plagiarized Fashion dataset.

Method	Collar	Sleeve	Waist	Hem	Overall
FashionNet[27]	.0878	.0954	.0854	.0818	.0872
DFA[28]	.0633	.0640	.0714	.0661	.0660
DLAN[46]	.0591	.0660	.0699	.0626	.0643
BCRNN[39]	.0410	.0660	.0513	.0544	.0484
DAFE [9]	.0296	.0362	.0312	.0398	.0342
Ours	**.0293**	**.0358**	**.0310**	**.0396**	**.0339**

Table 4. Quantitative results for clothes landmark detection on the DeepFashion [27] dataset, evaluated by normalized error (NE). The best scores are marked in bold.

From the above comparison results, we can find that two essential designs of our approach: landmark guided regional attention and region manipulation are vital for plagiarized clothes retrieval. Moreover, the model ensemble is also beneficial.

5.3. Landmark Estimation

Experimental Setup. The landmark estimation experiments are conducted on both DeepFashion [27] and DeepFashion2 [14] datasets. DeepFashion is the most widely used fashion landmark dataset with 123,016 clothes images. Followed by previous work [39], we use the standard dataset split and the evaluation is performed on 40,000 images. DeepFashion2 is the most challenging fashion landmark dataset at present, which has a different number of landmarks for each type of clothes. It contains 491,895 clothes images, and the experiment is evaluated on 33,669 images. According to the previous methods[39, 14], the results on DeepFashion dataset are evaluated by normalized error (NE). The results of DeepFashion2 dataset are evaluated by Average Precision (AP).

Evaluation Results. Our approach obtains an average result of 0.0339 on DeepFashion dataset, as shown in Table 4, which is much better than previous methods like DLAN [46] (0.0643), DFA [28] (0.0660), Fashion-Net [27] (0.0872), BCRNN [39] (0.0484) and the recent method with dual attention [9] (0.0342).

The results of DeepFashion2 dataset is shown in Table 5. Note that we conduct the experiment on both visible and occluded landmarks. Since the DeepFashion2 dataset is a new dataset with none comparison method, we compare our approach with the released networks of Simple-Baseline [41] and CPN [10] (two of the best methods for human landmark estimation). Our approach obtains an overall AP of 0.633, which greatly higher than the Simple-Baseline [41] (0.591), CPN [10] (0.579) and the Match-RCNN [14] (0.563).

In general, our approach obtains SOTA results in the landmark estimation task on two fashion datasets. It indicates that the design of the landmark branch is effective.

Method	Scale			Occlusion			Zoom-in			Viewpoint			Overall
	small	moderate	large	slight	medium	heavy	no	medium	large	no wear	frontal	side	
Match-RCNN[14]	0.497	0.607	0.555	0.643	0.530	0.248	0.616	0.489	0.319	0.510	0.596	0.456	0.563
CPN*[10]	0.512	0.619	0.560	0.663	0.542	0.261	0.625	0.501	0.330	0.523	0.621	0.468	0.579
Simple-Baseline*[41]	0.523	0.632	0.574	0.671	0.562	0.277	0.638	0.512	0.349	0.543	0.632	0.485	0.591
Ours	**0.581**	**0.682**	**0.633**	**0.713**	**0.606**	**0.332**	**0.691**	**0.567**	**0.408**	**0.592**	**0.679**	**0.533**	**0.633**

Table 5. Landmark estimation results on different subsets of DeepFashion2 [14], evaluated by Average Precision (AP). The best performance are marked in bold. * represents the results we achieved by the released network [10, 41].

5.4. Traditional Retrieval

Experimental Setup. The main goal of our work is to retrieve plagiarized clothes. On the other hand, we would like to demonstrate that the landmark guided regional attention can also enhance the performance of traditional retrieval task. Thus, we evaluate our approach with the same ResNet [19] backbone of previous methods on Deepfashion [27] and DeepFashion2 [14] datasets.

According to the previous methods [33, 14], Top-K recall on (a) DeepFashion [27] and Top-K accuracies on (b) DeepFashion2 [14] datasets are plotted in Figure 6. DeepFashion provides 52,712 clothes images for the retrieval within shops, while DeepFashion2 provides a Customer-to-Shops application scenario with 491,895 images. We conduct the evaluation on 26,830 and 33,669 images, respectively.

Evaluation Results. On the DeepFashion dataset, our approach achieves 0.889 recall rate at top-1, which is obviously better than the previous retrieval methods like DARN [21] (0.381), WTBI (0.347) [8], FashionNet [27] (0.533), and the various form of FashionNet. When retrieving the top-50 results, the recall rate of our approach has reached 0.991. Shin et al. [33] proposed a semi-supervised feature-level manipulation for fashion image retrieval. It achieves the same top-50 performance (0.991) compared to us, but its top-1 performance (0.887) is lower than ours.

Figure 6 (b) shows the retrieval results on DeepFashion2. Our approach obtains a top-10 result of 0.745, which is much higher than the result of PCB method [37] (0.703), achieving based on the released model. Compared to the best performance by Match-RCNN [14] (0.573), our performance obtains a huge improvement of 18 percent. The geometric distribution of the landmarks allows the model to know the structure of the clothes in any case, which is especially important for partial occluded or deformed clothes. By coincidence, these samples are often encountered in Customer-to-Shops scenario. Match R-CNN [14] method also trains the landmark branch and the retrieval branch simultaneously, but it only accumulates the loss functions without introducing any attention.

As can be seen from the result, introducing landmark guided regional attention during the similarity learning is very effective. In conclusion, our method has achieved state-of-the-art effects on both In-Shops and Customer-to-Shops search, which verifies the retrieval ability of our approach on different application backgrounds and image quality.

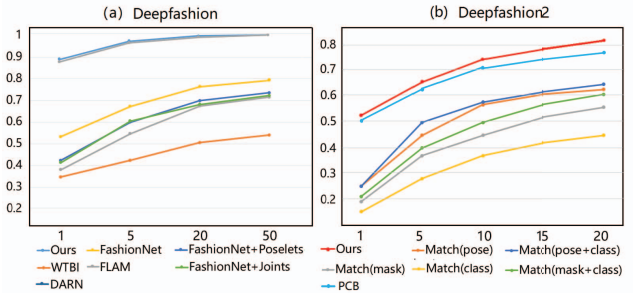


Figure 6. The results of the traditional clothes retrieval experiment. Top-K recall on (a) DeepFashion [27] and Top-K accuracies on (b) DeepFashion2 [14] datasets are plotted.

6. Conclusion

In this paper, we introduce a novel problem of plagiarized clothes retrieval for original design protection and provide a novel network named PS-Net with a dedicated Plagiarized Fashion dataset, which fills the gap in the field of fashion retrieval. We propose an attentive network based on regional representation and let the geometric information of landmarks guide the similarity learning, which outperforms the other SOTA counterparts. We design a region manipulation mechanism to solve the problem of plagiarized clothes retrieval. We learn the region weights for different categories of clothes on the proposed dataset, in order to manipulate the region weights automatically during similarity learning. By doing so, a plagiarized clothes image with a modified region can also be recalled, which has significant improvement compared to other methods. In general, our work can effectively alleviate the problem of plagiarized clothes retrieval and has great potential for original design protection.

Acknowledgement. This work was partly supported by the National Natural Science Foundation of China (No.61902347), the Zhejiang Provincial Natural Science Foundation (No.LQ19F020002) and the Alibaba-ZJU Joint Research Institute of Frontier Technologies.

References

[1] Kenan E Ak, Ashraf A Kassim, Joo Hwee Lim, and Jo Yew Tham. Fashionsearchnet: Fashion search with attribute manipulation. In *ECCV*, 2018. 2

[2] Kenan E Ak, Ashraf A Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR*, 2018. 1, 2

[3] Kenan E Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf A Kassim. Efficient multi-attribute similarity learning towards attribute-based fashion search. In *WACV*, 2018. 5

[4] Kenan E Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf A Kassim. Which shirt for my first date? towards a flexible attribute-based fashion query system. *Pattern Recognition Letters*, 2018. 1

[5] Kenan E Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf A Kassim. Attribute manipulation generative adversarial networks for fashion images. In *ICCV*, 2019. 2

[6] Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. Fashion forward: Forecasting visual style in fashion. In *ICCV*, 2017. 2

[7] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *ICCV Workshops*, 2019. 2

[8] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *ECCV*, 2012. 8

[9] Ming Chen, Yingjie Qin, Lizhe Qi, and Yunquan Sun. Improving fashion landmark detection by dual attention feature enhancement. In *ICCV Workshops*, 2019. 7

[10] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 7, 8

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[12] Wei Di, Catherine Wah, Anurag Bhardwaj, Robinson Piramuthu, and Neel Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPR Workshops*, 2013. 2

[13] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 4

[14] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *CVPR*, 2019. 2, 3, 5, 7, 8

[15] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *AVSS*, 2018. 2

[16] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, 2015. 1, 2

[17] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 2

[18] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *ACM MM*, 2017. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 8

[20] Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W Zheng, and Qi Liu. Explainable fashion recommendation: A semantic attribute region guided approach. *arXiv preprint arXiv:1905.12862*, 2019. 1

[21] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, 2015. 8

[22] Naoto Inoue, Edgar Simo-Serra, Toshihiko Yamasaki, and Hiroshi Ishikawa. Multi-label fashion image classification with minimal human supervision. In *ICCV*, 2017. 2

[23] M Hadi Kiapour, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, 2014. 2

[24] Yining Lang, Yuan He, Jianfeng Dong, Fan Yang, and Hui Xue. Design-gan: Cross-category fashion translation driven by landmark attention. In *ICASSP*, 2020. 2

[25] Jingyuan Liu and Hong Lu. Deep fashion analysis with feature map upsampling and landmark-driven attention. In *ECCV*, 2018. 4

[26] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, 2012. 1, 2

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 1, 2, 5, 7, 8

[28] Ziwei Liu, Sijie Yan, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Fashion landmark detection in the wild. In *ECCV*, 2016. 2, 7

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[30] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017. 1

[31] Zhe Ma, Jianfeng Dong, Yao Zhang, Zhongzi Long, Yuan He, Hui Xue, and Shouling Ji. Fine-grained fashion similarity learning by attribute-specific embedding network. *arXiv preprint arXiv:2002.02814*, 2020. 1

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3, 4

[33] Minchul Shin, Sanghyuk Park, and Taeksoo Kim. Semi-supervised feature-level attribute manipulation for fashion image retrieval. *arXiv preprint arXiv:1907.05007*, 2019. 2, 8

[34] Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. Neuroaesthetics in fashion: Modeling the perception of fashionability. In *CVPR*, 2015. 1

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2, 3

[37] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, 2018. 6, 8

[38] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *TIP*, 2017. 2

[39] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, 2018. 1, 2, 7

[40] Xianwang Wang and Tong Zhang. Clothes search in consumer photos via color matching and attribute learning. In *ACM MM*, 2011. 2

[41] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 7, 8

[42] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 2

[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[44] Kota Yamaguchi, M Hadi Kiapour, and Tamara L Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013. 2

[45] Kota Yamaguchi, M Hadi Kiapour, Luis E Ortiz, and Tamara L Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012. 1, 2

[46] Sijie Yan, Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Unconstrained fashion landmark detection via hierarchical recurrent transformer networks. In *ACM MM*, 2017. 2, 7

[47] Yijun Yan, Jinchang Ren, Genyun Sun, Huimin Zhao, Junwei Han, Xuelong Li, Stephen Marshall, and Jin Zhan. Unsupervised image saliency detection with gestalt-laws guided optimization and visual attention based refinement. *Pattern Recognition*, 2018. 2

[48] Xun Yang, Meng Wang, and Dacheng Tao. Person reidentification with metric learning using privileged information. *TIP*, 2017. 4

[49] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *CVPR*, 2017. 1, 2

[50] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017. 4

Structured Query-Based Image Retrieval Using Scene Graphs

Brigit Schroeder
University of California, Santa Cruz
brschroe@ucsc.edu

Subarna Tripathi
Intel Labs
subarna.tripathi@intel.com

Abstract

A structured query can capture the complexity of object interactions (e.g. 'woman rides motorcycle') unlike single objects (e.g. 'woman' or 'motorcycle'). Retrieval using structured queries therefore is much more useful than single object retrieval, but a much more challenging problem. In this paper we present a method which uses scene graph embeddings as the basis for an approach to image retrieval. We examine how visual relationships, derived from scene graphs, can be used as structured queries. The visual relationships are directed subgraphs of the scene graph with a subject and object as nodes connected by a predicate relationhship. Notably, we are able to achieve high recall even on low to medium frequency objects found in the long-tailed COCO-Stuff dataset, and find that adding a visual relationship-inspired loss boosts our recall by 10% in the best case.

1. Introduction

An image is composed of a complex arrangement of objects and their relationships to each other. As noted in [1][2], this is why content-based image retrieval is more successful when using complex structured queries (e.g. 'girl programs computer') rather than simply using single object instances (e.g. 'girl', 'computer', etc.). Instead of viewing objects in isolation, they can be coupled as a subject and object by a relationship that describes their interaction [3]. These *visual relationships*, in the form of *<subject, predicate, object>* (e.g. *<girl, programs, computer>*), can be used as complex structured queries [1] for performing image retrieval, which are more descriptive and efficient via their specificity.

Visual relation-based retrieval, although more useful, is much more difficult than single object-based retrieval as the representation learning for the former case is more difficult. For example, the current state-of-the-art (SOTA) for object detection including both small object and long-tailed distributions achieves well over 50% [4] mean average precision (mAP) and recall at k=100 is about 70%. However, when it comes to visual relationship detection (in particular 'scene graph detection task'), even only for large and frequent objects from Visual Genome, the SOTA [5] recall at k=100 is below 35%. This intuitively speaks to the difficulty of working with visual relationship representations, and thus the retrieval task at hand.

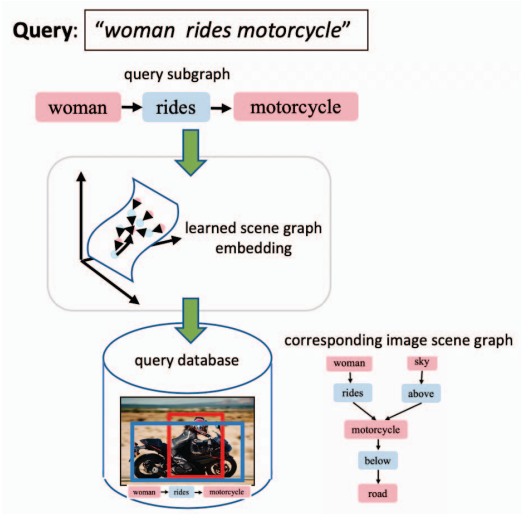


Figure 1. **Visual Relationship Subgraph Query.** We use visual relationships, represented as directed subgraphs extracted from scene graphs, to form structured queries. Each subgraph contains a subject and object as nodes connected by an edge representing a predicate relationship.

In this paper, we approach the image retrieval problem by using a learned scene graph embedding from a scene layout prediction model (Fig 2). Scene graphs are a structured data format which encodes semantic relationships between objects [6]. Objects are represented as nodes in the graph and are connected by edges that express relationship, in the form of triplets. As shown in Figure 1, we use visual relationships, represented as directed subgraphs extracted from scene graphs, to form structured queries. Each subgraph contains a subject and object as nodes connected by an edge representing a predicate relationship.

2160-7516/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPRW50498.2020.00097

Our work is unique in that we perform retrieval using only the embeddings extracted from scene graphs rather than visual features, which is a common modality in image retrieval [2]. We perform a quantitative and qualitative analysis which demonstrates our method's efficacy when dealing with a long-tailed dataset with overwhelming majority of low-frequency classes. We also observe that learning objectives derived directly from the visual relationships boost the image retrieval efficiency significantly.

2. Related Work

Early work by Johnson *et al*. [2] uses visual relationships derived from scene graphs for image retrieval. A scene graph is a structured data format which encodes semantic relationships between objects. A set of visual relationships containing a <*subject, predicate, object*> are the building blocks of these scene graphs. Johnson *et al*. [2] use a conditional random field (CRF) model trained over the distribution of object groundings (bounding boxes) contained in the annotated scene graphs. Object classification models are used as part of the CRF formulation. Wang *et al*. [7] use cross-modal scene graphs for image-text retrieval where they rely upon using word embeddings and image features. We distinguish our approach from [7] [2] as we do not use object groundings, word embeddings nor the input image features to perform our retrieval.

A line of work that emerged recently [8, 9, 10, 11, 12, 13, 14] takes scene graphs as input and produce final RGB images. All of these methods perform an intermediate layout prediction by learning embeddings of nodes. We use layout generation as a pretext task for learning the embedding to perform image retrieval as the downstream application. However, unlike the above layout generation models, our method utilizes <*subject, predicate, object*> triplets as additional supervisory signal for more effective structured prediction [15]. In a closely related work, Belilovsky *et al*. [16] learn a joint visual-scene graph embedding for use in image retrieval. In contrast, our model is trained from scene graphs without access to visual features.

3. Method

3.1. Dataset

In this work, we use the 2017 COCO-Stuff [17] dataset to generate synthetic scene graphs with clean predicate annotations. COCO-Stuff augments the COCO dataset [18] with additional stuff categories. The dataset annotates 40K train and 5K val images with bounding boxes and segmentation masks for 80 thing categories (people, cars, etc.) and 91 stuff categories (sky, grass, etc.). Similar to [8], we used thing and stuff annotations to construct synthetic scene graphs based on the 2D image coordinates of the objects. Six mutually exclusive geometric relationships are encoded and used as the predicate in visual relationships: *left of, right of, above, below, inside, surrounding*.

3.2. Learning a Scene Graph Embedding

We use a layout prediction network [15] inspired by the image generation pipeline in [8] to learn a scene graph embedding for image retrieval. Figure 2 gives an overview of the network architecture. A graph convolutional neural network (GCNN) processes an input scene graph to produce embeddings corresponding to object nodes in the graph. The GCNN is a 5-layer multilayer perceptron where D_{input} = D_{output} = 128 and D_{hidden} = 512. Singleton object embeddings are passed to the next stage of the layout prediction network per [8]. The outputs of the second stage of the layout prediction model are used to compose a scene layout mask with object localization. We utilize the object embeddings to form a set of triplet embeddings where each is composed of a <*subject, predicate, object*>. We pass these through a triplet mask prediction network which learns to label objects as either 'subject' or 'object' (see Figure 3), enforcing both an ordering and relationship between objects. We also pass triplet embeddings through a triplet 'superbox' regression network, where we train the network for joint localization over subject and object bounding boxes. A superbox is defined as the enclosing bounding box of both the subject and object bounding boxes as noted in Figure 2.

In this work, the layout prediction can be thought of as a pretext task in which the scene graph embedding is learned as an intermediary feature representation. To improve the learning in this task, we apply two triplet-based losses [15] in addition to those used in [8]. The first is a triplet mask loss, $L_{triplet-mask}$, penalizing differences between ground truth triplet masks and predicted triplet masks with pixelwise cross-entropy loss. The second is a triplet superbox loss, $L_{triplet-superbox}$, penalizing the L_2 difference between the ground truth and predicted triplet superboxes. We also train a layout prediction model without triplet-based losses for comparison.

4. Experimental Analysis

Query Database. In this approach we focus on utilizing the object embeddings from a learned scene graph embedding to form structured queries. We have experimented with multiple forms of queries, including but not limited to visual relationships. For our testing, we query a database of 3100 visual relationships extracted from annotated test scene graphs from the COCO-Stuff dataset. We do not limit our database by requiring visual relationships to have a minimum number of occurrences as was done in [2]. We use the similarity metric S to rank our retrieved images corresponding to their respective embedding space representation

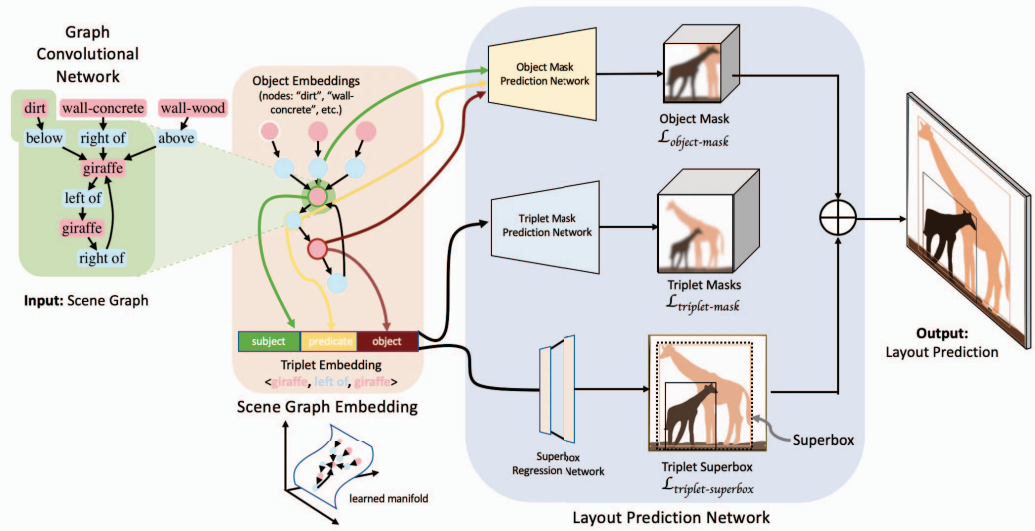


Figure 2. **Scene Graph Embeddings from Layout Prediction.** A scene graph embedding is learned via a *pretext task* which is training a layout prediction network. Later, image retrieval from structured queries, the downstream application we aim to address, uses the similarity metric in the learned scene graph embedding space.

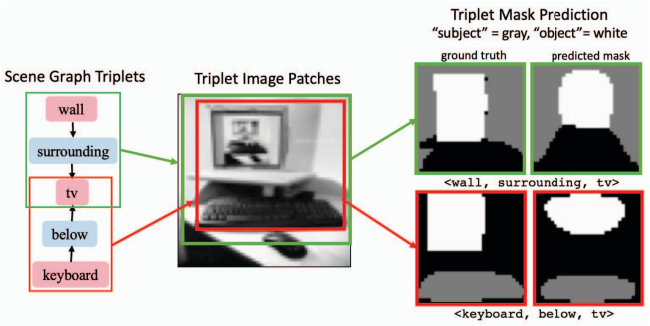


Figure 3. **Triplet Mask Prediction.** Triplets containing a *<subject,predicate,object>* found in a scene graph are used to predict corresponding triplet masks, labelling pixels either as subject and object. The mask prediction is used as supervisory signal during training.

Model	R@1	R@25	R@50	R@100
Triplet-s+o	**0.14**	**0.33**	**0.40**	**0.46**
Triplet-s+p+o	0.10	0.29	0.35	0.42
NoTriplet-s+o	0.10	0.24	0.31	0.36
NoTriplet-s+p+o	0.11	0.26	0.31	0.38
Baseline-s	0.07	0.19	0.23	0.29
Baseline-o	0.07	0.19	0.23	0.29
Baseline-p	0.00	0.01	0.02	0.03
Random	0.00	0.00	0.01	0.01
Head Classes	0.10	0.33	0.41	0.50
Long-tail Classes	**0.19**	**0.41**	**0.46**	**0.51**

Table 1. **Recall@k.** Image retrieval performance is measured in terms of Recall@{1,25,50,100} for all classes (upper portion) and then separately for head and long-tail classes (bottom portion)

.

$$S = \frac{1}{d(q, r_k)} \tag{1}$$

where d is the L_2 distance between the query q and retrieved result r_k at position k.

Long-tail Distributions. We do not restrict the vocabulary of object classes which are used to form our queries as done in [2]. Given the long-tailed nature of the COCO-Stuff dataset (see Figure 6), it is important to acknowledge the difficulty a model may have in learning all classes sufficiently [19], especially for low frequency classes. We wish to understand how well our scene graph embedding performs given these challenges. Therefore, we divide objects classes into two parts for our experimental analysis. The first is head classes which comprise the first 20% of dataset) (e.g.*person*, *tree* and *sky*). The second is long-tail classes which comprise the remaining 80% of dataset (e.g. *zebra* , *skateboard* and *laptop*).

Results. Initially, we break down our queries individually by subject (s), object (o) or predicate (p) as baselines. We see that the contribution from the subject and object embedding is much more significant over the predicate, as seen in Figure 5 on the left. Image retrieval using only the predicate embedding is poor in terms of Recall@100 of 3%, no better than random which has Recall@100 at 1%. However, the subject and object embeddings both have a Recall@100

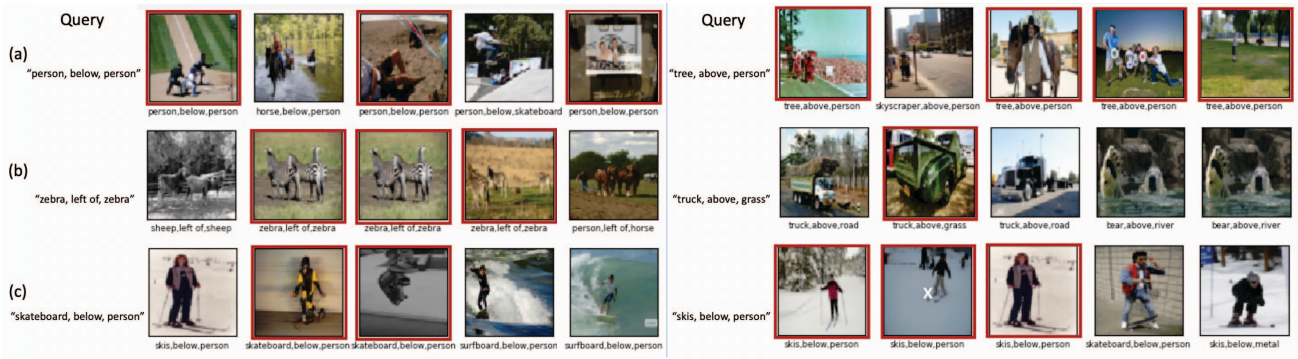


Figure 4. **Image Retrieval Results.** Retrieval for structured queries with object types with varying levels of frequency in COCO-Stuff dataset: (a) head (*person, tree*), (b) (long-tail) medium frequency (*zebra, truck*), and (c) (long-tail) low frequency (*skateboard, skis*). Query is in left-most column corresponding to red boxes.

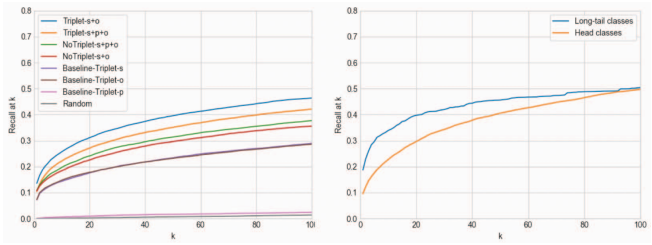


Figure 5. **Image Retrieval Performance.** Recall@k for all classes (left) and long-tail vs. head classes (right) found in COCO-Stuff.

of 29%, an increase of 26% over the predicate. The reason for this could be that the triplet supervision biases the scene graph embedding towards subject and object, pointing to the need for more supervisory signals for the predicate embedding.

Given the minimal contribution of the predicate, we examine structuring our queries with and without using the predicate (p) from the visual relationship. We compare them using two types of models, those trained with and without triplet supervision ('Triplet' and 'NoTriplet' in Figure 5). We see in Table 1 that the model trained with triplet supervision using a query structured with only subject and object ('Triplet-s+o') outperforms all model and query types by 10% in the best case (36% ('NoTriplet-s+o') vs. 46% ('Triplet-s+o') for Recall@100).

The omission of the predicate in the non-triplet modes is nominally worse ('NoTriplet-s+o' vs. 'NoTriplet-s+p+o' in Figure 5). However, we clearly see that the omission of the predicate in the triplet models ('Triplet-s+p+o' vs. 'Triplet-s+o') improves recall by 4% (42% vs. 46% for Recall@100 for). This follows the trend seen in the baseline of subject and object-only queries outperforming the predicate-based queries. We also observe that visual relationship-based queries (and structured variations thereof) outperform single object queries by 17% in the best case (29% (subject or object) alone vs. 46% ('Triplet-s+o') for Recall@100). The triplet-based losses emphasize the interaction between subject, object and predicate embeddings, and this may lend to the significant boost seen in retrieval done with structured queries.

Figure 5 (right) demonstrates the average retrieval performance on long-tail and head classes in the COCO-Stuff dataset. A low occurrence of an object class corresponds to a low occurrence of visual relationships with this object, making the task of image retrieval for long-tail classes more challenging. The long-tailed distribution of COCO-Stuff can be seen in Figure 6, where the majority of object classes in the long-tail have a frequency (count) of less than 25 instances. Despite this, the long-tail classes have a high recall@k, especially when k is less than or equal to 10. Figure 5 (right) and Table 1 show that long-tail classes tend to do at least as well as the high-frequency head classes or some cases, much better, especially at low values of k. This is exemplified in Figure 4 where even the middle to low frequency long-tail classes have several matches in the top k=5.

Qualitative retrieval results can be seen in Figure 3 (test image corresponding to query is shown for reference) using a triplet-based model. Even with middle to low frequency classes found in the long-tail distribution of COCO-Stuff, we have successful retrieval for k=5. Importantly, note that we are able to have exact matches *despite* omitting the predicate (e.g. s+o), and also exact matches in predicate (only) for all retrieval results despite its omission. Even incorrect results have the correct predicate and often are semantically similar (e.g.'surfboard below person' vs. 'skateboard below person').

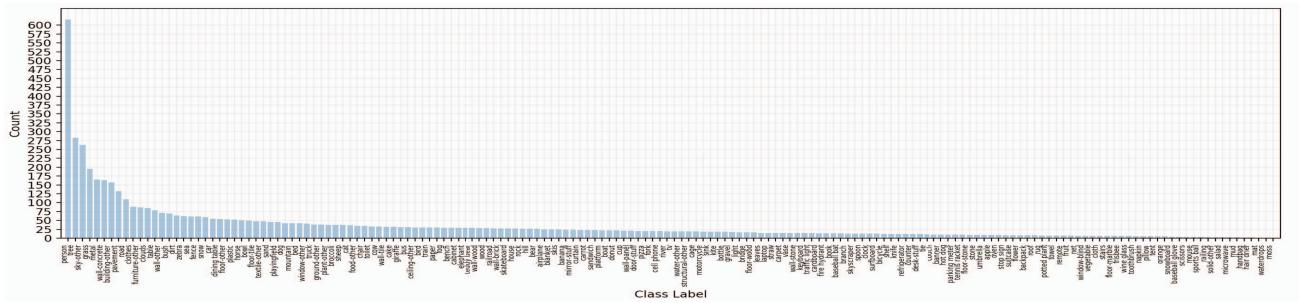


Figure 6. **Long-Tail Class Distribution.** COCO-Stuff dataset has a long-tail object class distribution. This can be partitioned into head classes (first 20%) and long-tailed classes (last 80%).

5. Conclusion

We have trained scene graph embeddings for layout prediction with triplet-based loss functions. For the downstream application of image retrieval, we use structured queries formed using the learned embeddings instead of input image content. Our approach achieves high recall even on long-tail object classes.

References

[1] Tian Lan, Weilong Yang, Yang Wang, and Greg Mori. Image retrieval with structured object queries using latent ranking SVM. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, volume 7577 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 2012.

[2] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] Xu Chen Ya Zhang Xiao Gu Yue Hu, Siheng Chen. Neural message passing for visual relationship detection. In *ICML Workshop on Learning and Reasoning with Graph-Structured Representations*, Long Beach, CA, June 2019.

[4] COCO detection leaderboard. `http://cocodataset.org/#detection-leaderboard`. Accessed: 2020-03-15.

[5] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[8] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. *CVPR*, 2018.

[9] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. Heuristics for image generation from scene graphs. *ICLR LLD workshop*, 2019.

[10] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *CoRR*, abs/1905.03743, 2019.

[11] Subarna Tripathi, Sharath Nittur Sridhar, Sairam Sundaresan, and Hanlin Tang. Compact scene graphs for layout composition and patch retrieval. *CVPRW*, 2019.

[12] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, L. Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. *ICCV*, 2019.

[13] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. `https://www.youtube.com/watch?v=V2v0qEPsjr0tm`, 2019. [ICCV 2019, Accessed: 2019-08-14].

[14] Duc Minh Vo and Akihiro Sugimoto. Visual-relation conscious image generation from structured-text, 2019.

[15] Brigit Schroeder, Subarna Tripathi, and Hanlin Tang. Triplet-aware scene graph embeddings. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[16] Eugene Belilovsky, Matthew Blaschko, Jamie Ryan Kiros, Raquel Urtasun, and Richard Zemel. Joint Embeddings of Scene Graphs and Images. International Conference On Learning Representations - Workshop, 2017. Poster.

[17] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[19] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. pages 5226–5234, 07 2017.

Signal Processing: Image Communication 80 (2020) 115652

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Robust image retrieval by cascading a deep quality assessment network ☆

Biju Venkadath Somasundaran [a], Rajiv Soundararajan [a,*], Soma Biswas [b]

[a] Department of ECE, Indian Institute of Science, Bangalore, 560012, India
[b] Department of EE, Indian Institute of Science, Bangalore, 560012, India

ARTICLE INFO

Keywords:
Image enhancement
Image quality assessment
Deep convolutional neural network
Denoising
Super resolution
Image retrieval

ABSTRACT

The performance of computer vision algorithms can severely degrade in the presence of a variety of distortions. While image enhancement algorithms have evolved to optimize image quality as measured according to human visual perception, their relevance in maximizing the success of computer vision algorithms operating on the enhanced image has been much less investigated. We consider the problem of image enhancement to combat Gaussian noise and low resolution with respect to the specific application of image retrieval from a dataset. We define the notion of image quality as determined by the success of image retrieval and design a deep convolutional neural network (CNN) to predict this quality. This network is then cascaded with a deep CNN designed for image denoising or super resolution, allowing for optimization of the enhancement CNN to maximize retrieval performance. This framework allows us to couple enhancement to the retrieval problem. We also consider the problem of adapting image features for robust retrieval performance in the presence of distortions. We show through experiments on distorted images of the Oxford and Paris buildings datasets that our algorithms yield improved mean average precision when compared to using enhancement methods that are oblivious to the task of image retrieval. [1]

1. Introduction

The proliferation of smart mobile devices has led to an explosion in the amount of images that are captured, stored and analyzed. On the other hand, the availability of increased compute power and internet connectivity has enabled the application of sophisticated computer vision algorithms for visual analytics. Indeed, the fruits of such advances have resulted in applications such as Google Lens which can improve the quality of lives of humans by providing a wealth of information. However, the performance of computer vision algorithms on camera captured images can degrade due to a variety of distortions such as noise, resolution, compression and illumination. In order to provide a reliable extraction of visual analytics, there is a need to ensure robustness of the computer vision algorithms in the presence of such distortions. In this paper, we focus on a specific instance of this robustness question by considering the problem of image retrieval. We consider the design of image enhancement algorithms to ensure the robust performance of retrieval algorithms in the presence of distortions due to noise and low resolution. We note that the image retrieval algorithm we refer to here is the classical retrieval problem where the goal is to retrieve images from a database with similar content or semantic similarity.

Image retrieval based on the bag of words model has been studied quite extensively [1,2]. Several improvements have also been proposed to overcome the limitations of feature detectors and descriptors, descriptor comparison metrics and quantization of descriptors [3–5]. Nevertheless, the performance of image retrieval in the presence of distortions and how to improve performance in such scenarios has been much less studied. Image denoising and super resolution are problems with rich literature and successful algorithms have been developed. Various techniques developed over the years have evolved to optimize the perceptual quality of the enhanced images. Improved statistical priors on natural images and the idea of exploiting the similarity of patch content across the image have led to image denoising algorithms with excellent performance [6,7]. The theory of sparse signal representations has been used to develop state of the art single image super resolution algorithms [8]. Recently, deep convolutional neural networks (CNN) have been successfully deployed for both image denoising and super resolution [9]. It is shown that state of the art performance can be achieved for both these problems using simple architectures of CNNs. While all these algorithms lead to images with very good perceptual

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.115652.

* Corresponding author.
E-mail addresses: bijuvselankur@gmail.com (B.V. Somasundaran), rajivs@iisc.ac.in (R. Soundararajan), somabiswas@iisc.ac.in (S. Biswas).

[1] The material in this paper appeared in part at the 2018 IEEE International Conference on Image Processing, Athens, Greece.

https://doi.org/10.1016/j.image.2019.115652
Received 7 February 2019; Received in revised form 28 September 2019; Accepted 28 September 2019
Available online 5 October 2019

quality, their relevance to the success to computer vision algorithms and in particular, image retrieval, has been much less studied.

At first sight, the optimization of classical denoising and/or single image super resolution algorithms for image retrieval tasks appears to be challenging. This is partly because the denoising or super resolution algorithms themselves are complex involving non-linear operations of various parameters that need to be optimized. While the use of deep CNNs simplifies the enhancement operation to some extent, networks are typically optimized for cost functions such as regularized mean squared error or perceptual quality indices such as the structural similarity index [10]. While these cost functions may be relevant for perceptual quality, their relevance in improving the performance of image retrieval is not clear. The measurement of image retrieval performance involves two components, the retrieval algorithm itself and the performance evaluation of the output of the retrieval algorithm in terms of metrics such as average precision by comparing the output with an annotated database. These involve a complex sequence of operations that cannot be written as a closed form expression. Thus it is not clear how a differentiable cost function can be obtained that can be used to optimize the image enhancement algorithms.

Our main contribution is in the design of a framework for image denoising and super resolution for image retrieval. We first design a deep CNN to predict the image retrieval performance in terms of average precision as a function of image distortions. We refer to this CNN as the quality assessment for image retrieval (QAIR) CNN since it predicts the image quality as relevant to image retrieval. We then cascade this CNN to the output of a denoising or super resolution CNN and use the output of the QAIR CNN to optimize the weights of the denoising or super resolution CNN through back propagation. This architecture provides a seamless method to optimize the denoising or super resolution for improving image retrieval performance. We conduct experiments to show that the QAIR CNN is efficient in predicting the image quality of the distorted image. Further we also show that by coupling the enhancement CNN with the QAIR CNN, we are able to improve the performance of image retrieval when compared to approaches which treat enhancement and retrieval as separate problems.

In contrast to the approach of image enhancement to achieve robust image retrieval, we then consider the complementary problem of feature adaptation for image retrieval. The goal of this problem is to design a framework that allows the learning of features for the image retrieval task at hand in the presence of distortions. Further, while we seek to learn features, the rest of the pipeline in the given retrieval algorithm remains unchanged. The features will need to be learnt appropriately for a different retrieval task. While a generic solution appears to be challenging, we present a solution for adapting deep CNN based features used for image retrieval [11]. In particular, we design a QAIR CNN which takes as input, the deep CNN based features in [12] and predicts the average precision of the image retrieval. We then cascade this QAIR CNN with the deep CNN used to generate the features to fine tune the later CNN while keeping the former fixed. We show that this feature adaptation leads to an improvement in performance of the deep CNN based features with respect to noise and low resolution.

We published preliminary results of our work in a conference version which only focussed on the problem of image denoising for image retrieval for specific noise levels [13]. In this paper, we also consider the complementary problem of feature adaptation for robust image retrieval on distorted images and show how our framework can be used to solve this problem as well. This material is contained in Section 5 and is completely new. We also extend our image denoising framework for a set of noise levels instead of individual noise levels. Further, we apply our framework to perform image super resolution for image retrieval. The extension to super resolution is discussed in Section 4.2. The experimental results corresponding to all the new material are contained in Sections 6.3, 6.4.3, 6.6, 6.7, and 6.9.

The rest of the paper is organized as follows. In Section 2, we present an overview of the related work. We describe our method of quality assessment for image retrieval in Section 3, the image enhancement framework in Section 4 and the feature enhancement approach in Section 5. We present detailed experiments and comparisons in Section 6 and conclude the paper in Section 7.

2. Related work

We now discuss prior work related to our problem. We identify five different areas in image retrieval, image denoising, super resolution, quality assessment and the connection between computer vision algorithms and image quality as related to our work. We discuss these in the following.

Image retrieval usually refers to the problem of retrieving a set of images relevant to a query image containing a particular object. Successful retrieval algorithms based on the construction of a bag of visual words have been developed [1,2]. Several researchers have improved the performance of retrieval algorithms by designing different feature descriptors [4]. Further, spatial and geometric constraints [14,15] have also led to improved performance. Compact codes have been designed based on local image descriptions to speed up the retrieval algorithms [16]. While majority of the approaches deal with improvements in the image retrieval pipeline, the robustness of the retrieval algorithm to image quality degradations such as noise and resolution has been much less studied.

There is rich literature in image denoising. One of the state of the art denoising methods is Block-Matching and 3D Filtering (BM3D) [6], which is based on non-local self similarity and combines multiple steps such as block matching, collaborative filtering on different blocks and aggregation of different blocks to form the denoised image. Other successful image denoising algorithms such as those based on expected patch log likelihood (EPLL) [7] and Gaussian scale mixture models [17], explore the availability of rich natural scene statistical models. Sparse representations of images have also led to successful image denoising algorithms [18]. While neural networks were initially explored for image denoising [19], deep convolutional neural networks (CNNs) such as DnCNN [9] and FFDNet [20] have been shown to achieve state of the art image denoising performance.

The problem of image super resolution has also been addressed by several researchers. Improving resolution by image registration [21] and example based super resolution [22] are examples of super resolution using multiple low resolution images. One of the earlier pieces of work on single image super resolution was done by Glasner et al. [23] by exploiting the recurrence of patches in an image, both at the same scale as well as across scales. Dong et al. designed a CNN called SRCNN which had 3 layers and achieves super resolution on image patches [24]. Kim et al. came up with a deep CNN based model for image super resolution [25] inspired by the VGG-net for image classification by predicting the residual image given an up sampled low resolution image. This residual image is then added to the up sampled image to generate the high resolution image. DnCNN [9] also adopts a similar approach to solve the super resolution problem.

The problem of perceptual image quality assessment has rich literature and significant progress has been made on no reference image quality assessment through algorithms such as DIVIINE [26], BLIINDS [27], BRISQUE [28] and CORNIA [29]. While the above algorithms operate based on natural scene statistics based features, there have been several efforts based on convolutional neural networks [30–33]. In [31], a pre-trained deep CNN to extract image features is combined with dense fully connected layers to predict perceptual image quality. CNN based architectures have been also been applied successfully in both full reference and no reference QA through a unified framework [34].

The impact of image quality on computer vision tasks has been much less studied. Perceptual image quality features are shown to be relevant for robust face detection [35]. The notion of machine vision quality is used to design image enhancement algorithms for

face detection [36]. The relation between image quality and image utility or the usefulness of an image with respect to performing a particular task is explored in [37]. The relation between image quality and the performance of object tracking has also been studied [38]. More recently, image denoising algorithms have been optimized for a deep learning based image classification problem [39].

3. Image quality assessment for image retrieval

In this section, we define the notion of image quality with respect to the success of the specific computer vision task of image retrieval. We first describe the performance measurement of image retrieval and then define our notion of quality for image retrieval. An image retrieval algorithm takes as input, an image database and a query image and returns as output, matching images from the database in order of their similarity to the query image. An example of a retrieval algorithm based on the scale invariant feature transform (SIFT) is shown in Fig. 1. Image retrieval involves the computing of image features and their comparison with a database of images subject to some geometric consistency checks. Thus, the output of the retrieval algorithm is a complex function of the input image. While we present an example based on the SIFT features above, our framework applies to any image retrieval algorithm in general.

3.1. Image quality index

We define the quality of an image for image retrieval in terms of the success of the retrieval task. In particular, we define quality as the average precision achieved on a given test image with respect to the database [1]. Mathematically, let precision and recall of retrieval be defined as

$$Precision = \frac{CM}{RI}, Recall = \frac{CM}{TM}, \tag{1}$$

where RI is the number of retrieved images for a query image, CM is the number of correct matches in the set of retrieved images and TM is the total number of true matches in the database for that query image. The number of images in the sorted list output by the retrieval algorithm can be varied using a threshold to obtain a precision–recall curve. We define image quality as the average precision or the area under the precision–recall curve. Note that the average precision that we seek to predict is a function of the given retrieval algorithm.

Before we present algorithms for predicting image quality for image retrieval, we discuss how this notion of image quality can be different from perceptual image quality, which is typically associated with a task free viewing condition and human perception. The example in Fig. 2 shows the difference between quality assessment for image retrieval and perceptual quality assessment. An image which looks visually good may not give good results when used for image retrieval. On the other hand, an image which has visible distortions may yet be good from the point of view of the success of image retrieval. Thus the relation between the presence of distortions in an image and the success of a computer vision task is complex and needs to be learnt carefully.

3.2. Image QA CNN

Having defined the notion of image quality with respect to image retrieval, we now consider the problem of designing algorithms to predict this quality given a potentially distorted image. We design a CNN to predict this quality directly from the image. The use of a CNN instead of specific features such as those in [26,28] for image retrieval QA is motivated by their suitability for optimizing image enhancement as discussed in Section 4. Since we do not have enough data to train a CNN for this purpose from scratch, we use pre-trained convolutional layers of the VGG-16 CNN [12] trained for image classification on the ImageNet dataset [40], and augment it with 5 fully connected layers at the end. This is similar in nature to the approach in [31] to predict perceptual image quality. The pre-trained CNN is shown in Fig. 3 and the fully connected layers are shown in Fig. 4. The first fully connected layer has 128 nodes and the last layer has a single node corresponding to the output. All the layers except the last layer have rectified linear units (ReLU) as activation functions. Initially, the convolutional layers are frozen and the only the fully connected layers are trained using Adam optimizer. After sufficient training, the last 9 convolutional layers of the VGG-16 network are unfrozen and fine tuned using stochastic gradient descent (SGD) optimizer with a low learning rate of 10^{-3}. We refer to our CNN architecture as the quality assessment for image retrieval (QAIR) CNN.

We divide the image into patches of size 124×124 and train the QAIR CNN on image patches to predict the average precision of the distorted image from which these patches are drawn. Let x_n and y_n be the ground truth and predicted quality scores (or average precision) of the nth image patch and let N be the total number of patches. Then for training the QA CNN, we use the mean absolute error loss function defined as,

$$L = \frac{1}{N}\sum_{n=1}^{N} |y_n - x_n|. \tag{2}$$

4. Image enhancement framework for image retrieval

We now describe our approach to image enhancement for image retrieval. We consider two different image enhancement scenarios for image retrieval, image denoising and image super resolution. Our goal is to optimize image denoising or image super resolution to maximize the success of image retrieval by using the quality index we define in Section 3. Since optimizing arbitrary denoising or super resolution methods for such an index appears difficult, we present a framework where both denoising and super resolution are achieved through CNNs. We believe that this is a reasonable approach since deep CNN based methods have also been shown to achieve state of the art enhancement results. Further, the use of a CNN to define the quality with respect to image retrieval allows for a differentiable cost function. Thus gradients can be computed during back propagation to update the denoising or super resolution CNN. Note that while the true average precision can be computed for every distorted image, it is not clear how to write a differentiable cost function that can be used to update the enhancement CNN. Computing the true average precision involves finding matching scores for every image in the database with respect to the query image and listing out images from the database based on a threshold on the matching scores. Further, the threshold needs to be varied to obtain the average precision. Our CNN based approach allows to predict the average precision using a differentiable cost function. We first present the details of image denoising in detail. Super resolution follows similarly.

4.1. Image denoising for image retrieval

The proposed architecture is given in Fig. 5. As shown in the figure, there are two CNNs, one for image denoising and another for QAIR. Initially, both these CNNs are trained independently (details mentioned in Section 6.2). Then, during the combined training stage, the QAIR CNN weights are kept frozen. Only the denoising CNN is fine tuned, by minimizing a combined loss function based on the output of the QAIR CNN and the mean squared error (MSE) of the denoised image with respect to the reference image. In particular, the combined loss function is defined as

$$L = (1 - AP) + \lambda * MSE, \tag{3}$$

where AP is the average precision predicted by the QAIR CNN and MSE is the mean square error between the denoised image and the reference image. This combined loss function ensures that the quality

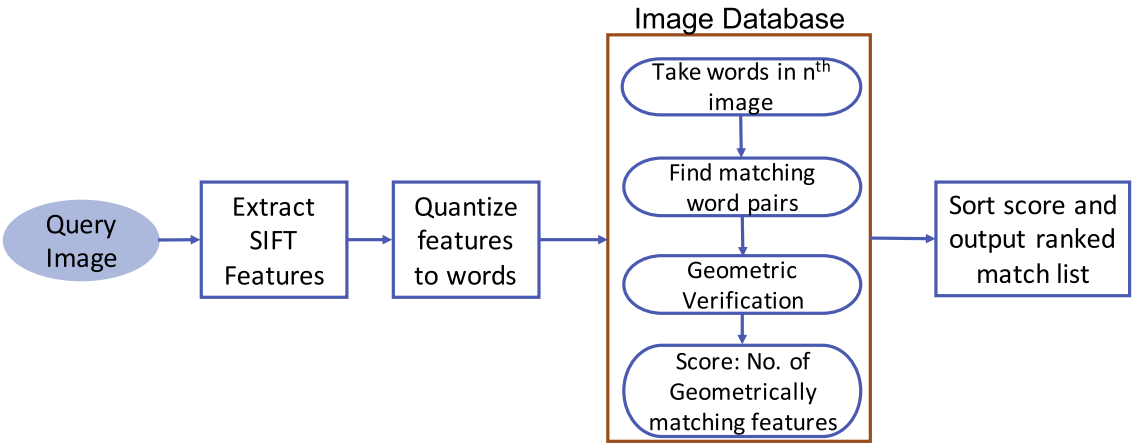


Fig. 1. A block diagram of the steps in an image retrieval algorithm.

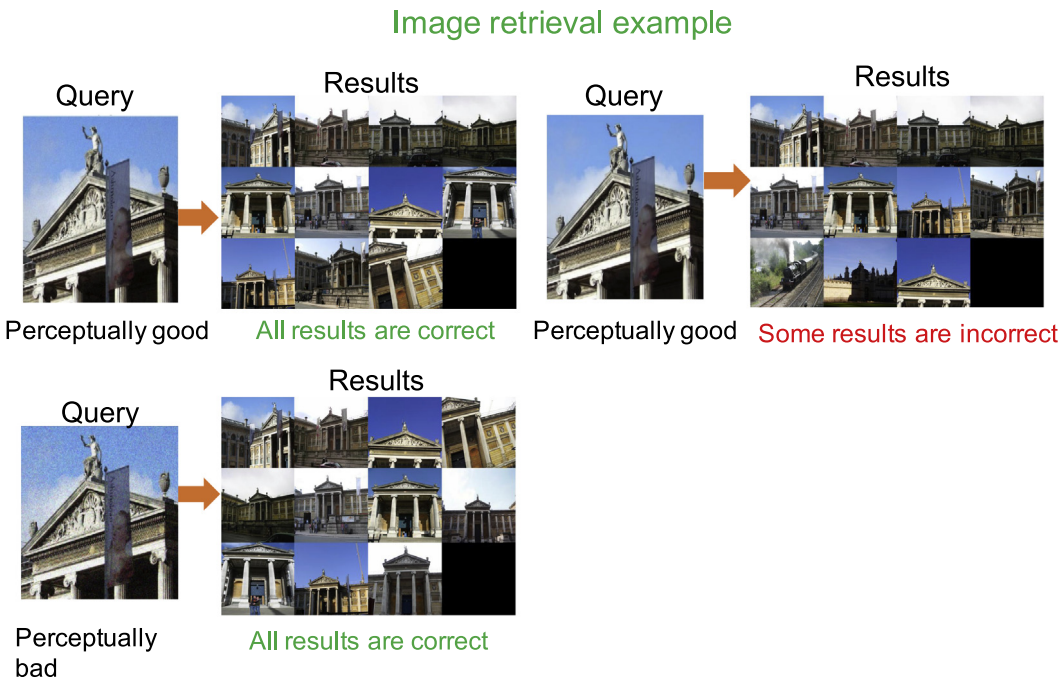


Fig. 2. Difference between perceptual quality and quality assessment for image retrieval.

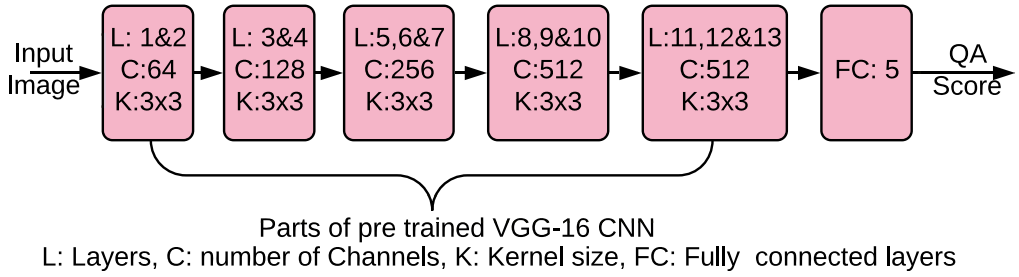


Fig. 3. Block diagram of QAIR CNN.

as predicted by the QA CNN improves without changing the denoised image too much from the actual image. λ is a parameter used to balance the two losses and the optimal value is learnt through a validation dataset. Note that in the combined loss function, AP is a function of weights of both the denoising CNN and the QA CNN, whereas MSE is a function only of the former. Once the combined training is over, the denoising CNN alone can be used for denoising and testing.

While several CNN architectures have been proposed in literature for denoising [9,20], we use a deep CNN based on the work by Zhang et al. [9], which predicts the residual noise in a noisy image. This residual noise image when subtracted from the noisy image gives the clean image. A block diagram of the network is given in Fig. 6. This CNN has 20 layers and each layer has 64 channels. All convolution kernels are of size 3×3.

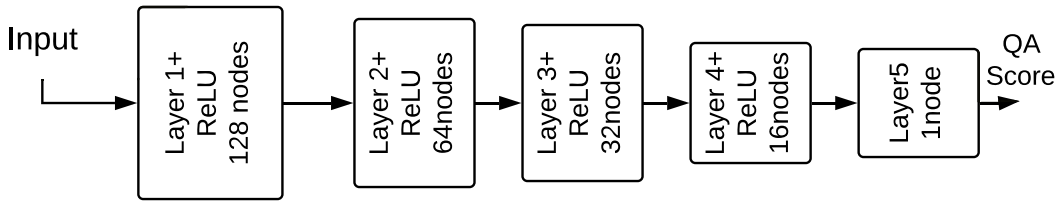


Fig. 4. Details of newly added fully connected layers.

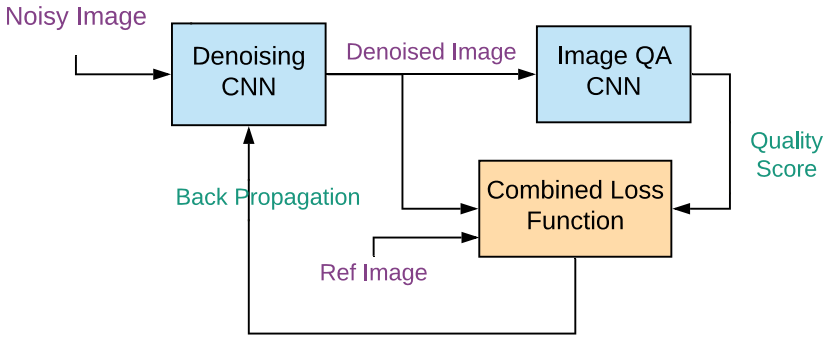


Fig. 5. Training phase of the denoising network with QAIR CNN.

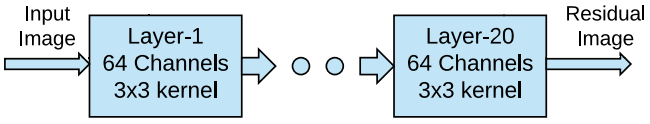


Fig. 6. Block diagram of denoising CNN [9].

4.2. Image super resolution for image retrieval

In addition to image denoising, we also consider the task of image super resolution for image retrieval. The framework we adopt is very similar to the above for image denoising, where the denoising CNN is replaced by the super resolution CNN. While several algorithms for single image super resolution exist in literature, we focus on CNN based approaches and optimize the CNNs for image retrieval using our QAIR network. In particular, we employ the DnCNN [9] used above for denoising, since it also achieves state of the art performance for image super resolution [9]. Given an up-sampled image using bi-cubic interpolation, the CNN is trained to predict the residual image, or the difference between the reference image and the upsampled image. The residual image is then added to the bi-cubic interpolated image to obtain the super resolved image. This CNN has 20 layers and each layer has 64 channels. All convolutional kernels are of size 3×3.

5. Feature adaptation for image retrieval

So far, we explored image enhancement for image retrieval. However, since retrieval is primarily based on image features, we now explore the problem of feature adaptation for image retrieval. The intuition behind this approach is that since features are ultimately used for retrieval, one could potentially perform better by adapting the features to account for distortions in addition to enhancing the images. We address this question in the context of a deep CNN based image retrieval algorithm [11], since it allows the flexibility to modify the features by changing the weights of the CNN. Thus, we consider whether we can improve retrieval performance by applying feature adaptation on top of image enhancement. Note that in Section 4, we fixed the feature vector and optimized image enhancement with respect to the given feature vector. However, we now fix the image enhancement and ask whether the feature extraction process can be optimized to improve image retrieval performance. In each of these two methods, different sets of parameters are optimized and one approach does not include the other.

5.1. Feature adaptation framework

We illustrate our framework for feature adaptation in Fig. 7. First, we pass the degraded image through an image enhancement network, potentially fine tuned as discussed in Section 4. The feature extraction procedure is then carried out on this enhanced image. The output of the feature extraction network is given as input to a feature QA CNN. We introduce the feature QA CNN to predict the performance of the features in terms of average precision as a measure of the success of the retrieval algorithm. A combined loss function based on the output of the feature QA CNN and the features of the enhanced image is used to fine tune the weights of the feature extraction network. The loss function is represented as

$$L = (1 - AP) + \lambda M, \tag{4}$$

where AP is the average precision predicted by the feature QA CNN and M is the mean square error between the present output of the feature extraction CNN and the initial output of the feature extraction CNN in its original configuration. The first term updates the feature extraction CNN such that the retrieval performance improves, while the second term ensures that the feature extraction CNN output does not deviate too much.

5.2. Feature QA network

We design the feature QA CNN to predict the average precision of the image given the output of the feature extraction CNN. The output of the feature extraction CNN, is a 4D tensor with shape $N \times 36 \times 36 \times 512$ where N is the batchsize. The feature QA CNN contains a global average pooling layer and five fully connected layers similar to Fig. 4. The global average pooling layer converts the 4D tensor to a 2D tensor. The fully connected layers are designed so that the size gradually reduces to 1 from 128. The first four fully connected layers have ReLU activation functions and the last layer has linear activation. A block diagram of the feature QA CNN is shown in Fig. 8.

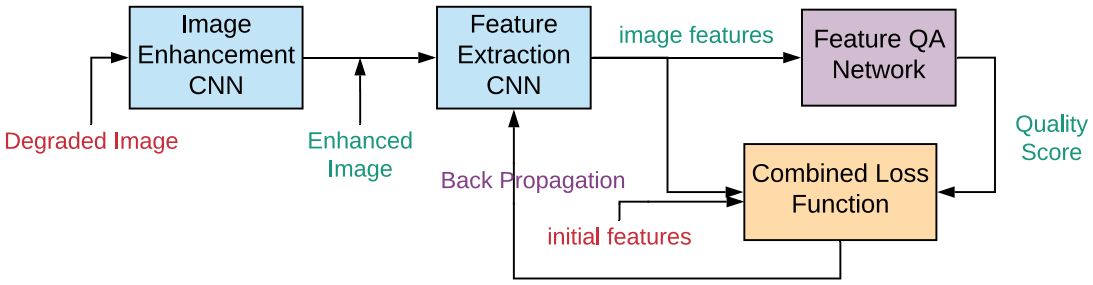


Fig. 7. Block diagram of feature enhancement framework.

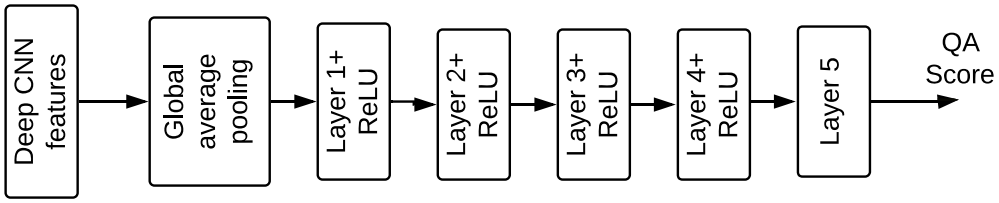


Fig. 8. Block diagram of feature QA network.

6. Experiments and results

We present the experimental results with respect to image denoising and super resolution for image retrieval and the feature adaptation framework. Before presenting the results, we describe the databases used for the experiments and the training details of the various CNNs involved. We use the SIFT based image retrieval algorithm described earlier for many of the experiments in the following unless otherwise stated.

6.1. Database

We use two standard image retrieval datasets in our experiments, the Extended Oxford buildings dataset [1] and the Paris landmarks dataset [41]. The extended Oxford buildings dataset has 5063 images of 11 different Oxford landmarks and one hundred thousand distractor images which makes a total of 105K images. Each landmark has 5 query images along with ground truth match details resulting in a total of 55 query images. The Paris dataset has 6412 images which contains 11 different Paris landmarks and 5 different query images per landmark. Out of the 55 query images in each dataset, 80%, i.e. 44 images and their distorted versions are used for training and the remaining 11 images and their distorted versions are used for testing. The train–test split is repeated across 5 iterations with a different split in each iteration such that there is no overlap in the content between the training and testing sets.

6.2. Training details

We adopt a patch based approach for image enhancement to use batch mode training. Therefore, each image is split into multiple patches of size 124 × 124. The enhancement and QAIR CNNs are trained and tested on patches. While training, all patches in an image are assigned the same quality score as the average precision on the full image. After enhancing, the image patches are merged to create the full image. All models are implemented in Python with Keras library using Tensorflow back end. We now describe the training of the CNNs used for denoising, super resolution and feature adaptation.

6.2.1. Image denoising

The image denoising CNN is trained on around 700K patches of size 50 × 50, generated from the image retrieval datasets. Different noise standard deviations are used for training the denoising CNN. We train the network for 40 epochs using Adam [42] optimizer.

During the fine tuning phase of the denoising CNN using the QAIR CNN, the images input to the QAIR CNN will be denoised images. Thus, the QAIR CNN needs to be trained on denoised images. Since the denoised images will have some amount of residual noise and blur artifacts, we create a dataset based on the 55 query images and obtain 10 different degraded versions of the same. The degradations include five different additive Gaussian noisy versions with noise standard deviation of $[3, 8, 14, 26, 44]$ and 3 blurred versions with a Gaussian blur kernel of standard deviations $[0.5, 1, 2]$. The degraded set also contains denoised images corresponding to a noise standard deviation of 50, denoised using the DnCNN [9] and BM3D algorithms [6]. The denoising CNN is fine tuned on noisy images that are split into patches of size 124 × 124. The fine tuning of the denoising CNN is performed for 40 epochs with SGD optimizer and a learning rate of 10^{-3}.

6.2.2. Image super resolution

The super resolution (SR) CNN is trained on patches of size 50 × 50. Around 700K image patches are used for training on each of the Oxford buildings dataset and the Paris dataset. The SR CNN is trained for a given super resolution factor (4 in our experiments) for 40 epochs using Adam optimizer with a mean square error loss function. The batch size is fixed to 100.

The output of the SR CNN may still have some amount of blur artefacts. We train the QAIR CNN for super resolution on varying degrees of blur to account for residual blur in the super resolved image. The degraded images include 3 blurred versions with Gaussian blur kernels of standard deviation of $[0.5, 1, 2]$ and upsampled images using bi-cubic interpolation for a scaling factor of 2 and 4. The SR CNN is fine tuned for 40 epochs using SGD optimizer with a learning rate of 10^{-3}.

6.2.3. Feature adaptation

We let the images processed by the DnCNN [9] as above for image enhancement pass through the feature adaptation CNN. In order to fine tune the feature extraction CNN, we first train the feature QA CNN on the output of the feature extraction CNN for different types of image distortions. For the denoising case, degraded versions include different levels of noisy and blurred images whereas for super resolution, the

Table 1
Mean absolute error (MAE) between predicted and actual QA scores for denoised images corresponding to noise standard deviation $\sigma = 50$.

Dataset	Oxford	Paris
MAE	0.086	0.060

Table 2
Mean absolute error (MAE) between predicted and actual QA scores for image super resolution by a factor of 4.

Dataset	Oxford	Paris
MAE	0.071	0.036

Table 3
Mean average precision for noisy and denoised images for noise standard deviation $\sigma = 50$.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN-QA
Oxford	0.632	0.487	0.590	0.589	0.585	**0.595**
Paris	0.633	0.562	**0.615**	0.610	0.608	0.611

degradations include different levels of blurred and downsampled–upsampled images. The feature QA network is trained using Adam optimizer with mean absolute error loss function for 60 epochs.

The fine tuning of the feature extraction network is done using degraded images of similar degradation levels on which we want to test. The feature extraction CNN is trained using SGD optimizer with a low learning rate of 3×10^{-4}.

6.3. QAIR performance

We evaluate the performance of the QAIR-CNN with respect to denoising in Table 1 by computing the mean absolute error between the actual average precision and predicted average precision. We test its quality prediction performance on images denoised using the CNN for a noise standard deviation of 50. As mentioned before, the evaluations are performed by the splitting the dataset of denoised images into training and testing in the ratio 80:20 ensuring no overlap of scene content between training and testing and averaging the performance across 5 iterations. The results indicate that the mean absolute error is quite low and the QAIR CNN is able to predict the image retrieval performance reasonably well.

Further, we measure the accuracy of prediction of the retrieval performance by the QAIR CNN with respect to super resolution in Table 2. The results indicate that the mean absolute error between the predicted and actual average precision scores is reasonable.

6.4. Image denoising

We now present the results of image denoising for image retrieval for different ranges of noise levels in the following subsections.

6.4.1. Denoising for noise standard deviation of 50

We first present the results of denoising for a noise standard deviation $\sigma = 50$. The results for Oxford and Paris dataset are given in Table 3. In this table, "Clean" refers to the mean average precision on the clean images, "Noisy" refers to the same on the noisy images, "BM3D" denotes the results of images denoised using BM3D algorithm, "NN-Org" refers to the pre-trained CNN as in [9], "NN" denotes the CNN trained by us using images in the retrieval datasets and "NN-QA" refers to the results of our method i.e. the fine tuned denoising CNN using the QA CNN.

As seen in the table, our method outperforms all other methods in Oxford dataset. On the Paris dataset, the performance of our method is slightly less than that of BM3D, but better than the pre-trained CNN and the CNN trained by us. We show examples of the noisy image and denoised images using different techniques in Fig. 9. The image denoised using our technique is visually sharper than the other images and also achieves a better average precision.

Table 4
Mean average precision for noisy and denoised images for noise standard deviation $\sigma = 90$.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN-QA
Oxford	0.632	0.158	0.455	NA	0.483	**0.499**
Paris	0.633	0.367	0.530	NA	0.556	**0.562**

Table 5
Mean average precision for noisy and denoised images for noise standard deviation in the range 30–60.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN-QA
Oxford	0.632	0.539	0.605	0.602	0.605	**0.618**
Paris	0.633	0.589	**0.622**	0.617	0.618	0.619

6.4.2. Denoising for a noise standard deviation of 90

We evaluate the performance of our algorithm at a higher noise level of $\sigma = 90$. The denoising CNN is trained on the dataset for this noise level. The mean average precision for different methods are given in Table 4. The results show that our denoising method is superior to all other denoising methods in terms of image retrieval performance. We note that the improvements with respect to the other methods are slightly higher for $\sigma = 90$ when compared to $\sigma = 50$.

6.4.3. Denoising for variable noise levels

While so far we evaluated our algorithm on fixed noise levels, we now test our algorithm for noise standard deviations belonging to a range between 30 and 60. We do not consider noise levels corresponding to standard deviation less than 30 since there is very minimal drop in the retrieval performance at those noise levels. Here, the denoising CNN is trained on a random subset of noise standard deviations in the range of 30 to 60. The results of our method and other methods for the variable noise level case is given in Table 5.

Across the three sets of results described above, we observe that BM3D is competitive and even achieves slightly better performance than our framework sometimes on the Paris dataset. We believe that the Paris dataset has a larger set of matching blocks which lends itself to superior denoising performance of BM3D. However, we note that for the BM3D algorithm, we need to specify the exact noise standard deviation while the other methods do not require such knowledge. We observe that on the Oxford dataset, our method performs better than all the other methods and on the Paris dataset, the performance of our method is only slightly less than that of BM3D which requires knowledge of the noise standard deviation.

6.5. Image retrieval using SURF features

We now evaluate the performance of our method for another image retrieval method based on Speeded Up Robust Features (SURF) [43]. SURF is a speeded up version of SIFT and uses box filters to approximate difference of Gaussian. The feature length is 64 for SURF, in comparison to 128 for SIFT. Initially, the SURF features are computed for all images in the database and stored. A given query image is then compared with all images in the database based on the number of matching SURF features and a geometric consistency check.

We present results of this method on the Paris dataset for noise levels of standard deviation $\sigma = 50$ and $\sigma = 90$. The image retrieval results in Table 6 reveal that for $\sigma = 50$, our method performs almost as well as the original DnCNN, and better than the DnCNN trained by us. For $\sigma = 90$, our method outperforms all other methods. The performance of "NN-Org" is marked as "NA" since that CNN is not trained for this noise level.

Fig. 9. (a) Noisy image for $\sigma = 50$; (b) denoised image using BM3D; (c) denoised image using the pre-trained CNN in [9]; (d) denoised image using the CNN in [9] trained by us (e) denoised image using a CNN which is trained in combination with QA network.

Table 6
Mean average precision for noisy and denoised images for SURF based image retrieval.

σ	Clean	Noisy	BM3D	NN-Org	NN	NN-QA
50	0.392	0.251	0.355	**0.357**	0.352	0.355
90	0.392	0.057	0.222	NA	0.259	**0.267**

Table 7
Mean average precision for noisy and denoised images for CNN features based image retrieval.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN-QA
Oxford	0.451	0.014	**0.220**	0.141	0.214	0.219
Paris	0.658	0.212	**0.513**	0.436	0.486	0.487

6.6. Image retrieval using deep CNN features

We also test our image enhancement framework on a different image retrieval method using deep features [11]. In this method, a pre-trained deep CNN, VGG-19 [12], is used to extract features from the images. The last convolutional layer output of the network is used as features. A block diagram of the feature extraction and aggregation steps for image retrieval are shown in Fig. 10. The experimental results for $\sigma = 50$ on both the Oxford and Paris datasets are shown in Table 7. While the improvements with respect to other learning methods are marginal and there is a performance gap with respect to BM3D, we show later on that the performance of our framework can be further improved using feature adaptation.

6.7. Super resolution results

We now present the performance analysis of our enhancement framework for super resolution in Table 8. In the table, "Original" refers to the actual query image, "Down sample by 4" refers to the image down sampled by 4, "Up sample bicubic 4" refers to the image which is down sampled and then bi-cubic interpolated by a factor of 4, "SR NN" refers to the images output by the SR CNN, "SR CNN QA" refers to the output of the fine tuned SR CNN using the QAIR CNN. The results indicate that our method outperforms all other methods on both the Oxford and Paris datasets. We also observe that super resolution of downsampled images can sometimes improve the retrieval performance. We believe this can be viewed as some preprocessing of the image before image retrieval that can improve retrieval performance.

An example image and its different SR versions are given in Fig. 11. We see that the image enhanced using our method is sharper than other images leading to a performance improvement in terms of average precision.

Table 8
Mean average precision for image super resolution by a factor of 4.

Dataset	Original	Down sample by 4	Up sample bi-cubic 4	SR NN	SR NN QA
Oxford	0.632	0.265	0.604	0.620	**0.626**
Paris	0.633	0.323	0.619	0.635	**0.642**

Table 9
Mean average precision for both noisy and low resolution (LR) images using CNN based enhancement and image retrieval.

Dataset	Clean	Noisy/LR	NN-Org	NN	NN-QA
Paris	0.6330	0.6020	0.6128	0.6308	**0.6322**

Table 10
Mean average precision for noisy and denoised images for noise standard deviation $\sigma = 50$.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN + tweaked features
Oxford	0.451	0.014	0.220	0.141	0.214	**0.230**
Paris	0.658	0.212	**0.513**	0.436	0.486	0.507

6.8. Denoising and super resolution using the same network

We also conduct an experiment where both denoising and super resolution are achieved using a single enhancement CNN on the Paris dataset. The corresponding QAIR CNN is trained on a mix of denoised, noisy, blurred and low resolution images upsampled using bicubic interpolation and CNN based algorithms. Further, the enhancement CNN was trained as before on 8 distorted versions of each query image, with 4 noisy images with noise standard deviation between 30 and 60 and 4 low resolution images with downsampling factors of 3.8, 3.9, 4 or 4.1. The test set consists of 2 distorted versions of each query image with one noisy image with standard deviation between 30 and 60 and 1 low resolution image with downsampling factor of 4. The same train–test split among query images explained earlier across multiple iterations was used. The results in Table 9 indicate that the "NN-QA" approach does indeed offer benefits when compared to not using the QAIR CNN for training the enhancement CNN.

6.9. Feature adaptation

We now analyze the performance of the feature adaptation framework for both denoising and super resolution.

6.9.1. Feature adaptation for noisy images with $\sigma = 50$

The image retrieval performance on the Oxford and Paris dataset are shown in Table 10. In this table, "NN+tweaked features" refers to the results of our proposed method in which the images are passed through a denoising CNN and the feature extraction CNN is fine tuned

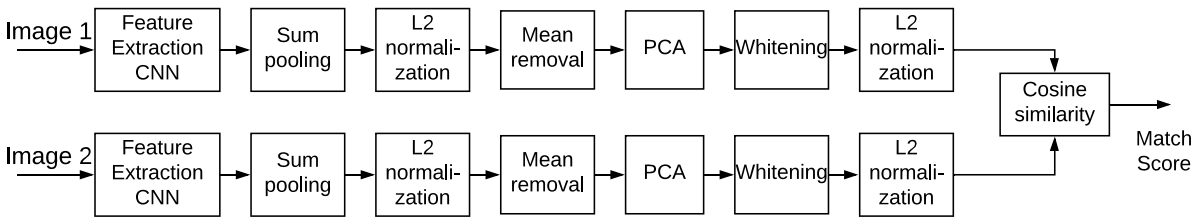


Fig. 10. Block diagram of CNN based feature extraction and image matching.

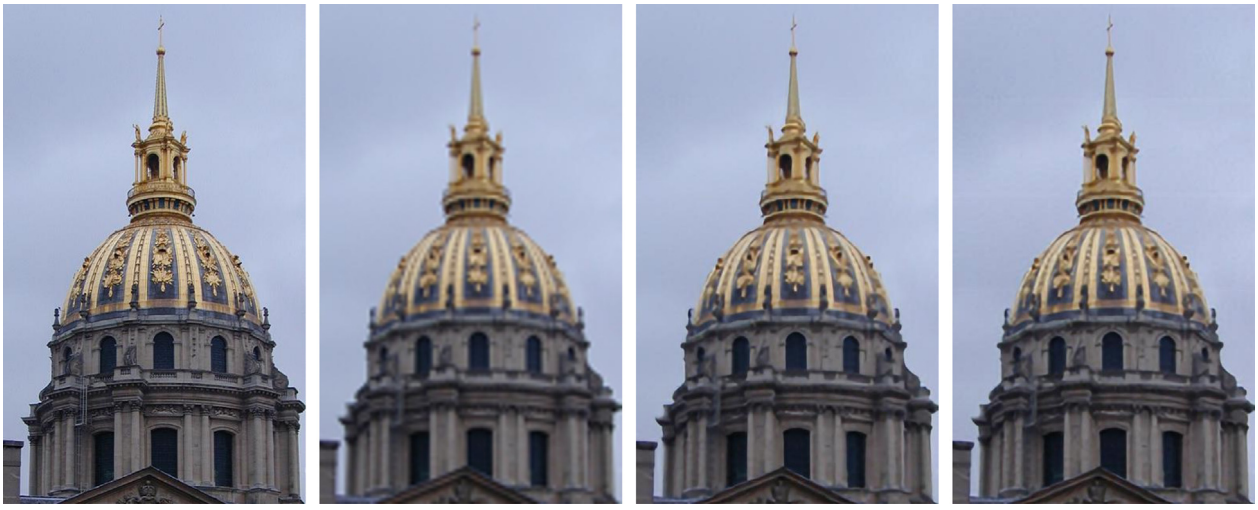


Fig. 11. (a) Original Image; (b) Image interpolated using bicubic interpolation; (c) super resolution image using the CNN in [9] trained by us (d) super resolution image using a CNN which is trained in combination with QA network.

Table 11
Mean average precision for noisy and denoised images for noise standard deviation in the range 30 to 60.

Dataset	Clean	Noisy	BM3D	NN-Org	NN	NN + tweaked features
Oxford	0.451	0.047	0.235	0.156	0.222	**0.255**
Paris	0.658	0.276	**0.536**	0.468	0.513	0.533

using a feature QA CNN. The results shows that fine tuning the feature extraction CNN can lead to a performance improvement. We also note that the feature enhancement method performs better when compared to image enhancement in Table 7 for the same scenario.

6.9.2. Feature adaptation for noisy images with σ in the range $[30, 60]$

We now analyze how the feature adaption method performs when we train for noise levels in a range instead of a single noise level. Here, we train for noise values with standard deviation in the range $[30, 60]$. The feature QA network is trained in a similar setup as that of the previous section. The results in Table 11 indicate that there is a good improvement in both datasets for our method when compared to the case where features are not fine tuned.

6.9.3. Feature adaptation results for image super resolution by a factor of 4

We also test the feature adaptation framework for the super resolution case. We design the feature adaptation framework by first passing the low resolution image though a super resolution network and then through a fine tuned feature extraction network. The image SR CNN and its training method are same as that in Section 4.2. The feature QA network is trained on query images and different degraded versions of the same as discussed before. The results obtained on the Oxford and Paris dataset are given in Table 12. In the table, "SR NN + tweaked features" refers to the performance of SR CNN output using enhanced features. Again, the results show that our method out performs all other methods.

In summary, we observe that our image enhancement and feature adaptation frameworks yield improvements in the mean average precision in several scenarios. In other scenarios, the performance is almost as good as the best performing enhancement method.

7. Conclusion and future work

In this work, we developed a framework for image enhancement for image retrieval by defining a relevant notion of image quality. The quality of a query image for image retrieval is defined as the area under the precision–recall curve for that image and we designed a deep CNN based method for image quality prediction. By modeling the image enhancement as a deep CNN, we can fine tune such a network for the success of image retrieval. Note that this particular modeling is not very restrictive owing to the success of deep CNN methods in a variety of image processing applications. We showed the benefits of such an approach for two image enhancement cases, image denoising and image super resolution.

We also developed a framework for feature adaptation to improve the image retrieval performance of degraded images using a feature QA network. This framework of feature adaptation is applicable for

Table 12
Mean average precision for image super resolution by a factor of 4.

Dataset	Original	Down sample by 4	Up sample bi-cubic 4	SR NN	SR NN + tweaked features
Oxford	0.451	0.015	0.150	0.166	**0.186**
Paris	0.658	0.296	0.397	0.496	**0.537**

deep CNN features based image retrieval. We tested such an approach for both image denoising and image super resolution cases and were successful in showing that fine tuning the feature extraction framework using a feature QA network leads to better image retrieval performance.

While we showed the utility of our framework, one could potentially improve the results by further enhancing the prediction of average precision. Moreover, we considered the homogenous distortions and a patch based approach for enhancement. It will be interesting to study enhancement operations that operate on the entire image with the average precision of the entire image. In order to attempt such an approach, methods for predicting the average precision using much lesser data may need to be explored. Further, the framework can be extended to study other enhancement settings such as low light enhancement, defogging and so on.

While we considered the specific case of image retrieval, our framework can also be used to study image enhancement for various other computer vision applications. Our framework is particularly useful when the performance of the computer vision task is arbitrary and cannot be modeled by closed form expressions of the output of a deep CNN. Thus, by converting any computer vision task performance to a quality assessment CNN, one could potentially optimize image enhancement for the relevant computer vision task.

Acknowledgment

This research was supported by a grant from Robert Bosch Center for Cyber Physical Systems (RBCCPS), Indian Institute of Science.

References

[1] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[3] J. Philbin, M. Isard, J. Sivic, A. Zisserman, Descriptor learning for efficient retrieval, in: European Conference on Computer Vision, 2010, pp. 677-691.

[4] R. Arandjelović, A. Zisserman, Three things everyone should know to improve object retrieval, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2911-2918.

[5] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, Int. J. Comput. Vis. 87 (3) (2010) 316–336.

[6] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[7] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: International Conference on Computer Vision, Nov 2011, pp. 479-486.

[8] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Trans. Image Process. 26 (7) (2017) 3142–3155.

[10] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[11] A.B. Yandex, V. Lempitsky, Aggregating local deep features for image retrieval, in: 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 1269-1277.

[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556.

[13] B.V. Somasundaran, R. Soundararajan, S. Biswas, Image denoising for image retrieval by cascading a deep quality assessment network, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Oct 2018, ser. ICIP '18, 2018.

[14] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial-bag-of-features, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3352-3359.

[15] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in CVPR 2011, 2011, pp. 809-816.

[16] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2011) 1704–1716.

[17] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (2003) 1338–1351.

[18] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[19] H.C. Burger, C.J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with bm3d? in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 2392-2399.

[20] K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a fast and flexible solution for CNN based image denoising, CoRR, 2017. [Online]. Available: http://arxiv.org/abs/1710.04026.

[21] M. Irani, S. Peleg, Improving resolution by image registration, CVGIP: Graph. Models Image Process. 53 (3) (1991) 231–239, [Online]. Available: http://dx.doi.org/10.1016/1049-9652(91)90045-L.

[22] W.T. Freeman, T.R. Jones, E.C. Pasztor, Example-based super-resolution, IEEE Comput. Graph. Appl. 22 (2002) 56–65.

[23] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: 2009 IEEE 12th International Conference on Computer Vision, Sept 2009, pp. 349-356.

[24] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, CoRR, vol. abs/1501.00092, 2015. [Online]. Available: http://arxiv.org/abs/1501.00092.

[25] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 1646-1654.

[26] M.A. Saad, A.C. Bovik, C. Charrier, Blind image quality assessment: A natural scene statistics approach in the dct domain, IEEE Trans. Image Process. 21 (8) (2012) 3339–3352.

[27] M.A. Saad, A.C. Bovik, C. Charrier, Blind prediction of natural video quality, IEEE Trans. Image Process. 23 (3) (2014) 1352–1365.

[28] A. Mittal, A.K. Moorthy, A.C. Bovik, No-reference image quality assessment in the spatial domain, IEEE Trans. Image Process. 21 (12) (2012) 4695–4708.

[29] P. Ye, J. Kumar, L. Kang, D. Doermann, Unsupervised feature learning framework for no-reference image quality assessment, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 1098-1105.

[30] L. Kang, P. Ye, Y. Li, D. Doermann, Convolutional neural networks for no-reference image quality assessment, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, in: CVPR '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 1733–1740, [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.224.

[31] S. Bianco, L. Celona, P. Napoletano, R. Schettini, On the use of deep learning for blind image quality assessment, CoRR, vol. abs/1602.05531, 2016. [Online]. Available: http://arxiv.org/abs/1602.05531.

[32] K. Ma, W. Liu, T. Liu, Z. Wang, D. Tao, DipIQ: Blind image quality assessment by learning-to-rank discriminable image pairs, IEEE Trans. Image Process. 26 (8) (2017) 3951–3964.

[33] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, W. Zuo, End-to-end blind image quality assessment using deep neural networks, IEEE Trans. Image Process. 27 (3) (2017) 1202–1213.

[34] S. Bosse, D. Maniry, K. Muller, T. Wiegand, W. Samek, Deep neural networks for no-reference and full-reference image quality assessment, IEEE Trans. Image Process. 27 (1) (2018) 206–219.

[35] S. Gunasekar, J. Ghosh, A.C. Bovik, Face detection on distorted images augmented by perceptual quality-aware features, IEEE Trans. Inf. Forensics Secur. 9 (12) (2014) 2119–2131.

[36] R. Soundararajan, S. Biswas, Machine vision quality assessment for robust face detection, Signal Process., Image Commun. 72 (2019) 92–104.

[37] D.M. Rouse, R. Pépion, S.S. Hemami, P.L. Callet, Image utility assessment and a relationship with image quality assessment, in: Human Vision and Electronic Imaging, 2009.

[38] A. Gala, S. Shah, Joint modeling of algorithm behavior and image quality for algorithm performance prediction, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2010, pp. 31.1–31.11, http://dx.doi.org/10.5244/C.24.31.

[39] D. Liu, B. Wen, X. Liu, Z. Wang, T. Huang, When image denoising meets high-level vision tasks: A deep learning approach, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 842–848, [Online]. Available: https://doi.org/10.24963/ijcai.2018/117.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. (IJCV) 115 (3) (2015) 211–252.

[41] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[42] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[43] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Comput. Vis. Image Underst. 110 (3) (2008) 346–359, [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014.

Composing Text and Image for Image Retrieval - An Empirical Odyssey

Nam Vo[1]*, Lu Jiang[2], Chen Sun[2], Kevin Murphy[2]
Li-Jia Li[2,3], Li Fei-Fei[2,3], James Hays[1]
[1]Georgia Tech, [2]Google AI, [3]Stanford University

Abstract

In this paper, we study the task of image retrieval, where the input query is specified in the form of an image plus some text that describes desired modifications to the input image. For example, we may present an image of the Eiffel tower, and ask the system to find images which are visually similar, but are modified in small ways, such as being taken at nighttime instead of during the day. To tackle this task, we embed the query (reference image plus modification text) and the target (images). The encoding function of the image text query learns a representation, such that the similarity with the target image representation is high iff it is a "positive match". We propose a new way to combine image and text through residual connection, that is designed for this retrieval task. We show this outperforms existing approaches on 3 different datasets, namely Fashion-200k, MIT-States and a new synthetic dataset we create based on CLEVR. We also show that our approach can be used to perform image classification with compositionally novel labels, and we outperform previous methods on MIT-States on this task.

1. Introduction

A core problem in image retrieval is that the user has a "concept" in mind, which they want to find images of, but they need to somehow convey that concept to the system. There are several ways of formulating the concept as a search query, such as a text string, a similar image, or even a sketch, or some combination of the above. In this work, we consider the case where queries are formulated as an input image plus a text string that describes some desired modification to the image. This represents a typical scenario in session search: users can use an already found image as a reference, and then express the difference in text, with the aim of retrieving a relevant image. This problem is closely related to attribute-based product retrieval (see e.g., [12]), but differs in that the text can be multi-word, rather than a single attribute.

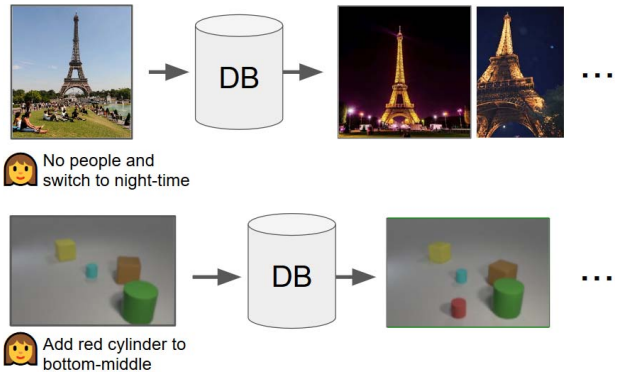


Figure 1. Example of image retrieval using text and image query. The text states the desired modification to the image and is expressive in conveying the information need to the system.

We can use standard deep metric learning methods such as triplet loss (e.g., [15]) for computing similarity between a search query and candidate images. The main research question we study is how to represent the query when we have two different input modalities, namely the input image and the text. In other words, how to learn a meaningful cross-modal feature composition for the query in order to find the target image.

Feature composition between text and image has been extensively studied in the field of vision and language, especially in Visual Question Answering (VQA) [2]. After encoding an image (e.g., using a convolutional neural network, or CNN) and the text (e.g., using a recurrent neural network, or RNN), various methods for feature composition have been used. These range from simple techniques (e.g., concatenation or shallow feed-forward networks) to advanced mechanisms (e.g., relation [43], or parameter hashing [35]). These approaches have also been successfully used in related problems such as query classification, compositional learning, etc. (See Section 2 for more discussion of related work.)

The question of which image/text feature composition to use for image retrieval has not been studied, to the best of our knowledge. In this paper, we compare several existing methods, and propose a new one, which often gives

*Work done during an internship at Google AI.

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00660

improved results. The key idea behind the new method is that the text should modify the features of the query image, but we want the resulting feature vector to still "live in" the same space as the target image. We achieve this goal by having the text modify the image feature via a gated residual connection. We call this "Text Image Residual Gating " (or TIRG for short). We give the details in Section 3.

We empirically compare these methods on three benchmarks: Fashion-200k dataset from [12], MIT-States dataset [17], and a new synthetic dataset for image retrieval, which we call "CSS" (color, shape and size), based on the CLEVR framework [20]. We show that our proposed feature combination method outperforms existing methods in all three cases. In particular, significant improvement is made on Fashion-200k compared to [12] whose approach is not ideal for this image retrieval task. Besides, our method works reasonably well on a recent task of learning feature composition for image classification [31, 33], and achieves the state-of-the-art result on the task on the MIT-States dataset [17].

To summarize, our contribution is threefold:

- We systematically study feature composition for image retrieval, and propose a new method.
- We create a new dataset, CSS, which we will release, which enables controlled experiments of image retrieval using text and image queries.
- We improve previous state of the art results for image retrieval and compositional image classification on two public benchmarks, Fashion-200K and MIT-States.

2. Related work

Image retrieval and product search: Image retrieval is an important vision problem and significant progress has been made thanks to deep learning [5, 51, 10, 38]; it has numerous applications such as product search [28], face recognition [44, 36] or image geolocalization [13]. Cross-modal image retrieval allows using other types of query, examples include text to image retrieval [52], sketch to image retrieval [42] or cross view image retrieval [26], and event detection [19]. We consider our set up an image to image retrieval task, but the image query is augmented with an additional modification text input.

A lot of research has been done to improve product retrieval performance by incorporating user's feedback to the search query in the form of relevance [40, 18], relative [23] or absolute attribute [56, 12, 1]. Tackling the problem of image based fashion search, Zhao *et al*. [56] proposed a memory-augmented deep learning system that can perform attribute manipulation. In [12], spatially-aware attributes are automatically learned from product description labels and used to facilitate attribute-feedback product retrieval application. We are approaching the same image search task, but incorporating text into the query instead, which can be potentially more flexible than using a predefined set of attribute values. Besides, unlike previous work which seldom shows its effectiveness beyond image retrieval, we show our method also work reasonably well for a classification task on compositional learning.

Parallel to our work is dialog-based interactive image retrieval [11], where Guo *et al*. showed promising result on simulated user and real world user study. Though the task is similar, their study focuses on modeling the interaction between user and the agent; meanwhile we study and benchmark different image text composition mechanisms.

Vision question answering: The task of Visual Question Answering (VQA) has achieved much attention (see e.g., [2, 20]). Many techniques have been proposed to combine the text and image inputs effectively [7, 34, 22, 35, 37, 43, 27, 48, 25]. Fukui et al [7] proposed Multimodal Compact Bilinear Pooling as a feature fusion mechanism to combine image and text. In [35], the text feature is incorporated by mapping into parameters of a fully connected layer within the image CNN. Another important tool that's proved effective for VQA task is attention [34, 48, 27]. In [22, 37], residual connections are used to combine image and text. Specifically, [22] proposed method outputs the text feature plus a residual mapping obtained by joint element-wise multiplication of image and text. [37] introduced FiLM layer as a way to inject text features into an image CNN, notably by residual connections. While using similar technical components, we actually try to keep and "modify" the input image feature, instead of "fusing" it with text creating a "brand new" feature.

Vision and Language: beside VQA, there's other tasks that also learn to make prediction from image and text input. Chen et al [4] proposed a recurrent attentive model to edit and colorize images given text descriptions. [16, 29, 54, 53] study the referring expression comprehension task, which aim to localize the object in the input image given its reference description.

Compositional Learning: We can think of our query as a composition of an image and a text. The core of compositional learning is that a complex concept can be developed by combing multiple simple concepts or attributes [31]. The idea is reminiscent of earlier work on visual attribute [6, 41] and also related to zero-shot learning [24, 39, 55]. Among recent contributions, Misra *et al*. [31] investigated learning a composition classifier by combining an existing object classifier and attribute classifier. Nagarajan *et al*. [33] proposed an embedding approach to carry out the composition using the attribute embedding as an operator to change the object classifier. Kota *et al*. [21] applied this idea to action recognition. By contrast, our composition is cross-modal and only has a single image versus abundant training exam-

ples to train the classifiers.

3. Method

As explained in the introduction, our goal is to learn an embedding space for the text+image query and for target images, such that matching (query, image) pairs are close (see Fig. 2).

First, we encode the query (or reference) image x using a ResNet-17 CNN to get a 2d spatial feature vector $f_{\text{img}}(x) = \phi_x \in \mathbb{R}^{W\times H\times C}$, where W is the width, H is the height, and $C = 512$ is the number of feature channels. Next we encode the query text t using a standard LSTM. We define $f_{\text{text}}(t) = \phi_t \in \mathbb{R}^d$ to be the hidden state at the final time step whose size d is 512. We want to keep the text encoder as simple as possible. Encoding texts by other encoders, e.g. bi-LSTM or LSTM attention, is definitely feasible but beyond the scope of our paper. Finally, we combine the two features to compute $\phi_{xt} = f_{\text{combine}}(\phi_x, \phi_t)$. Below we discuss various ways to perform this combination.

3.1. Summary of existing combination methods

In this paper, we study the following approaches for feature composition. For a fair comparison, we train all methods including ours using the same pipeline, with the only difference being in the composition module.

- Image Only: we set $\phi_{xt} = \phi_x$.
- Text Only: we set $\phi_{xt} = \phi_t$.
- Concatenate computes $\phi_{xt} = f_{\text{MLP}}([\phi_x, \phi_t])$. This simple has proven effective in a variety of applications [2, 11, 56, 31]. In particular, we use two layers of MLP with RELU, the batch-norm and the dropout rate of 0.1.
- Show and Tell [49]. In this approach, we train an LSTM to encode both image and text by inputting the image feature first, following by words in the text; the final state of this LSTM is used as representation ϕ_{xt}.
- Attribute as Operator [33] embeds each text as a transformation matrix, T_t, and applies T_t to ϕ_x to create ϕ_{xt}.
- Parameter hashing [35] is a technique used for the VQA task. In our implementation, the encoded text feature ϕ_t is hashed into a transformation matrix T_t, which can be applied to image feature; it is used to replace a fc layer in the image CNN, which now outputs a representation ϕ_{xt} that takes into account both image and text feature.
- Relationship [43] is a method to capture relational reasoning in the VQA task. It first uses CNN to extract a 2D feature map from image, then create a set of relationship features, each is a concatenation of the text feature ϕ_t and 2 local features in the 2D feature map; this set of features is passed through an MLP and the result is summed to get a single feature. Another MLP is applied to obtain the output ϕ_{xt}.
- Multimodal Residual Networks (MRN) [22] is a VQA method that uses element-wise multiplication for the joint residual mappings. Here starting with $\phi^0_{xt} = \phi_t$, each of its block layer is defined as $\phi^i_{xt} = \phi^{i-1}_{xt} + fc(tanh(\phi^{i-1}_{xt}))\cdot fc(tanh(fc(tanh(\phi_x))))$. The last feature is linearly transformed to obtain the image text composition output.
- FiLM [37] is another VQA method where the text feature is also injected into the image CNN. In more detail, the text feature ϕ_t is used to predict modulation features: $\gamma^i, \beta^i \in \mathbb{R}^C$, where i indexes the layer and C is the number of feature or feature map. Then it performs a feature-wise affine transformation of the image features, $\phi^i_{xt} = \gamma^i \cdot \phi^i_x + \beta^i$. As stated in [37], a FiLM layer only handles a simple operation like scaling, negating or thresholding the feature. To perform complex operations, it has to be used in every layer of the CNN. By contrast, we only modify one layer of the image feature map, and we do this using a gated residual connection, described in 3.2.

3.2. Proposed approach: TIRG

Inspired by [47, 14, 30], we propose to combine image and text features using the following approach which we call Text Image Residual Gating (or TIRG for short).

$$\phi^{rg}_{xt} = w_g f_{\text{gate}}(\phi_x, \phi_t) + w_r f_{\text{res}}(\phi_x, \phi_t), \quad (1)$$

where $f_{\text{gate}}, f_{\text{res}} \in \mathbb{R}^{W\times H\times C}$ are the gating and the residual features shown in Fig. (2). w_g, w_r are learnable weights to balance them. The gating connection is computed by:

$$f_{\text{gate}}(\phi_x, \phi_t) = \sigma(W_{g2} * \text{RELU}(W_{g1} * [\phi_x, \phi_t]) \odot \phi_x \quad (2)$$

where σ is the sigmoid function, $\odot$ is element wise product, $*$ represents 2d convolution with batch normalization, and W_{g1} and W_{g2} are 3x3 convolution filters. Note that we broadcast ϕ_t along the height and width dimension so that its shape is compatible to the image feature map ϕ_x. The residual connection is computed by:

$$f_{\text{res}}(\phi_x, \phi_t) = W_{r2} * \text{RELU}(W_{r1} * ([\phi_x, \phi_t])), \quad (3)$$

The intuition is that we want to "modify" the query image feature instead of traditional "feature fusion" that creates a new feature from existing ones. This is facilitated by the ResBlock design: the gated identity establishes the input image feature as a reference to the output composition feature, as if they were in the same meaningful image feature space; then the added residual connection represents the modification or "walk" in this feature space.

When training, it essentially starts off as a working image to image retrieval system, then gradually learn meaningful modification. Differently, other methods would start off with random retrieval result at the beginning.

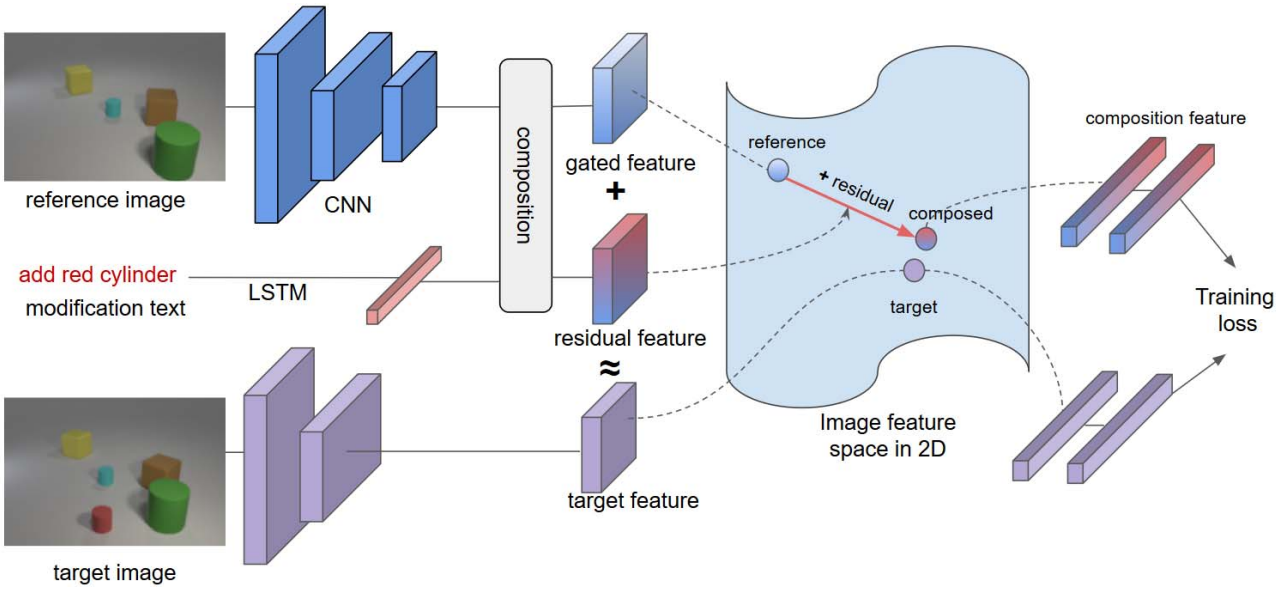


Figure 2. The system pipeline for training. We show a 2d feature space for visual simplicity.

Fig. 2 shows modification applied to the convolutional layer of the CNN. However, we can alternatively apply modification to the fully-connected layer (where $W = H = 1$) to alter the non-spatial properties of the representation. In our experiments, we modify the last fc layer for Fashion200k and MIT-States, since the modification is more global and abstract. For CSS, we modify the last 2D feature map before pooling (last conv layer) to capture the low-level and spatial changes inside the image. The choice of which layer to modify is a hyperparameter of the method and can be chosen based on a validation set.

3.3. Deep Metric Learning

Our training objective is to push closer the features of the "modified" and target image, while pulling apart the features of non-similar images. We employ a classification loss for this task. More precisely, suppose we have a training minibatch of B queries, where $\psi_i = f_{\text{combine}}(x_i^{query}, t_i)$ is the final modified representation (from the last layer) of the image text query, and $\phi_i^+ = f_{\text{img}}(x_i^{target})$ is the representation of the target image of that query. We create a set $\mathcal{N}_i$ consisting of one positive example ϕ_i^+ and $K-1$ negative examples $\phi_1^-, \ldots, \phi_{K-1}^-$ (by sampling from the minibatch ϕ_j^+ where j is not i). We repeat this M times, denoted as $\mathcal{N}_i^m$, to evaluate every possible set. (The maximum value of M is $\binom{B}{K}$, but we often use a smaller value for tractability.) We then use the following softmax cross-entropy loss:

$$L = \frac{-1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} \log\{\frac{\exp\{\kappa(\psi_i, \phi_i^+)\}}{\sum_{\phi_j \in \mathcal{N}_i^m} \exp\{\kappa(\psi_i, \phi_j)\}}\}, \quad (4)$$

where κ is a similarity kernel and is implemented as the dot product or the negative l_2 distance in our experiments. When we use the smallest value of $K = 2$, Eq. (4) can be easily rewritten as:

$$L = \frac{1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} \log\{1 + \exp\{\kappa(\psi_i, \phi_{i,m}^-) - \kappa(\psi_i, \phi_i^+)\}\}, \quad (5)$$

since each set $\mathcal{N}_i^m$ contains a single negative example. This is equivalent to the the soft triplet based loss used in [50, 15]. When we use $K = 2$, we choose $M = B - 1$, so we pair each example i with all possible negatives.

If we use larger K, each example is contrasted with a set of other negatives; this loss resembles the classification based loss used in [9, 46, 32, 45, 8]. With the largest value $K = B$, we have $M = 1$, so the function is simplified as:

$$L = \frac{1}{B} \sum_{i=1}^{B} -\log\{\frac{\exp\{\kappa(\psi_i, \phi_i^+)\}}{\sum_{j=1}^{B} \exp\{\kappa(\psi_i, \phi_j^+)\}}\}, \quad (6)$$

In our experience, this case is more discriminative and fits faster, but can be more vulnerable to overfitting. As a result, we set $K = B$ for Fashion200k since it is more difficult to converge and $K = 2$ for other datasets. Ablation studies on K are shown in Table 5.

4. Experiments

We perform our empirical study on three datasets: Fashion200k [12], MIT-States [17], and a new synthetic dataset we created called CSS (see Section 4.3). Our main metric for retrieval is recall at rank k (R@K), computed as the percentage of test queries where (at least 1) target or correct

labeled image is within the top K retrieved images. Each experiment is repeated 5 times to obtain a stable retrieval performance, and both mean and standard deviation are reported. In the case of MIT-States, we also report classification results.

We use PyTorch in our experiments. We use ResNet-17 (output feature size = 512) pretrained on ImageNet as our image encoder and the LSTM (hidden size is 512) of random initial weights as our text encoder. By default, training is run for 150k iterations with a start learning rate 0.01. We will release the code and CSS dataset to the public. Using the same training pipeline, we implement and compare various methods for combining image and text, described in section 3.1, with our feature modification via residual values, described in section 3.2, denoted as **TIRG**.

4.1. Fashion200k

Fashion200k [12] is a challenging dataset consisting of ∼200k images of fashion products. Each image comes with a compact attribute-like product description (such as black biker jacket or wide leg culottes trouser). Following [12], queries are created as following: pairs of products that have one word difference in their descriptions are selected as the query images and target images; and the modification text is that one different word. We used the same training split (around 172k images) and generate queries on the fly for training. To compare with [12], we randomly sample 10 validation sets of 3,167 test queries (hence in total 31,670 test queries) and report the mean.[1].

Table 1 shows the results, where the recall of the first row is from [12] and the others are from our framework. We see that our pipeline even with different kind of composition mechanisms outperforms their approach. We believe this is because they perform image text joint embedding training, instead of attribute-feedback or text-modification image retrieval training. In terms of the different ways of computing ϕ_{xt}, we see that our approach performs the best. Some qualitative retrieval examples are shown in Fig. 3.

4.2. MIT-States

MIT-States [17] has ∼60k images, each comes with an object/noun label and a state/adjective label (such as "red tomato" or "new camera"). There are 245 nouns and 115 adjectives, on average each noun is only modified by ∼9 adjectives it affords. We use it to evaluate both image retrieval and image classification, as we explain below.

[1] We contacted the authors of [12] for the original 3,167 test queries, but got only the product descriptions. We attempted to recover the set from the description. However, on average, there are about 3 product images for each unique product description.

Method	R@1	R@10	R@50
Han *et al*. [12]	6.3	19.9	38.3
Image only	3.5	22.7	43.7
Text only	1.0	12.3	21.8
Concatenation	$11.9^{\pm 1.0}$	$39.7^{\pm 1.0}$	$\underline{62.6}^{\pm 0.7}$
Show and Tell	$12.3^{\pm 1.1}$	$40.2^{\pm 1.7}$	$61.8^{\pm 0.9}$
Param Hashing	$12.2^{\pm 1.1}$	$40.0^{\pm 1.1}$	$61.7^{\pm 0.8}$
Relationship	$13.0^{\pm 0.6}$	$\underline{40.5}^{\pm 0.7}$	$62.4^{\pm 0.6}$
MRN	$\underline{13.4}^{\pm 0.4}$	$40.0^{\pm 0.8}$	$61.9^{\pm 0.6}$
FiLM	$12.9^{\pm 0.7}$	$39.5^{\pm 2.1}$	$61.9^{\pm 1.9}$
TIRG	$\mathbf{14.1}^{\pm 0.6}$	$\mathbf{42.5}^{\pm 0.7}$	$\mathbf{63.8}^{\pm 0.8}$

Table 1. Retrieval performance on Fashion200k. The best number is in bold and the second best is underlined.

Method	R@1	R@5	R@10
Image only	$3.3^{\pm 0.1}$	$12.8^{\pm 0.2}$	$20.9^{\pm 0.1}$
Text only	$7.4^{\pm 0.4}$	$21.5^{\pm 0.9}$	$32.7^{\pm 0.8}$
Concatenation	$11.8^{\pm 0.2}$	$30.8^{\pm 0.2}$	$42.1^{\pm 0.3}$
Show and Tell	$11.9^{\pm 0.1}$	$31.0^{\pm 0.5}$	$42.0^{\pm 0.8}$
Att. as Operator	$8.8^{\pm 0.1}$	$27.3^{\pm 0.3}$	$39.1^{\pm 0.3}$
Relationship	$\mathbf{12.3}^{\pm 0.5}$	$\mathbf{31.9}^{\pm 0.7}$	$\underline{42.9}^{\pm 0.9}$
MRN	$11.9^{\pm 0.6}$	$30.5^{\pm 0.3}$	$41.0^{\pm 0.2}$
FiLM	$10.1^{\pm 0.3}$	$27.7^{\pm 0.7}$	$38.3^{\pm 0.7}$
TIRG	$\underline{12.2}^{\pm 0.4}$	$\mathbf{31.9}^{\pm 0.3}$	$\mathbf{43.1}^{\pm 0.3}$

Table 2. Retrieval performance on MIT-States.

4.2.1 Image retrieval

We use this dataset for image retrieval as follows: pairs of images with the same object labels and different state labeled are sampled. They are using as query image and target image respectably. The modification text will be the state of the target image. Hence the system is supposed to retrieve images of the same object as the query image, but with the new state described by text. We use 49 of the nouns for testing, and the rest is for training. This allows the model to learn about state/adjective (modification text) during training and has to deal with unseen objects presented in the test query.

Some qualitative results are shown in Fig. 4 and the R@K performance is shown in Table 2. Note that similar types of objects with different states can look drastically different, making the the role of modification text more important. Hence on this dataset, the [Text Only] baseline outperforms [Image Only]. Nevertheless, combining them gives better results. The difference between composition methods is not too significant here. Still TIRG is comparable to Relationship while outperforming others.

4.2.2 Classification with compositionally novel labels

To be able to compare to prior work on this dataset, we also consider the classification setting proposed in [31, 33]. The goal is to learn models to recognize unseen combination of

Figure 3. Retrieval examples on Fashion200k dataset.

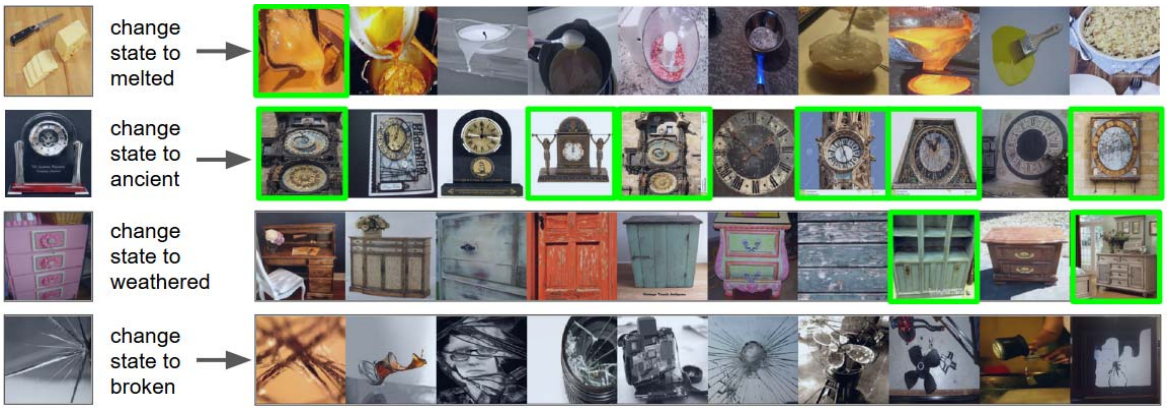


Figure 4. Some retrieval examples on MIT-States.

(state, noun) pairs. For example, training on "red wine" and "old tomato" to recognize "red tomato" where there exist no "red tomato" images in training.

To tackle this in our framework, we define ϕ_x to be the feature vector derived from the image x (using ResNet-17 as before), and ϕ_t to be the feature vector derived from the text t. The text is composed of two words, a noun n and an adjective a. We learn an embedding for each of these words, ϕ_n and ϕ_a, and then use our TIRG method to compute the combination ϕ_{an}. Given this, we perform image classification using nearest neighbor retrieval, so $t(x) = \arg\max_t \kappa(\phi_t, \phi_x)$, where κ is a similarity kernel applied to the learned embeddings. (In contrast to our other experiments, here we embed text and image into the same shared space.)

The results, using the same compositional split as in [31, 33], are shown in Table 3. Even though this problem is not the focus of our study, we see that our method outperforms prior methods on this task. The difference from the previous best method, [33], is that their composition feature is represented as a dot product between adjective transformation matrix and noun feature vector; by contrast, we represent both adjective and noun as feature vectors and combine them using our composition mechanism.

Method	Accuracy
Analogous Attribute [3]	1.4
Red wine [31]	13.1
Attribute as Operator [33]	14.2
VisProd NN [33]	13.9
Label Embedded+ [33]	14.8
TIRG	**15.2**

Table 3. Comparison to the state-of-the-art on the unseen combination classification task on MIT-States. All baseline numbers are from previous works.

4.3. CSS dataset

Since existing benchmarks for image retrieval do not contain complex text modifications, we create a new dataset, as we describe below.

4.3.1 Dataset Description

We created a new dataset using the CLEVR toolkit [20] for generating synthesized images in a 3-by-3 grid scene. We render objects with different *C*olor, *S*hape and *S*ize (CSS) occupy. Each image comes in a simple 2D blobs version and a 3D rendered version. Examples are shown in Fig. 5.

We generate three types of modification texts from templates: adding, removing or changing object attributes. The "add object" modification specifies a new object to be placed in the scene (its color, size, shape, position). If any of the attribute is not specified, its value will be randomly chosen. Examples are "add object", "add red cube", "add big red cube to middle-center". Likewise, the "remove object" modification specifies the object to be removed from the scene. All objects that match the specified attribute values will be removed, e.g. "remove yellow sphere", "remove middle-center object". Finally, the "change object" modification specifies the object to be changed and its new attribute value. The new attribute value has to be either color or size. All objects that match the specified attribute will be changed, e.g. "make yellow sphere small", "make middle-center object red".

In total, we generate 16K queries for training and 16K queries for test. Each query is of a reference image (2D or 3D) and a modification, and the target image. To be specific, we first generate 1K random scenes as the reference. Then we randomly generate modifications and apply them to the reference images, resulting in a set of 16K target images. In this way, one reference image can be transformed to multiple different target images, and one modification can be applied to multiple different reference images. We then repeat the process to generate the test images. We follow the protocol proposed in [20] in which certain object shape and color combinations only appear in training, and not in testing, and vice versa. This provides a stronger test of generalization.

Although the CSS dataset is simple, it allows us to perform controlled experiments, with multi-word text queries, similar to the CLEVR dataset. In particular, we can create queries using a 2d image and text string, to simulate the case where the user is sketching something, and then wants to modify it using language. We can also create queries using slightly more realistic 3d image and text strings.

4.3.2 Results

Table 4 summarizes R@1 retrieval performance on the CSS dataset. We examine two retrieval settings using 3d query images (2nd column) and 2d images (3rd column). As we can see, our TIRG combination outperforms other composition methods for the retrieval task. In addition, we see that retrieving a 3D image from a 2D query is much harder, since the feature spaces are quite different. (In these experiments, we use different feature encoders for the 2D and 3D inputs). Some qualitative results are shown in Fig. 6.

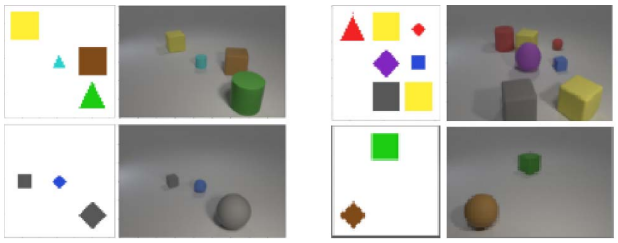


Figure 5. Example images in our CSS dataset. The same scene are rendered in 2D and 3D images.

Method	3D-to-3D	2D-to-3D
Image only	6.3	6.3
Text only	0.1	0.1
Concatenate	$60.6^{\pm 0.8}$	27.3
Show and Tell	$33.0^{\pm 3.2}$	6.0
Parameter hashing	$60.5^{\pm 1.9}$	31.4
Relationship	$62.1^{\pm 1.2}$	30.6
MRN	$60.1^{\pm 2.7}$	26.8
FiLM	$\underline{65.6}^{\pm 0.5}$	$\underline{43.7}$
TIRG	$\mathbf{73.7}^{\pm 1.0}$	**46.6**

Table 4. Retrieval performance (R@1) on the CSS Dataset using 2D and 3D images as the query.

To gain more insight into the nature of the combined features, we trained a transposed convolutional network to reconstruct the images from their features and then apply it to composition feature. Fig. 7 shows the reconstructed images from the composition features of three methods. Images generated from our feature representation look visually better, and are closer to the top retrieved image. We see that all the images are blurry as we use the regression loss to train the network. However, a nicer reconstruction may not mean better retrieval, as the composition feature is learned to capture the discriminative information need to find the target image, and this may be a lossy representation.

4.4. Ablation Studies

Method	Fashion	MIT-States	CSS
Our Full Model	14.1	12.2	73.7
- gated feature only	13.9	07.1	06.5
- residue feature only	12.1	11.9	60.6
- mod. at last fc	14.1	12.2	71.2
- mod. at last conv	12.4	10.3	73.7
DML loss, $K = 2$	9.5	12.2	73.7
DML loss, $K = B$	14.1	10.9	69.8

Table 5. Retrieval performance (R@1) of ablation studies.

In this section, we report the results of various ablation studies, to gain insight into which parts of our approach matter the most. The results are in Table 5.

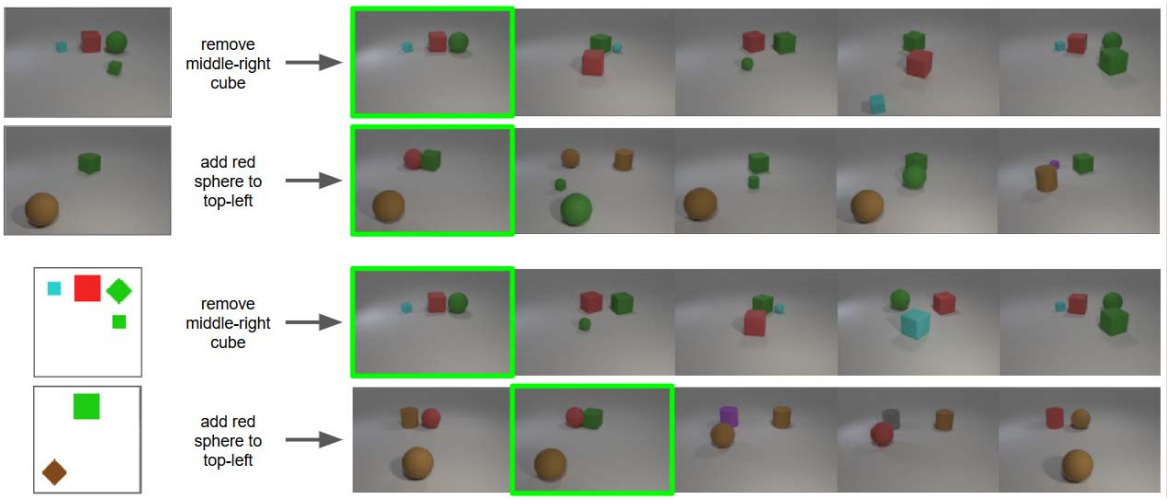


Figure 6. Some retrieval examples on CSS Dataset.

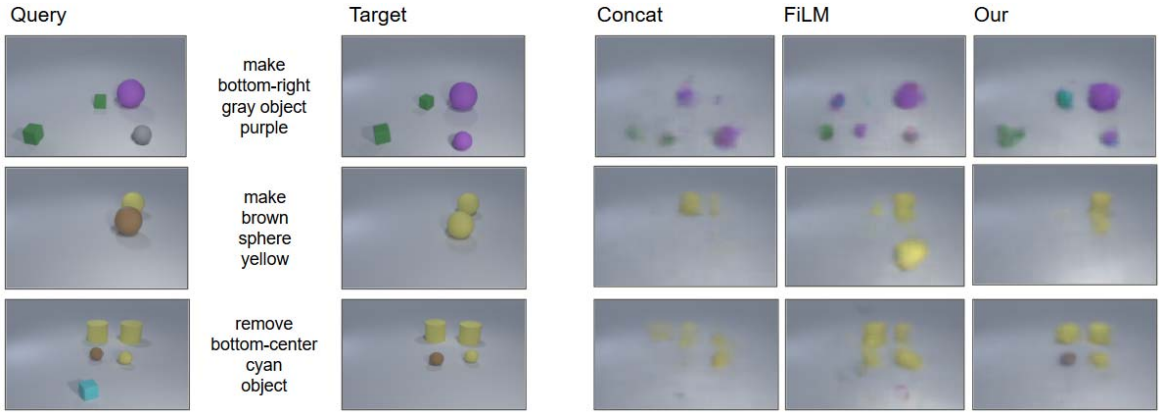


Figure 7. Reconstruction images from the learned composition features.

Efficacy of feature modification: as shown in Fig. 2, our composition module has two types of connections, namely residual connection and gated connection. Row 2 and 3 show that removing the residual features or gating features leads to drops in performance. In these extreme cases, our model can degenerate to the concatenate fusion baseline.

Spatial versus non-spatial modification: Row 5 and 6 compares the effect of applying our feature modification to the last fc layer versus the last convolution layer. When our modification is applied to the last fc layer feature, it yields competitive performance compared to the baseline across all datasets. Applying the modification to the last convolution feature map only improves the performance on CSS. We believe this is because the modifications in the CSS dataset is more spatially localized (see Fig. 6) whereas they are more global on the other two datasets (See Fig. 3 and Fig. 4)

The impact of K in the loss function: The last two rows compares the loss function of two different K values in Section 3.3. We use $K = 2$ (soft triplet loss) in most experiments. As Fashion200k is much bigger, we found that the network underfitted. In this case by using $K = B$ (same as batch size in our experiment), the network fits well and produces better results. On the other two datasets, test time performance is comparable, but training becomes less stable. Note that the difference here regards our metric learning loss and does not reflect the difference between the feature composition methods.

5. Conclusion

In this work, we explored the composition of image and text in the context of image retrieval. We experimentally evaluated several existing methods, and proposed a new one, which gives improved performance on three benchmark datasets. In the future, we would like to try to scale this method up to work on real image retrieval systems "in the wild".

References

[1] K. E. Ak, A. A. Kassim, J. H. Lim, and J. Y. Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR*, 2018. 2

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 1, 2, 3

[3] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *CVPR*, 2014. 6

[4] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018. 2

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2

[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2

[7] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[8] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 4

[9] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2005. 4

[10] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 2

[11] X. Guo, H. Wu, Y. Cheng, S. Rennie, and R. S. Feris. Dialog-based interactive image retrieval. *arXiv preprint arXiv:1805.00145*, 2018. 2, 3

[12] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 1, 2, 4, 5

[13] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[15] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1, 4

[16] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 2

[17] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2, 4, 5

[18] L. Jiang, A. G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *ACM MM*, 2012. 2

[19] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *ICMR*, 2015. 2

[20] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2, 7

[21] K. Kato, Y. Li, and A. Gupta. Compositional learning for human object interaction. In *ECCV*, 2018. 2

[22] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *NIPS*, 2016. 2, 3

[23] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012. 2

[24] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2

[25] J. Liang, L. Jiang, L. Cao, L.-J. Li, and A. Hauptmann. Focal visual-text attention for visual question answering. In *CVPR*, 2018. 2

[26] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, 2015. 2

[27] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun. ivqa: Inverse visual question answering. In *CVPR*, 2018. 2

[28] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 2

[29] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2

[30] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 3

[31] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 2, 3, 5, 6

[32] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 4

[33] T. Nagarajan and K. Grauman. Attributes as operators. 2018. 2, 3, 5, 6

[34] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017. 2

[35] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, 2016. 1, 2, 3

[36] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015. 2

[37] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. 2018. 2, 3

[38] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 2

[39] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[40] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998. 2

[41] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 2

[42] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016. 2

[43] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 1, 2, 3

[44] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[45] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 4

[46] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016. 4

[47] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015. 3

[48] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*, 2018. 2

[49] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 3

[50] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, 2016. 4

[51] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[52] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2

[53] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 2

[54] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 2

[55] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 2

[56] B. Zhao, J. Feng, X. Wu, and S. Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *CVPR*, 2017. 2, 3

2019 IEEE/CVF International Conference on Computer Vision (ICCV)

CAMP: Cross-Modal Adaptive Message Passing for Text-Image Retrieval

Zihao Wang[1*] Xihui Liu[1*] Hongsheng Li[1] Lu Sheng[3] Junjie Yan[2] Xiaogang Wang[1] Jing Shao[2]
[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]SenseTime Research [3]Beihang University
zihaowang@cuhk.edu.hk {xihuiliu, hsli, xgwang}@ee.cuhk.edu.hk
lsheng@buaa.edu.cn {yanjunjie, shaojing}@sensetime.com

Abstract

Text-image cross-modal retrieval is a challenging task in the field of language and vision. Most previous approaches independently embed images and sentences into a joint embedding space and compare their similarities. However, previous approaches rarely explore the interactions between images and sentences before calculating similarities in the joint space. Intuitively, when matching between images and sentences, human beings would alternatively attend to regions in images and words in sentences, and select the most salient information considering the interaction between both modalities. In this paper, we propose Cross-modal Adaptive Message Passing (CAMP), which adaptively controls the information flow for message passing across modalities. Our approach not only takes comprehensive and fine-grained cross-modal interactions into account, but also properly handles negative pairs and irrelevant information with an adaptive gating scheme. Moreover, instead of conventional joint embedding approaches for text-image matching, we infer the matching score based on the fused features, and propose a hardest negative binary cross-entropy loss for training. Results on COCO and Flickr30k significantly surpass state-of-the-art methods, demonstrating the effectiveness of our approach. [1]

1. Introduction

Text-image cross-modal retrieval has made great progress recently [16, 9, 22, 5, 4]. Nevertheless, matching images and sentences is still far from being solved, because of the large visual-semantic discrepancy between language and vision. Most previous work exploits visual-semantic embedding, which independently embeds images and sentences into the same embedding space, and then measures their similarities by feature distances in the joint space [11, 5]. The model is trained with ranking loss, which forces the similarity of positive pairs to be higher than that of negative pairs. However, such independent embedding approaches do not exploit the interaction between images and sentences, which might lead to suboptimal features for text-image matching.

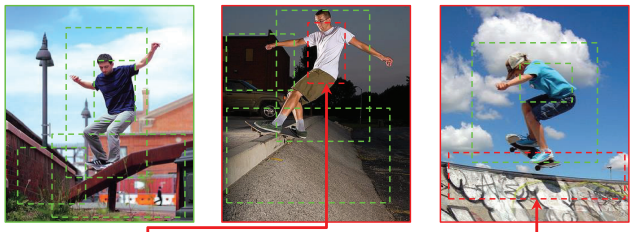


Figure 1. Illustration of how our model distinguish the subtle differences by cross-modal interactions. Green denotes positive evidence, while red denotes negative cross-modal evidence.

Let us consider how we would perform the task of text-image matching ourselves. Not only do we concentrate on salient regions in the image and salient words in the sentence, but also we would alternatively attend to information from both modalities, take the interactions between regions and words into consideration, filter out irrelevant information, and find the fine-grained cues for cross-modal matching. For example, in Figure 1, all of the three images seem to match with the sentence at first glance. When we take a closer observation, however, we would notice that the sentence describes "blue shirt" which cannot be found in the second image. Similarly, the description of "a railing not far from a brick wall" cannot be found in the third image. Those fine-grained misalignments can only be noticed if we have a gist of the sentence in mind when looking at the images. As a result, incorporating the interaction between images and sentences should benefit in capturing the fine-grained cross-modal cues for text-image matching.

In order to enable interactions between images and sentences, we introduce a ***C**ross-modal **A**daptive **M**essage **P**assing* model (CAMP), composed of the *Cross-modal Message Aggregation* module and the *Cross-modal Gated Fusion* module. Message passing for text-image retrieval is

*The first two authors contributed equally to this work.
[1]https://github.com/ZihaoWang-CV/CAMP_iccv19

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.00586

non-trivial and essentially different from previous message passing approaches, mainly because of the existing of negative pairs for matching. If we pass cross-modal messages between negative pairs and positive pairs in the same manner, the model would get confused and it would be difficult to find alignments that are necessary for matching. Even for matched images and sentences, information unrelated to text-image matching (*e.g.*, background regions that are not described in the sentence) should also be suppressed during message passing. Hence we need to adaptively control to what extent the messages from the other modality should be fused with the original features. We solve this problem by exploiting a soft gate for fusion to adaptively control the information flow for message passing.

The **Cross-Modal Message Aggregation module** aggregates salient visual information corresponding to each word as messages passing from visual to textual modality, and aggregates salient textual information corresponding to each region as messages from textual to visual modality. The Cross-modal Message Aggregation is done by cross-modal attention between words and image regions. Specifically, we use region features as cues to attend on words, and use word features as cues to attend on image regions. In this way, we interactively process the information from visual and textual modalities in the context of the other modality, and aggregate salient features as messages to be passed across modalities. Such a mechanism considers the word-region correspondences and empowers the model to explore the fine-grained cross-modal interactions.

After aggregating messages from both modalities, the next step is fusing the original features with the aggregated messages passed from the other modality. Despite the success of feature fusion in other problems such as visual question answering [7, 8, 13, 32, 23], cross-modal feature fusion for text-image retrieval is nontrivial and has not been investigated before. In visual question answering, we only fuse the features of images and corresponding questions which are matched to the images. For text-image retrieval, however, the key challenge is that the input image-sentence pair does not necessarily match. If we fuse the negative (mismatched) pairs, the model would get confused and have trouble figuring out the misalignments. Our experiments indicate that naïve fusion approach does not work for text-image retrieval. To filter out the effects of negative (mismatched) pairs during fusion, we propose a novel **Cross-modal Gated Fusion module** to adaptively control the fusion intensity. Specifically, when we fuse the original features from one modality with the aggregated message passed from another modality, a soft gate adaptively controls to what extent the information should be fused. The aligned features are fused to a larger extent. While non-corresponding features are not intensively fused, and the model would preserve original features for negative pairs. The Cross-modal Gated Fusion module incorporates deeper and more comprehensive interactions between images and sentences, and appropriately handles the effect of negative pairs and irrelevant background information by an adaptive gate.

With the fused features, a subsequent question is: how to exploit the fused cross-modal information to infer the text-image correspondences? Since we have a joint representation consisting of information from both images and sentences, the assumption that visual and textual features are respectively embedded into the same embedding space no longer holds. As a result, we can no longer calculate the feature distance in the embedding space and train with ranking loss. We directly predict the cross-modal matching score based on the fused features, and exploit binary cross-entropy loss with hardest negative pairs as training supervision. Such reformulation gives better results, and we believe that it is superior to embedding cross-modal features into a joint space. By assuming that features from different modalities are separately embedded into the joint space, visual semantic embedding naturally prevents the model from exploring cross-modal fusion. On the contrary, our approach is able to preserve more comprehensive information from both modalities, as well as fully exploring the fine-grained cross-modal interactions.

To summarize, we introduce a Cross-modal Adaptive Message Passing model, composed of the Cross-modal Message Aggregation module and the Cross-modal Gated Fusion module, to adaptively explore the interactions between images and sentences for text-image matching. Furthermore, we infer the text-image matching score based on the fused features, and train the model by a hardest negative binary cross-entropy loss, which provides an alternative to conventional visual-semantic embedding. Experiments on COCO [17] and Flickr30k [11] validate the effectiveness of our approach.

2. Related Work

Text-image retrieval. Matching between images and sentences is the key to text-image cross-modal retrieval. Most previous works exploited visual-semantic embedding to calculate the similarities between image and sentence features after embedding them into the joint embedding space, which was usually trained by ranking loss [14, 27, 28, 15, 6, 4, 25, 11]. Faghri *et al*. [5] improved the ranking loss by introducing the hardest negative pairs for calculating loss. Zheng *et al*. [34] explored text CNN and instance loss to learn more discriminative embeddings of images and sentences. Zhang *et al*. [33] used projection classification loss which categorized the vector projection of representations from one modality onto another with the improved norm-softmax loss. Niu *et al*. [24] exploited a hierarchical LSTM model for learning visual-semantic embedding. Huang *et*

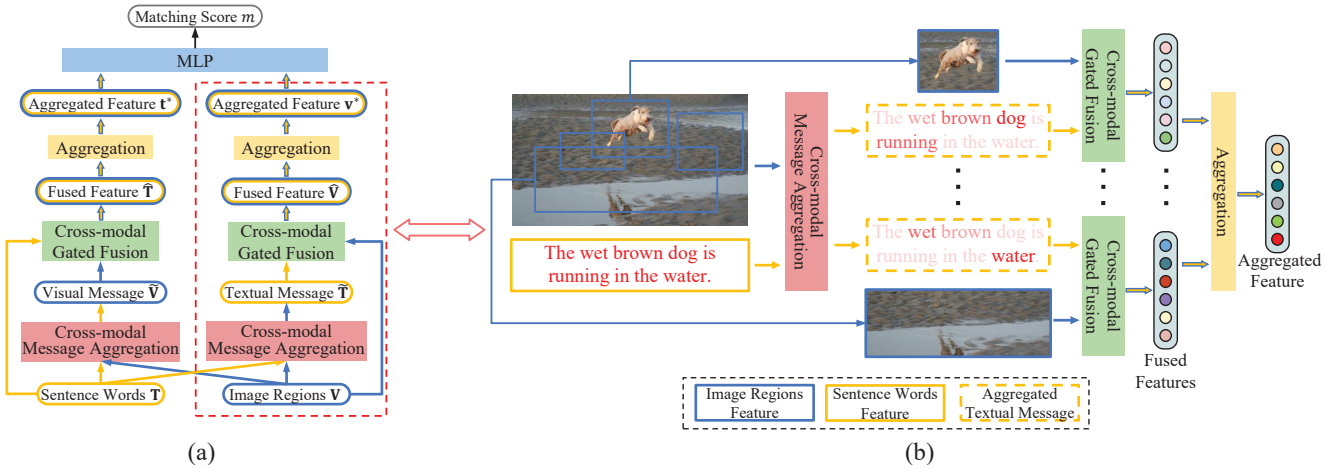


Figure 2. (a) is the overview of the Cross-modal Adaptive Message Passing model. The input regions and words interact with each other and are aggregated to fused features to predict the matching score. (b) is an illustration of the message passing from textual to visual modality (the dashed red box in (a)). Word features are aggregated based on the cross-modal attention weights, and the aggregated textual messages are passed to fuse with the region features. The message passing from visual to textual modality operates in a similar way.

al. [10] proposed a model to learn semantic concepts and order for better image and sentence matching. Gu *et al*. [9] leveraged generative models to learn concrete grounded representations that capture the detailed similarity between the two modalities. Lee *et al*. [16] proposed stacked cross attention to exploit the correspondences between words and regions for discovering full latent alignments. Nevertheless, the model only attends to either words or regions, and it cannot attend to both modalities symmetrically. Different from previous methods, our model exploits cross-modal interactions by adaptive message passing to extract the most salient features for text-image matching.

Interactions between language and vision. Different types of interactions have been explored in language and vision tasks beyond text-image retrieval [32, 2, 20, 35, 12, 29, 21, 18, 19]. Yang *et al*. [30] proposed stacked attention networks to perform multiple steps of attention on image feature maps. Anderson *et al*. [1] proposed bottom-up and top-down attention to attend to uniform grids and object proposals for image captioning and visual question answering (VQA). Previous works also explored fusion between images and questions [7, 8, 13, 32, 23] in VQA. Despite the great success in other language and vision tasks, few works explore the interactions between sentences and images for text-image retrieval, where the main challenge is to properly handle the negative pairs. To our best knowledge, this is the first work to explore deep cross-modal interactions between images and sentences for text-image retrieval.

3. Cross-modal Adaptive Message Passing

In this section, we introduce our Cross-modal Adaptive Message Passing model to enable deep interactions between images and sentences, as shown in Fig. 2. The model is composed of two modules, *Cross-modal Message Aggregation* and *Cross-modal Gated Fusion*. Firstly we introduce the Cross-modal Message Aggregation based on cross-modal attention, and then we consider fusing the original information with aggregated messages passed from the other modality, which is non-trivial because fusing the negative (mismatched) pairs makes it difficult to find informative alignments. We introduce our Cross-modal Gated Fusion module to adaptively control the fusion of aligned and misaligned information.

Problem formulation and notations. Given an input sentence $\mathcal{C}$ and an input image $\mathcal{I}$, we extract the word-level textual features $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_N] \in \mathbb{R}^{d \times N}$ for N words in the sentence and region-level visual features $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_R] \in \mathbb{R}^{d \times R}$ for R region proposals in the image.[2] Our objective is to calculate the matching score between images and sentences based on $\mathbf{V}$ and $\mathbf{T}$.

3.1. Cross-modal Message Aggregation

We propose a Cross-modal Message Aggregation module which aggregates the messages to be passed between regions and words. The aggregated message is obtained by a cross-modal attention mechanism, where the model takes the information from the other modality as cues to attend to the information from the self modality. In particular, our model performs word-level attention based on the cues from region features, and performs region-level attention based on the cues from word features. Such a message aggregation enables the information flow between textual and visual information, and the cross-modal attention for aggregating messages selects the most salient cross-modal information specifically for each word/region.

Mathematically, we first project region features and word features to a low dimensional space, and then compute the

[2]The way of extracting word and region features is described in Sec 4.1.

region-word affinity matrix,

$$\mathbf{A} = (\tilde{\mathbf{W}}_v \mathbf{V})^\top (\tilde{\mathbf{W}}_t \mathbf{T}), \tag{1}$$

where $\tilde{\mathbf{W}}_v, \tilde{\mathbf{W}}_s \in \mathbb{R}^{d_h \times d}$ are projection matrices which project the d-dimensional region or word features into a d_h-dimensional space. $\mathbf{A} \in \mathbb{R}^{R \times N}$ is the region-word affinity matrix where $\mathbf{A}_{ij}$ represents the affinity between the ith region and the jth word. To derive the attention on each region with respect to each word, we normalize the affinity matrix over the image region dimension to obtain a word-specific region attention matrix,

$$\tilde{\mathbf{A}}_v = \mathrm{softmax}(\frac{\mathbf{A}^\top}{\sqrt{d_h}}), \tag{2}$$

where the ith row of $\tilde{\mathbf{A}}_v$ is the attention over all regions with respect to the ith word. We then aggregate all region features with respect to each word based on the word-specific region attention matrix,

$$\tilde{\mathbf{V}} = \tilde{\mathbf{A}}_v \mathbf{V}^\top, \tag{3}$$

where the ith row of $\tilde{\mathbf{V}} \in \mathbb{R}^{N \times d}$ denotes the visual features attended by the ith word.

Similarly, we can calculate the attention weights on each word with respect to each image region, by normalizing the affinity matrix $\mathbf{A}$ over the word dimension. And based on the region-specific word attention matrix $\tilde{\mathbf{A}}_s$, we aggregate the word features to obtain the textual features attended by each region $\tilde{\mathbf{T}} \in \mathbb{R}^{R \times d}$,

$$\tilde{\mathbf{A}}_t = \mathrm{softmax}(\frac{\mathbf{A}}{\sqrt{d_h}}), \quad \tilde{\mathbf{T}} = \tilde{\mathbf{A}}_t \mathbf{T}^\top. \tag{4}$$

Intuitively, the ith row of $\tilde{\mathbf{V}}$ represents the visual features corresponding to the ith word, and the jth row of $\tilde{\mathbf{T}}$ represents the textual features corresponding to the jth region. Such a message aggregation scheme takes cross-modal interactions into consideration. $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{T}}$ are the aggregated messages to be passed from visual features to textual features, and from textual features to visual features, respectively.

3.2. Cross-modal Gated Fusion

The Cross-modal Message Aggregation module aggregates the most salient cross-modal information for each word/region as messages to be passed between textual and visual modalities, and the process of aggregating messages enables the interactions between modalities. However, with such a mechanism, the word and region features are still aggregated from each modality separately, without being fused together. To explore deeper and more complex interactions between images and sentences, the next challenge we face is how to fuse the information from one modality with the messages passed from the other modality.

However, conventional fusion operation assumes that the visual and textual features are matched, which is not the case for text-image retrieval. Directly fusing between the negative (mismatched) image-sentence pairs may lead to meaningless fused representation and may impede training and inference. Experiments also indicate that fusing the negative image-sentence pairs degrades the performance. To this end, we design a novel *Cross-modal Gated Fusion* module, as shown in Fig. 3, to adaptively control the cross-modal feature fusion. More specifically, we want to fuse textual and visual features to a large extent for matched pairs, and suppress the fusion for mismatched pairs.

By the aforementioned Cross-modal Adaptive Message Passing module, we obtain the aggregated message $\tilde{\mathbf{V}}$ passed from visual to textual modality, and the aggregated message $\tilde{\mathbf{T}}$ passed from textual to visual modality. Our Cross-modal Gated Fusion module fuses $\tilde{\mathbf{T}}$ with the original region-level visual features $\mathbf{V}$ and fuses $\tilde{\mathbf{T}}$ with the original word-level textual features $\mathbf{T}$. We denote the fusion operation as $\oplus$ (*e.g.* element-wise add, concatenation, element-wise product). In practice, we use element-wise add as the fusion operation. In order to filter out the mismatched information for fusion, a region-word level gate adaptively controls to what extent the information is fused.

Take the fusion of original region features $\mathbf{V}$ and messages passed from the textual modality $\tilde{\mathbf{T}}$ as an example. Denote the ith region features as $\mathbf{v}_i$ (the ith column of $\mathbf{V}$), and denote the attended sentence features with respect to the ith region as $\tilde{\mathbf{t}}_i^\top$ (the ith row of $\tilde{\mathbf{T}}$). $\tilde{\mathbf{t}}_i^\top$ is the message to be passed from the textual modality to the visual modality. We calculate the corresponding gate as,

$$\mathbf{g}_i = \sigma(\mathbf{v}_i \odot \tilde{\mathbf{t}}_i^\top), \ i \in \{1, \cdots, R\}. \tag{5}$$

where $\odot$ denotes the element-wise product, $\sigma(\cdot)$ denotes the sigmoid function, and $\mathbf{g}_i \in \mathbb{R}^d$ is the gate for fusing $\mathbf{v}_i$ and $\tilde{\mathbf{t}}_i^\top$. With such a gating function, if a region matches well with the sentence, it will receive high gate values which encourage the fusion operation. On the contrary, if a region does not match well with the sentence, it will receive low gate values, suppressing the fusion operation. We represent the region-level gates for all regions as $\mathbf{G}_v = [\mathbf{g}_1, \cdots, \mathbf{g}_R] \in \mathbb{R}^{d \times R}$ We then use these gates to control how much information should be passed for cross-modality fusion. In order to preserve original information for samples that should not be intensively fused, the fused features are further integrated with the original features via a residual connection.

$$\hat{\mathbf{V}} = \mathcal{F}_v(\mathbf{G}_v \odot (\mathbf{V} \oplus \tilde{\mathbf{T}}^\top)) + \mathbf{V}, \tag{6}$$

where $\mathcal{F}_v$ is a learnable transformation composed of a linear layer and non-linear activation function. $\odot$ denotes element-wise product, $\oplus$ is the fusing operation (element-wise sum), and $\hat{\mathbf{V}}$ is the fused region features. For positive pairs where the regions match well with the sentence, high

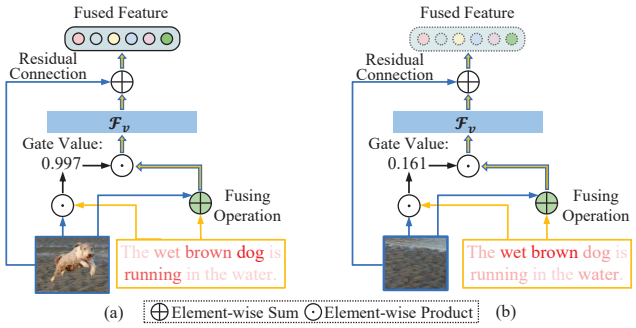


Figure 3. Illustration of the fusion between original region features and aggregated textual messages for the Cross-modal Gated Fusion module. (a) denotes the fusion of a positive region and textual message pair, and (b) denotes the fusion of a negative region and textual message pair.

gate values are assigned, and deeper fusion is encouraged. On the other hand, for negative pairs with low gate values, the fused information is suppressed by the gates, and thus $\hat{\mathbf{V}}$ is encouraged to keep the original features $\mathbf{V}$. Symmetrically, $\mathbf{T}$ and $\tilde{\mathbf{V}}$ can be fused to obtain $\hat{\mathbf{T}}$.

$$\mathbf{h}_i = \sigma(\tilde{\mathbf{v}}_i^\top \odot \mathbf{t}_i),\ i \in \{1, \cdots, N\}, \tag{7}$$

$$\mathbf{H}_t = [\mathbf{h}_1, \cdots, \mathbf{h}_N] \in \mathbb{R}^{d \times N}, \tag{8}$$

$$\hat{\mathbf{T}} = \mathcal{F}_t(\mathbf{H}_t \odot (\mathbf{T} \oplus \tilde{\mathbf{V}}^\top)) + \mathbf{T}. \tag{9}$$

3.3. Fused Feature Aggregation for Cross-modal Matching

We use a simple attention approach to aggregate the fused features of R regions and N words into feature vectors representing the whole image and the whole sentence. Specifically, given the fused features $\hat{\mathbf{V}} \in \mathbb{R}^{d \times R}$ and $\hat{\mathbf{T}} \in \mathbb{R}^{d \times N}$, the attention weight matrix is calculated by a linear projection and SoftMax normalization, and we aggregate the region features with the attention weights.

$$\mathbf{a}_v = \text{softmax}\Big(\frac{\mathbf{W}_v \hat{\mathbf{V}}}{\sqrt{d}}\Big)^\top, \qquad \mathbf{v}^* = \hat{\mathbf{V}}\mathbf{a}_v. \tag{10}$$

$$\mathbf{a}_t = \text{softmax}\Big(\frac{\mathbf{W}_t \hat{\mathbf{T}}}{\sqrt{d}}\Big)^\top, \qquad \mathbf{t}^* = \hat{\mathbf{T}}\mathbf{a}_t. \tag{11}$$

where $\mathbf{W}_v, \mathbf{W}_t \in \mathbb{R}^{1 \times d}$ denotes the linear projection parameters, and $\mathbf{a}_v \in \mathbb{R}^R$ denotes the attention weights for the fused feature of R regions, and $\mathbf{a}_t \in \mathbb{R}^N$ denotes the attention weights for the fused feature of N words. $\mathbf{v}^* \in \mathbb{R}^d$ is the aggregated features representation from $\hat{\mathbf{V}}$, and $\mathbf{t}^* \in \mathbb{R}^d$ is the aggregated features representation from $\hat{\mathbf{R}}$.

3.4. Infer Text-image Matching with Fused Features

Most previous approaches for text-image matching exploit visual-semantic embedding, which map the images and sentences into a common embedding space and calculates their similarities in the joint space [16, 5, 9, 34, 22]. Generally, consider the sampled positive image-sentence pair $(\mathcal{I}, \mathcal{C})$ and negative image-sentence pairs $(\mathcal{I}, \mathcal{C}')$, $(\mathcal{I}', \mathcal{C})$, the visual-semantic alignment is manipulated by the ranking loss with hardest negatives,

$$\mathcal{L}_{rank-h}(\mathcal{I}, \mathcal{C}) = \max_{\mathcal{C}'}[\alpha - m(\mathcal{I}, \mathcal{C}) + m(\mathcal{I}, \mathcal{C}')]_+ + \max_{\mathcal{I}'}[\alpha - m(\mathcal{I}, \mathcal{C}) + m(\mathcal{I}', \mathcal{C})]_+, \tag{12}$$

where $m(\mathcal{I}, \mathcal{C})$ denotes the matching score, which is calculated by the distance of features in the common embedding space. $[x]_+ = \max(0, x)$, α is the margin for ranking loss, and $\mathcal{C}'$ and $\mathcal{I}'$ are negative sentences and images, respectively.

With our proposed cross-modal Cross-modal Adaptive Message Passing model, however, the fused features can no longer be regarded as separate features in the same embedding space. Thus we cannot follow conventional visual-semantic embedding assumption to calculate the cross-modal similarities by feature distance in the joint embedding space. Instead, given the aggregated fused features v^* and s^*, we re-formulate the text-image matching as a classification problem (*i.e.* "match" or "mismatch") and propose a hardest negative cross-entropy loss for training. Specifically, we use a two-layer MLP followed by a sigmoid activation to calculate the final matching scores between images and sentences,

$$m(\mathcal{I}, \mathcal{C}) = \sigma(\text{MLP}(\mathbf{v}^* + \mathbf{t}^*)). \tag{13}$$

Although ranking loss has been proven effective for joint embedding, it does not perform well for our fused features. We exploit a hardest negative binary cross-entropy loss for training supervision.

$$\mathcal{L}_{BCE-h}(\mathcal{I}, \mathcal{C}) = \underbrace{\log(m(\mathcal{I}, \mathcal{C})) + \max_{\mathcal{C}'}[\log(1 - m(\mathcal{I}, \mathcal{C}'))]}_{\text{image-to-text matching loss}} + \underbrace{\log(m(\mathcal{I}, \mathcal{C})) + \max_{\mathcal{I}'}[\log(1 - m(\mathcal{I}', \mathcal{C}))]}_{\text{text-to-image matching loss}}, \tag{14}$$

where the first term is the image-to-text matching loss, and the second term is the text-to-image matching loss. We only calculate the loss of positive pairs and the hardest negative pairs in a mini-batch. Experiments in ablation study in Sec. 4.5 demonstrates the effectiveness of this loss.

In fact, projecting the comprehensive features from different modalities into the same embedding space is difficult for cross-modal embedding, and the complex interactions between different modalities cannot be easily described by a simple embedding. However, our problem formulation based on the fused features do not require the image and language features to be embedded in the same space, and thus encourages the model to capture more comprehensive and fine-grained interactions from images and sentences.

4. Experiments

4.1. Implementation Details

Word and region features. We describe how to extract the region-level visual features $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_R]$ and word-level sentence features $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_N]$.

We exploit the Faster R-CNN [26] with ResNet-101 to pretrained by Anderson *et al*. [1] to extract the top 36 region proposals for each image. A feature vector $\mathbf{m}_i \in \mathbb{R}^{2048}$ for each region proposal is calculated by average-pooling the spatial feature map. We obtain the 1024-dimentional region features with a linear projection layer,

$$\mathbf{v}_i = \mathbf{W_I}\mathbf{m}_i + \mathbf{b_I}, \tag{15}$$

where $\mathbf{W_I}$ and $\mathbf{b_I}$ are model parameters, and $\mathbf{v}_i$ is the visual feature for the ith region.

Given an input sentence with N words, we first embed each word to a 300-dimensional vector $x_i, i \in \{1, \cdots, N\}$ and then use a single-layer bidirectional GRU [3] with 1024-dimensional hidden states to process the whole sentence,

$$\overrightarrow{\mathbf{h}_i} = \overrightarrow{\mathrm{GRU}}(\overrightarrow{\mathbf{h}_{i-1}}, \mathbf{x}_i),\ \overleftarrow{\mathbf{h}_i} = \overleftarrow{\mathrm{GRU}}(\overleftarrow{\mathbf{h}_{i+1}}, \mathbf{x}_i). \tag{16}$$

The feature of each word is represented as the average of hidden states from the forward GRU and backward GRU,

$$\mathbf{t}_i = \frac{\overrightarrow{\mathbf{h}_i} + \overleftarrow{\mathbf{h}_i}}{2}, i \in \{1, \cdots, N\} \tag{17}$$

In practice, we set the maximum number of words in a sentences as 50. We clip the sentences which longer than the maximum length, and pad sentences with less than 50 words with a special padding token.

Training strategy. Adam optimizer is adopted for training. The learning rate is set to 0.0002 for the first 15 epochs and 0.00002 for the next 25 epochs. Early stopping based on the validation performance is used to choose the best model.

4.2. Experimental Settings

Datasets. We evaluate our approaches on two widely used text-image retrieval datasets, Flickr30K [31] and COCO [17]. Flickr30K dataset contains 31,783 images where each image has 5 unique corresponding sentences. Following [11, 5], we use 1,000 images for validation and 1,000 images for testing. COCO dataset contains 123,287 images, each with 5 annotated sentences. The widely used Karpathy split [11] contains 113,287 images for training, 5000 images for validation and 5000 images for testing. Following the most commonly used evaluation setting, we evaluate our model on both the 5 folds of 1K test images and the full 5K test images.

Evaluation Metrics. For text-image retrieval, the most commonly used evaluation metric is R@K, which is the abbreviation for recall at K and is defined as the proportion of correct matchings in top-k retrieved results. We adopt R@1, R@5 and R@10 as our evaluation metrics.

COCO 1K test images						
	Caption Retrieval			Image Retrieval		
Method	R@1	R@5	R@10	R@1	R@5	R@10
Order [27]	46.7	-	88.9	37.9	-	85.9
DPC [34]	65.6	89.8	95.5	47.1	79.9	90.0
VSE++ [5]	64.6	-	95.7	52.0	-	92.0
GXN [9]	68.5	-	97.9	56.6	-	94.5
SCO [10]	69.9	92.9	97.5	56.7	87.5	94.8
CMPM [33]	56.1	86.3	92.9	44.6	78.8	89.0
SCAN t-i [16]	67.5	92.9	97.6	53.0	85.4	92.9
SCAN i-t [16]	69.2	93.2	97.5	54.4	86.0	93.6
CAMP (ours)	**72.3**	**94.8**	**98.3**	**58.5**	**87.9**	**95.0**

COCO 5K test images						
	Caption Retrieval			Image Retrieval		
Method	R@1	R@5	R@10	R@1	R@5	R@10
Order [27]	23.3	-	84.7	31.7	-	74.6
DPC [34]	41.2	70.5	81.1	25.3	53.4	66.4
VSE++ [5]	41.3	-	81.2	30.3	-	72.4
GXN [9]	42.0	-	84.7	31.7	-	74.6
SCO [10]	42.8	72.3	83.0	33.1	62.9	75.5
CMPM [33]	31.1	60.7	73.9	22.9	50.2	63.8
SCAN i-t [16]	46.4	77.4	87.2	34.4	63.7	75.7
CAMP (ours)	**50.1**	**82.1**	**89.7**	**39.0**	**68.9**	**80.2**

Table 1. Results by CAMP and compared methods on COCO.

Flickr30K 1K test images						
	Caption Retrieval			Image Retrieval		
Method	R@1	R@5	R@10	R@1	R@5	R@10
VSE++ [5]	52.9	-	87.2	39.6	-	79.5
DAN [22]	55.0	81.8	89.0	39.4	69.2	79.1
DPC [34]	55.6	81.9	89.5	39.1	69.2	80.9
SCO [10]	55.5	82.0	89.3	41.1	70.5	80.1
CMPM [33]	49.6	76.8	86.1	37.3	65.7	75.5
SCAN t-i [16]	61.8	87.5	93.7	45.8	74.4	83.0
SCAN i-t [16]	67.7	88.9	94.0	44.0	74.2	82.6
CAMP (ours)	**68.1**	**89.7**	**95.2**	**51.5**	**77.1**	**85.3**

Table 2. Results by CAMP and compared methods on Flickr30K.

4.3. Quantitative Results

Table 1 presents our results compared with previous methods on 5k test images and 5 folds of 1k test images of COCO dataset, respectively. Table 2 shows the quantitative results on Flickr30k dataset of our approaches and previous methods. VSE++ [5] jointly embeds image features and sentence features into the same embedding space and calculates image-sentence similarities as distances of embedded features, and train the model with ranking loss with hardest negative samples in a mini-batch. SCAN [16] exploits stacked cross attention on either region features or word features, but does not consider message passing or fusion between image regions and words in sentences. Note that the best results of SCAN [16] employ an ensemble of two models. For fair comparisons, we only report their single model results on the two datasets.

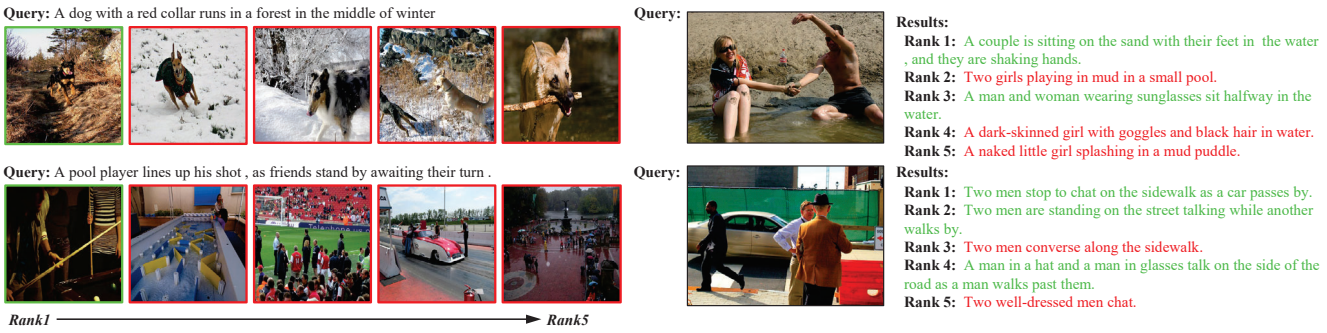


Figure 4. Qualitative retrieval results. The top-5 retrieved results are shown. Green denotes the ground-truth images or captions. Our model is able to capture the comprehensive and fine-grained alignments between images and captions by incorporating cross-modal interactions.

Ablation study results on Flickr30K						
	Caption Retrieval			Image Retrieval		
Method	R@1	R@5	R@10	R@1	R@5	R@10
CAMP	**68.1**	**89.7**	**95.2**	**51.5**	**77.1**	**85.3**
Base model	63.5	87.1	93.1	46.2	74.2	83.4
w/o cross-attn	59.7	83.5	88.9	41.2	65.5	79.1
w/o fusion	65.6	88.0	94.9	48.2	75.7	84.9
Fusion w/o gates	61.7	86.3	92.6	45.1	72.1	80.7
Fusion w/o residual	56.7	83.9	91.5	43.7	72.6	79.3
w/o attn-based agg	63.4	86.8	93.5	47.5	73.1	82.8
Concat fsuion	66.3	89.0	94.3	51.0	74.1	83.3
Product fusion	61.5	87.3	93.2	49.9	74.0	80.5
Joint embedding	62.0	87.8	92.4	46.3	73.7	80.3
MLP+Ranking loss	60.9	87.5	92.4	44.3	70.1	79.4
BCE w/o hardest	65.5	89.1	94.6	50.8	76.1	83.2

Table 3. Results of ablation studies on Flickr30K .

Experimental results show that our Cross-modal Adaptive Message Passing (CAMP) model outperforms previous approaches by large margins, demonstrating the effectiveness and necessity of exploring the interactions between visual and textual modalities for text-image retrieval.

4.4. Qualitative Results

We show qualitative results by our gated fusion model for text-to-image and image-to-text retrieval in Fig. 4. Take images in the first row of the left part as an example. We retrieve images based on the query caption "A dog with a red collar runs in a forest in the middle of winter." Our model successfully retrieves the ground-truth image. Note that the all of the top 5 retrieved images all related to the query caption, but the top 1 image matches better in details such as "runs in a forest" and "red collar". By alternatively attending to, passing messages and fusing between both modalities to incorporate deep cross-modal interactions, the model would have the potential of discovering such fine-grained alignments between images and captions.

4.5. Ablation Study

Our carefully designed Cross-modal Adaptive Message Passing model has shown superior performance, compared with conventional approaches that independently embed images and sentences to the joint embedding space without fusion. We carry several ablation experiments to validate the effectiveness of our design.

Base model without Cross-modal Adaptive Message Passing. To illustrate the effectiveness of our model, we design a baseline model without any cross-modal interactions. The baseline model attends to region features and word features separately to extract visual and textual features, and compare their similarities by cosine distance. The detailed structure is provided in the supplementary material. Ranking loss with hardest negatives is used as training supervision. The results are shown as "Base model" in Table 3, indicating that our CAMP model improves the base model without interaction by a large margin.

The effectiveness of cross-modal attention for Cross-modal Message Aggregation. In the Cross-modal Message Aggregation module, we aggregate messages to be passed to the other modality by cross-modal attention between two modalities. We experiment on removing the cross-modal attention and simply average the region or word features, and using the average word/region features as aggregated messages. Results are shown as "w/o cross-attn" in Table 3, indicating that removing the cross-modal attention for message aggregation would decrease the performance. We visualize some examples of cross-modal attention in the supplementary material.

The effectiveness of Cross-modal Gated Fusion. We implement a cross-modal attention model without fusion between modalities. The cross-modal attention follows the same way as we aggregate cross-modal messages for message passing in Sec. 3.1. Text-to-image attention and image-to-text attention are incorporated symmetrically. It has the potential to incorporate cross-modal interactions by attending to a modality with the cue from another modality, but no cross-modal fusion is adopted. The detailed structures are provided in the supplementary material. By comparing the performance of this model (denoted as "w/o fusion" in Table 3) with our CAMP model, we demonstrate that cross-modal fusion is effective in incorporating deeper cross-modal interactions. Additionally, the average gate values for positive and negative pairs are 0.971 and $2.7087 * 10^{-9}$, respectively, indicating that the adaptive gates are able to filter out the mismatched information and encourage fusion between aligned information.

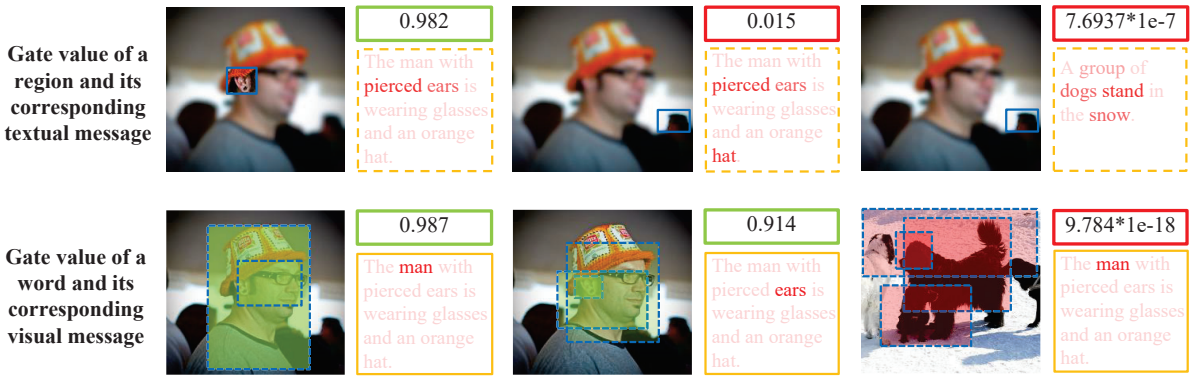


Figure 5. Gate values for aggregated textual/visual messages and original regions/words. High gate values indicate strong textual-visual alignments, encouraging deep cross-modal fusion. Low gate values suppress the fusion of uninformative regions or words for matching.

The necessity of adaptive gating and residual connection for Cross-modal Gated Fusion. We propose the adaptive gates to control to what extent the cross-modality information should be fused. Well-aligned features are intensively fused, while non-corresponding pairs are slightly fused. Moreover, there is a residual connection to encourage the model to preserve the original information if the gate values are low. We conduct experiments on fusion without adaptive gates or residual connection, denoted by "Fusion w/o gates" and "Fusion w/o residual" in Table 3. Also, to show the effectiveness of our choice among several fusion operations, two experiments denoted as "Concat fusion" and "Product fusion" are conducted to show the element-wise addition is slightly better. Results indicate that using a conventional fusion would confuse the model and cause a significant decline in performance. Moreover, we show some examples of gate values in Fig. 5. Words/regions that are strongly aligned to the image/sentence obtains high gate values, encouraging the fusing operation. While the low gate values would suppress the fusion of uninformative regions or words for matching. Note that the gate values between irrelevant background information may also be low even though the image matches with the sentence. In this way, the information from the irrelevant background is suppressed, and the informative regions are highlighted.

The effectiveness of attention-based fused feature aggregation. In Sec. 3.3, a simple multi-branch attention is adapted to aggregate the fused region/word-level features into a feature vector representing the whole image/sentence. We replace this attention-based fused feature aggregation with a simple average pooling along region/word dimension. Results denoted as "w/o attn-based agg" show the effectiveness of our attention-based fused feature aggregation.

Different choices for inferring text-image matching score and loss functions. Since the fused features cannot be regarded as image and sentence features embedded in the joint embedding space anymore, they should not be matched by feature distances. In Sec. 3.4, we reformulate the matching problem based on the fused features, by predicting the matching score with MLP on the fused features, and adopting hardest negative cross-entropy loss as training supervision. In the experiment denoted as "joint embedding" in Table 3, we follow conventional joint embedding approaches to calculate the matching score by cosine distance of the fused features $\hat{s}$ and $\hat{v}$, and employ the ranking loss (Eq.(12)) as training supervision. In the experiment denoted as "MLP+ranking loss", we use MLP on the fused features to predict the matching score, and adopt ranking loss for training supervision. We also test the effectiveness of introducing hardest negatives in a mini-batch for cross-entropy loss. In the experiment denoted as "BCE w/o hardest", we replace our hardest negative BCE loss with the conventional BCE loss without hardest negatives, where b is the number of negative pairs in a mini-batch, to balance the loss of positive pairs and negative pairs. Those experiments show the effectiveness of our scheme for predicting the matching score based on the fused features, and validates our hardest negative binary cross-entropy loss designed for training text-image retrieval.

5. Conclusion

Based on the observation that cross-modal interactions should be incorporated to benefit text-image retrieval, we introduce a novel Cross-modal Gated Fusion (CAMP) model to adaptively pass messages across textual and visual modalities. Our approach incorporates the comprehensive and fine-grained cross-modal interactions for text-image retrieval, and properly deals with negative (mismatched) pairs and irrelevant information with an adaptive gating scheme. We demonstrate the effectiveness of our approach by extensive experiments and analysis on benchmarks.

Acknowledgements This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14202217, CUHK14203118, CUHK14205615, CUHK14207814, CUHK14213616, CUHK14208417, CUHK14239816, in part by CUHK Direct Grant.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[2] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5659–5667, 2017.

[3] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[4] Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017.

[6] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015.

[7] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[8] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326, 2016.

[9] Jiuxiang Gu, Jianfei Cai, Shafiq Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7181–7189, 2018.

[10] Yan Huang, Qi Wu, and Liang Wang. Learning semantic concepts and order for image and sentence matching. *arXiv preprint arXiv:1712.02036*, 2017.

[11] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[12] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *arXiv preprint arXiv:1805.07932*, 2018.

[13] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016.

[14] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[15] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4437–4446, 2015.

[16] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. *arXiv preprint arXiv:1803.08024*, 2018.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[18] Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 338–354, 2018.

[19] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1950–1959, 2019.

[20] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, 2017.

[21] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[22] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. *arXiv preprint arXiv:1611.00471*, 2016.

[23] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. *arXiv preprint arXiv:1804.00775*, 2018.

[24] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1899–1907. IEEE, 2017.

[25] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. *arXiv preprint arXiv:1711.08389*, 2017.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[27] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[28] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[29] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[30] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[31] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[32] Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. Multi-level attention networks for visual question answering. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4187–4195. IEEE, 2017.

[33] Ying Zhang and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In *ECCV*, 2018.

[34] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. Dual-path convolutional image-text embedding. *arXiv preprint arXiv:1711.05535*, 2017.

[35] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. Structured attentions for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017.

Context-Aware Attention Network for Image-Text Retrieval

Qi Zhang[1,2] Zhen Lei[1,2*] Zhaoxiang Zhang [1,2] Stan Z. Li[3]
[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Center for AI Research and Innovation, Westlake University, Hangzhou, China.
{qi.zhang,zhaoxiang.zhang}@ia.ac.cn, {zlei,szli}@nlpr.ia.ac.cn

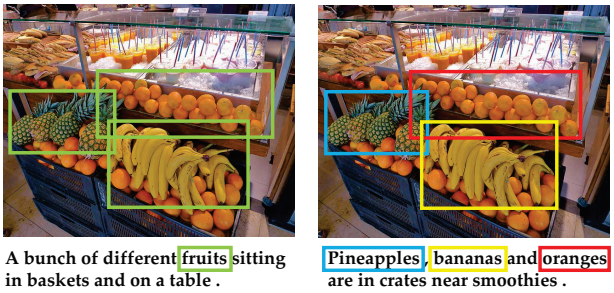
Abstract

As a typical cross-modal problem, image-text bi-directional retrieval relies heavily on the joint embedding learning and similarity measure for each image-text pair. It remains challenging because prior works seldom explore semantic correspondences between modalities and semantic correlations in a single modality at the same time. In this work, we propose a unified Context-Aware Attention Network (CAAN), which selectively focuses on critical local fragments (regions and words) by aggregating the global context. Specifically, it simultaneously utilizes global inter-modal alignments and intra-modal correlations to discover latent semantic relations. Considering the interactions between images and sentences in the retrieval process, intra-modal correlations are derived from the second-order attention of region-word alignments instead of intuitively comparing the distance between original features. Our method achieves fairly competitive results on two generic image-text retrieval datasets Flickr30K and MS-COCO.

1. Introduction

Associating vision with language and exploring the relations between them have attracted great interest in the past decades. Many tasks have efficiently combined these two modalities and made significant progress, such as visual question answering (VQA) [1, 2, 33, 25], image captioning [1, 9], and person search with natural language [22, 23]. Image-text bidirectional retrieval [40, 44] is one of the most popular branches in the field of cross-modal research. It aims to retrieve images given descriptions or find sentences from image queries. Due to the large discrepancy between these two modalities, the main challenge is how to learn joint embeddings and accurately measure the image-text similarity.

While describing a target image, people tend to make frequent references to salient objects and depict their attributes and actions. Based on the observation, some approaches [15, 16, 33] map regions in images and words in sentences into a latent space and explore alignments between them. Although validating the efficacy of exploring region-word correspondences, they ignore the different importance of each local fragment. Recently, attention-based methods [19, 20, 26, 41] have taken steps toward attending differently to the specific regions and words and shown very promising results in the image-text retrieval task. SCAN [19] is a typical one to decide the importance of fragments based on fragments from another modality, aiming to discover full region-word alignments. Nevertheless, it ignores semantic correlations (common or exclusive attributes, categories, scenes *etc*.) between fragments in a single modality. Furthermore, some works [20, 26] have been proposed to either learn visual relation features with pre-trained neural scene graph generators or eliminate irrelevant fragments based on intra-modal relations, which alleviate the problems mentioned above to some extent.



(a) (b)

Figure 1. Illustration of the adaptive retrieval process with different contexts. An image is annotated with two different sentences. The regions highlighted with green in (a) correspond to "fruits" in the left sentence. However, they correspond to "pineapples", "bananas" and "oranges" in the right sentence, highlighted with blue, yellow and red in (b), respectively.

*Corresponding author.

2575-7075/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPR42600.2020.00359

However, most previous attention-based methods [19, 20, 26, 41] ignore the fact that a word or region might have different semantics in different global contexts. Specifically, the global context refers to both interaction and alignments between two modalities (inter-modal context) and semantic summaries and correlations in a single modality (intra-modal context). As shown in Figure 1, people sometimes automatically summarize high-level semantic concepts (such as fruits) based on the relationships between objects in Figure 1(a), and sometimes describe each object separately (such as pineapple, banana, orange) in Figure 1(b). Therefore, it is beneficial to take into account intra-modal and inter-modal contexts simultaneously and perform image-text bidirectional retrieval with adaptation to various contexts.

To address the problems above, we first propose a unified Context-Aware Attention Network (CAAN) to selectively attend to local fragments based on the global context. It formulates the image-text retrieval as an attention process, which integrates both the inter-modal attention to discover all possible alignments between word-region pairs and intra-modal attention to learn semantic correlations of fragments in a single modality. By exploiting the context-aware attention, our model can simultaneously perform image-assisted textual attention and text-assisted visual attention. As a result, the attention scores assigned for fragments aggregate the context information.

Instead of intuitively using feature-based similarities, we further propose Semantics-based Attention (SA) to explore latent intra-modal correlations. Our semantics-based attention is formulated as the second-order attention of region-word alignments, which explicitly considers interactions between modalities and effectively utilizes region-word relations to infer the semantic correlations in a single modality. It is aware of the current input pair, and the comprehensive context from the image-text pair can directly influence the computation of each other's responses in the retrieval process. Therefore, it achieves the actual adaptive matching according to the given context.

In summary, the main contributions of our work are listed as follow:

- We propose a unified Context-Aware Attention Network to adaptively select informative fragments based on the given context from a global perspective, including semantic correlations in a single modality and possible alignments between region and words.
- We propose the Semantics-based Attention to capture latent intra-modal correlations. It is the interpretable second-order attention of region-word alignments.
- We evaluate our proposed model on two benchmark datasets Flickr30K [46] and MS-COCO [24] and it achieves fairly competitive results.

2. Related Work

Most existing methods for image-text retrieval either embed whole images and full sentences into a shared space or consider latent correspondences between local fragments. Some recent approaches further adopt the attention mechanism to focus on the most important local fragments.

2.1. Image-Text Retrieval

Global embeddings based methods. A common solution is to learn joint embeddings for images and sentences. DeViSE [10] made the first attempt to unify image features and skip-gram word features by a linear mapping. Wang *et al*. [39] combined the bi-directional ranking constraints with neighborhood structure preservation constraints in a single modality. Li *et al*. [22] used identity-level annotations and a two-stage framework to learn better feature representations. More recent works focus on the design of objective functions. Zheng *et al*. [47] learned the dual-path convolutional image-text embeddings with the proposed instance loss.

Although these methods have achieved a certain degree of success, image-text retrieval remains challenging due to a lack of detailed understanding of the fine-grained interplay between images and sentences.

Local fragments based methods. Different from the methods above, many efforts have been devoted to addressing the problem of image-text retrieval on top of local fragments. DVSA [16] first adopted R-CNN to detect salient objects and inferred latent alignments between words in sentences and regions in images. Ma *et al*. [30] proposed to learn relations between images and fragments composed from words at different levels. sm-LSTM [13] attempted to jointly predict instance-aware saliency maps for both images and sentences and use their similarities within several timesteps. HM-LSTM [33] exploited hierarchical relations between sentences and phrases, and between whole images and image regions, to jointly establish their representations. Huang *et al*. [14] proposed a semantic-enhanced image and sentence matching model, which learns semantic concepts and organizes them in a correct semantic order.

In this paper, we adopt the same local fragments based strategy to consider the contents of images and text at a finer level instead of using a rough overview.

2.2. Attention Mechanism

Attention mechanism recently has gained popularity and been applied to various applications, including image classification [31, 38], image captioning [29, 43] and question answering [36, 42, 45]. Benefiting from its great power, many attention-based methods have been proposed in the image-text retrieval task. DAN [32] introduced Dual Attention Networks to attend to specific regions in images and words in text through multiple steps. SCAN [19] used

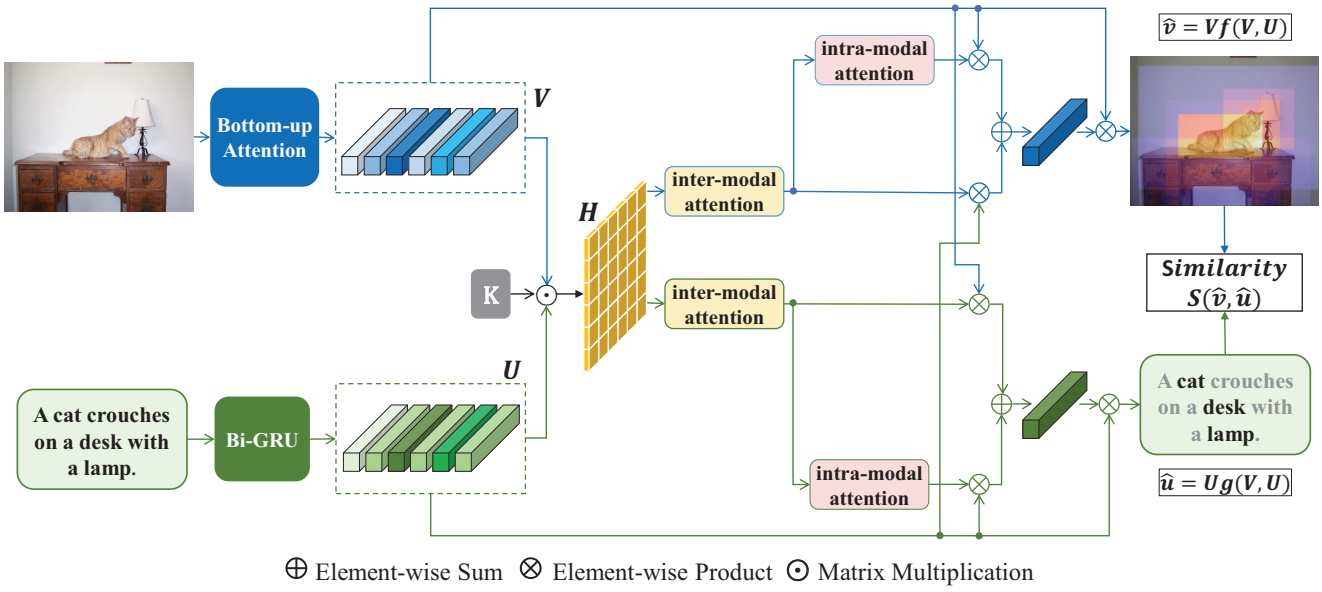


Figure 2. The pipeline of our proposed context-aware attention network (CAAN). It consists of three modules, (a) extracting and encoding regions in images and words in sentences, (b) context-aware attention with adaptation to the dynamic global context and (c) joint optimization of the final representations with the bi-directional ranking loss.

Stacked Cross Attention to perform either image-to-text attention or text-to-image attention at a time. CAMP [41] proposed Cross-Modal Adaptive Message Passing to attend to fragments. Considering visual relations between regions, recent approach [20] adopted cross-modal attention and learned visual relation features with pretrained neural scene graph generators.

In addition to methods above, there some recent methods extend the popular BERT [5] architecture to jointly learn visual and textual representations. These methods [21, 4, 28] either use a single-stream model to fuse textual and visual data as input, or take a two-stream model to process each modality separately and then fuse them. Benefiting from the self-attention module of BERT, they have achieved the state-of-the art performance.

3. Method

In this section, we will present an overview of our proposed Context-Aware Attention Network (CAAN). As shown in Figure 2, given an image-text pair, we first embed regions in images and words in sentences into a shared space. Concretely, the bottom-up attention [1] is utilized to generate image regions and their representations. Meanwhile, we encode words in sentences along with the sentence context. In the association module, we perform our context-aware attention network on the extracted features of local fragments, which captures semantic alignments between region-word pairs and semantic correlations between fragments in a single modality. Finally, the model is trained with image-text matching loss.

Next, we will introduce details of our proposed method from the following aspects: 1) visual representations, 2) textual representations, 3) context-aware attention network for global context aggregation, 4) objective function to optimize image-text retrieval.

3.1. Visual Representations

Given an image, we observe that people tend to make frequent references to salient objects and describe their actions and attributes, *etc*. Instead of extracting the global CNN feature from a pixel-level image, we focus on local regions and take advantage of bottom-up attention [1]. Following [1, 19, 20], we detect objects and other salient regions in each image utilizing a Faster R-CNN [34] model in conjunction with ResNet-101 [12] in two stages, which is pretrained on Visual Genome [18]. In the first stage, the model uses greedy non-maximum suppression with an IoU threshold to select the top-ranked box proposals. In the second stage, the extracted features of those bounding boxes are obtained after the mean-pooled convolutional layer. The features are used to predict both instance and attribute classes, in addition to refining bounding boxes. For each region i, x_i denotes the original mean-pooling convolutional feature with 2048 dimensions. The final feature v_i is transformed by a linear mapping of x_i into a D-dimensional vector as follows:

$$v_i = W_x x_i + b_i. \tag{1}$$

Therefore, the target image v can be presented as a set of features of selected ROIs with the highest class detection confidence scores.

3.2. Textual Representations

In order to discover region-word correspondences, words in sentences are mapped into the same D-dimensional space as image regions. Instead of processing each word individually, we consider to encode the word and its context at a time. Given one-hot encodings $W = \{w_1, ..., w_m\}$ of m input words in a sentence, we first embed them into 300-dimensional vectors by the word embedding layer as $x_i = W_e w_i$, where W_e is a parametric matrix learned end-to-end. We then feed vectors into a bi-directional GRU [3, 35], which is written as:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(x_i, \overrightarrow{h_{i-1}}), i \in [1, m], \tag{2}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(x_i, \overleftarrow{h_{i+1}}), i \in [1, m], \tag{3}$$

where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ denote hidden states from the forward and backward directions, respectively. The final word embedding u_i is the mean of bi-directional hidden states, which collects the context centered in the word w_i:

$$u_i = \frac{\overrightarrow{h_i} + \overleftarrow{h_i}}{2}, i \in [1, m]. \tag{4}$$

3.3. Context-aware Attention

3.3.1 Formulation

The attention mechanism aims to focus on the most pertinent information of the corresponding task rather than using all available information equally. We first provide a general formulation of attention mechanism designed for the cross-modal retrieval problem. For image v and text u, their feature maps are formulated as $V = [v_1, ..., v_n]$ and $U = [u_1, ..., u_m]$, respectively. We define the attention process for image-text retrieval as:

$$\hat{v} = Vf(V, U) = \sum_{i=1}^{n} [f(V, U)]_i v_i, \tag{5}$$

$$\hat{u} = Ug(V, U) = \sum_{j=1}^{m} [g(V, U)]_j u_j, \tag{6}$$

where $f(\cdot)$ and $g(\cdot)$ are attention functions to calculate scores for each local fragment v_i and u_j, respectively. The final image and text features $\hat{v}$ and $\hat{u}$ are computed as the weighted sum of local fragments. Following [**?**, 29], we calculate similarities between region-word pairs for the target image and text. The similarity matrix H is written as:

$$H = \tanh(V^T K U), \tag{7}$$

where $K \in \mathbb{R}^{d \times d}$ is a weight matrix. Attentive Pooling Networks [6] performs column-wise and row-wise max-pooling based on the assumption that the importance of each fragment is represented as its maximal similarity over fragments of another modality. It is an alternative version of the proposed attention process when $f(V, U)$ becomes the softmax computation after applying row-wise max-pooling operation on H. Furthermore, we not only calculate the similarity matrix but use it as a feature to predict the attention map. To be more specific, the importance score of a fragment is decided by all the relevant fragments, taking into account intra-modal correlations in a single modality and inter-modal alignments between all region-word pairs. Based on the consideration, the normalized attention function $f(V, U)$ for regions can be formulated as follows:

$$\widetilde{f}(V, U) = \tanh(H^v V^T Q_1 + H^{uv} U^T Q_2), \tag{8}$$

$$f(V, U) = \text{softmax}(W^v \widetilde{f}(V, U)), \tag{9}$$

where $W^v \in \mathbb{R}^z$ is a projection vector. $Q_1, Q_2 \in \mathbb{R}^{d \times z}$ are parametric matrices to do dimension-wise fusion. $H^v \in \mathbb{R}^{n \times n}$ is the attention matrix capturing intra-modal correlations for regions. $H^{uv} \in \mathbb{R}^{n \times m}$ is the attention matrix for word-to-region re-weighting. Likewise, the normalized attention function $g(V, U)$ for words is written as follows:

$$\widetilde{g}(V, U) = \tanh(H^u U^T Q_3 + H^{vu} V^T Q_4), \tag{10}$$

$$g(V, U) = \text{softmax}(W^u \widetilde{g}(V, U)), \tag{11}$$

where $Q_3, Q_4 \in \mathbb{R}^{d \times z}$ and $W^u \in \mathbb{R}^z$ are learned weights.

The designed attention functions $f(V, U)$ and $g(V, U)$ selectively attend to those informative fragments according to the global context, applying both inter-modal attention and intra-modal attention.

3.3.2 Inter-modal Attention: H^{uv}, H^{vu}

The matrix H calculates similarities of local region-word pairs. Following [15, 19, 20], we threshold the similarities to zero and normalize them to obtain alignment scores. The word-to-region attention H^{uv} is computed as:

$$H_{ij}^{uv} = \frac{[H_{ij}]_+}{\sqrt{\sum_{k=1}^{n} [H_{kj}]_+^2}}, \tag{12}$$

where $[x]_+ \equiv max(0, x)$. Each element $H_{i,j}^{uv}$ in the word-to-region attention matrix H^{uv} represents the relative pairwise correspondences of two local fragments region v_i and word u_j. Similarity, the region-to-word attention H^{vu} is computed as:

$$H_{ij}^{vu} = \frac{[H_{ij}]_+}{\sqrt{\sum_{k=1}^{m} [H_{ik}]_+^2}}, \tag{13}$$

Both H^{uv} and H^{vu} infer fine-grained interplay between images and sentences by aligning regions and words.

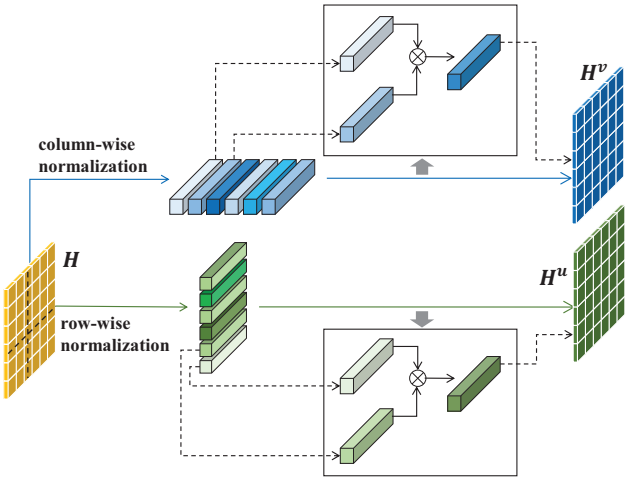


Figure 3. Detailed illustration of the semantics-based intra-modal attention process. The intra-modal affinity matrices H^v and H^u are designed to capture latent region-to-region and word-to-word relations, respectively. They are calculated by fully utilizing the inter-modal alignments.

3.3.3 Intra-modal Attention: H^v, H^u

Next, we will discuss two versions of H^v and H^u, which model intra-modal correlations from two different perspectives.

Feature-based attention (FA). A natural choice of measuring intra-modal correlations is to calculate feature similarities. That is, the intra-modal attention matrices H^v and H^u can be defined as:

$$H^v = V^T M_1 V, \tag{14}$$

$$H^u = U^T M_2 U, \tag{15}$$

where $M_1, M_2 \in \mathbb{R}^{d\times d}$ are learned weight parameters. When they are equal to identity matrices, elements in H^v and H^u denote dot-product similarities between local fragments in a single modality. The matrix product of a learned matrix and its transpose is another alternative version, which projects U into a new space. It not only allows the calculated intra-modal attention matrices to represent the cosine similarities between normalized features, but also preserves the model capacity.

However, it ignores that the semantic summary (intra-modal context) in one modality varies for different queries. Therefore, the semantic correlation mining between fragments in a single modality should be conducted in an interactive way.

Semantics-based attention (SA). Considering the interactions and message passing across two modalities in the retrieval process, we propose the semantics-based attention to explore intra-modal correlations based on region-word relations. In our work, we use the interpretable second-order attention of inter-modal alignments. The detailed procedure of SA is illustrated in Figure 3. The intra-modal attention matrices H^v and H^u are defined as:

$$H^v = \begin{bmatrix} \mathrm{norm}(H^{uv}_{1\cdot}) \\ \mathrm{norm}(H^{uv}_{2\cdot}) \\ \vdots \\ \mathrm{norm}(H^{uv}_{n\cdot}) \end{bmatrix} \begin{bmatrix} \mathrm{norm}(H^{uv}_{1\cdot}) \\ \mathrm{norm}(H^{uv}_{2\cdot}) \\ \vdots \\ \mathrm{norm}(H^{uv}_{n\cdot}) \end{bmatrix}^T, \tag{16}$$

$$H^u = \begin{bmatrix} \mathrm{norm}(H^{vu}_{1\cdot}) \\ \mathrm{norm}(H^{vu}_{2\cdot}) \\ \vdots \\ \mathrm{norm}(H^{vu}_{m\cdot}) \end{bmatrix} \begin{bmatrix} \mathrm{norm}(H^{vu}_{1\cdot}) \\ \mathrm{norm}(H^{vu}_{2\cdot}) \\ \vdots \\ \mathrm{norm}(H^{vu}_{m\cdot}) \end{bmatrix}^T, \tag{17}$$

where $\mathrm{norm}(\cdot)$ means the l_2-normalized operation on the input vector. As the i-th row of the inter-modal attention matrix H^{uv}, $H^{uv}_{i\cdot}$ is considered to be the word-to-region affinity distribution or response vector for all words with respect to the given v_i. It measures the distance between v_i and the entire word features set $\{u_1, ..., u_m\}$. Therefore, each element H^v_{ij} is the cosine similarity of two region-word response vectors $H^{uv}_{i\cdot}$ and $H^{uv}_{j\cdot}$. The intra-modal attention matrix H^v calculates pairwise relations of any two affinity distributions.

The intra-modal summaries and correlations are related to the global context in the retrieval process, and they implicitly contain both statistics and semantic information, $i.e.$ co-existence, dependencies and affiliation. When two regions v_i and v_j have similar responses to the same sentence, they are viewed as a high-correlated pair. Accordingly, SA focuses more on region v_i in the process of assigning attention scores with respect to region v_j. It comprehensively takes into account the similarity of two responses, which models the relationship between the movement of similarities of fragments between two modalities.

To summarize, the adaptive intra-modal attention process is driven by the global semantic information. It requires discrimination on semantics based on the given context rather than original context-free features.

3.4. Objective Function

The hinge-based bi-directional ranking loss [8, 16, 19] is the most popular objective function for image-text retrieval, which can be formulated as follows:

$$\begin{aligned} L(\hat{v}, \hat{u}) = \sum_{\hat{v}^-, \hat{u}^-} \{ & max[0, m - S(\hat{v}, \hat{u}) + S(\hat{v}, \hat{u}^-)] \\ & + max[0, m - S(\hat{v}, \hat{u}) + S(\hat{v}^-, \hat{u})]\}, \end{aligned} \tag{18}$$

where m is a margin constraint, $(\hat{v}, \hat{u}^-)$ and $(\hat{v}^-, \hat{u})$ are negative pairs. $S(\cdot)$ is a matching function, which is defined as the inner product in our experiments. The objective function attempts to pull positive image-text pairs close

Methods	MS-COCO 5-fold 1K Test Images						Flickr30K 1K Test Images					
	Sentence Retrieval			Image Retrieval			Sentence Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
(R-CNN, AlexNet)												
DVSA [16]	38.4	69.9	80.5	27.4	60.2	74.8	22.2	48.2	61.4	15.2	37.7	50.5
(VGG)												
VQA-A [25]	50.5	80.1	89.7	37.0	70.9	82.9	33.9	62.5	74.5	24.9	52.6	64.8
sm-LSTM [13]	53.2	83.1	91.5	40.7	75.8	87.4	42.5	71.9	81.5	30.2	60.4	72.3
2WayNet [7]	55.8	75.2	-	39.7	63.3	-	49.8	67.5	-	36.0	55.6	-
(ResNet)												
RRF-Net [27]	56.4	85.3	91.5	43.9	78.1	88.6	47.6	77.4	87.1	35.4	68.3	79.9
VSE++ [8]	64.6	90.0	95.7	52.0	84.3	92.0	52.9	80.5	87.2	39.6	70.1	79.5
DAN [32]	-	-	-	-	-	-	55.0	81.8	89.0	39.4	69.2	79.1
DPC [47]	65.6	89.8	95.5	47.1	79.9	90.0	55.6	81.9	89.5	39.1	69.2	80.9
GXN [11]	68.5	-	97.9	56.6	-	94.5	56.8	-	89.6	41.5	-	80.
SCO [14]	69.9	92.9	97.5	56.7	87.5	94.8	55.5	82.0	89.3	41.1	70.5	81.1
(Faster-RCNN, ResNet)												
SCAN-single [19]	70.9	94.5	97.8	56.4	87.0	94.8	67.9	89.0	94.4	43.9	74.2	82.8
R-SCAN [20]	70.3	94.5	98.1	57.6	87.3	93.7	66.3	90.6	96.0	51.4	77.8	84.9
CAMP [41]	72.3	94.8	98.3	58.5	87.9	95.0	68.1	89.7	95.2	51.5	77.1	85.3
BFAN-single [26]	73.7	94.9	-	58.3	87.5	-	64.5	89.7	-	48.8	77.3	-
CAAN (ours)	**75.5**	**95.4**	**98.5**	**61.3**	**89.7**	**95.2**	**70.1**	**91.6**	**97.2**	**52.8**	**79.0**	**87.9**

Table 1. Results of the cross-modal retrieval on MS-COCO 5-fold 1K test set and Flickr30K 1K test setre. The best performance is denoted with bold text. '-': the result is not provided.

and push negative ones away. Despite widely used in the cross-modal task, it suffers from high redundancy and slow convergence caused by the random triplet sampling process. Rather than summing over all the negative pairs in a mini-batch, bi-directional ranking loss with the hardest negatives is often adopted for computational efficiency. It focuses on the hardest samples which are the negative ones closest to positive pairs. Given a positive pair $(\hat{v}, \hat{u})$, the hardest negatives are formulated as $v_h = \arg\max_{p \neq \hat{v}} S(p, \hat{u})$ and $u_h = \arg\max_{k \neq \hat{u}} S(\hat{v}, k)$. Therefore, the bi-directional ranking loss with the hardest negatives is written as:

$$\begin{aligned} L_{hard}(\hat{v}, \hat{u}) = max[0, m - S(\hat{v}, \hat{u}) + S(\hat{v}, \hat{u}_h^-)] \\ + max[0, m - S(\hat{v}, \hat{u}) + S(\hat{v}_h, \hat{u})]. \end{aligned} \tag{19}$$

4. Experiments

4.1. Datasets and Metrics

Datasets. We evaluate our model on the Flickr30K [46] and MS-COCO [24] datasets. Flickr30K contains 31,000 images and each image is associated with five sentences. We adopt the same protocol in [8, 16] to split the dataset into 1,000 test images, 1,000 validation images, and 29,000 training images. MS-COCO contains 123,287 images and each is annotated with five descriptions. In [16], MS-COCO is split into 82,783 training images, 5000 validation images and 5,000 test images. We follow [8, 19] to use other 30,504 images as part of the training set, which were originally in the validation set but have been left out in the split.The experiments are conducted on both 5K and 1K test images, where the result of 1K test images is reported by averaging over 5-fold on the full 5K test images.

Evaluation Metrics. We use R@K and mR to evaluate our models. R@K is the percentage of correct matchings in the top-K lists. R@1, R@5 and R@10 are adopted in the experiments. mR is the mean value of R@K (K=1,5,10).

4.2. Implementation Details

The Adam optimizer [17] is employed for optimization. In the MS-COCO, we set the initial learning rate to 0.0005 for the first 10 epochs and then decay it by 10 times in the following 10 epochs. In the Flickr30K, the learning rate is 0.0002 in the first 15 epochs, and reduced to 0.00002 in the next 15 epochs. The best model is chosen based on the sum of recalls on the validation set.

4.3. Quantitative Results

4.3.1 Comparisons with non-BERT Methods

We compare our model with several recent state-of-the-art non-BERT methods on the MS-COCO and Flickr30K datasets. As shown in Table 1, CAAN outperforms other methods by a large margin. For fair comparisons, we only report single model results of SCAN [19] and BFAN [26]

Methods	MS-COCO 5K Test Images					
	Sentence Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
(R-CNN, AlexNet)						
DVSA [16]	16.5	39.2	52.0	10.7	29.6	42.2
(VGG)						
VQA-A [25]	23.5	50.7	63.6	16.7	40.5	53.8
(ResNet)						
VSE++ [8]	41.3	69.2	81.2	30.3	59.1	72.4
GXN [11]	42.0	-	84.7	31.7	-	74.6
SCO [14]	42.8	72.3	83.0	33.1	62.9	75.5
(Faster-RCNN, ResNet)						
PVSE [37]	45.2	74.3	84.5	32.4	63.0	75.0
SCAN-single [19]	46.4	77.4	87.2	34.4	63.7	75.7
R-SCAN [20]	45.4	77.9	87.9	36.2	65.6	76.7
CAMP [41]	50.1	82.1	89.7	39.0	68.9	80.2
CAAN (ours)	**52.5**	**83.3**	**90.9**	**41.2**	**70.3**	**82.9**

Table 2. Comparisons of the cross-modal retrieval results on the MS-COCO full 5K test set.

on the two datasets rather than using the ensemble version. On the 1K test set, CAAN gives R@10=98.5 and 95.2 with image and text as queries, respectively. It achieves the performance with R@1=61.3 for image retrieval, which is a 3% relative improvement compared to the current state-of-the-art non-BERT methods, $i.e.$, BFAN-single [26]. On the Flickr30K dataset, CAAN achieves better R@1 at 70.1 and 52.8 with sentence and image retrieval, respectively. The results on the MS-COCO 5K test set are summarized in Table 2. CAAN significantly outperforms the current non-BERT methods on all metrics, which verifies the effectiveness of our proposed method. As illustrated in the section 3.3, our introduced attention process explores both region-word alignments and semantic correlations in a single modality. The performance gain compared with other non-BERT methods demonstrates the superior to consider the specific context in the adaptive retrieval process.

	Sentence Retrieval			Image Retrieval		
	R@1	R@10	mR	R@1	R@10	mR
ViLBERT†[28]	-	-	-	45.5	85.0	69.1
UNITER†[4]	-	-	83.3	-	-	73.9
ViLBERT‡[28]	-	-	-	58.2	91.5	78.2
Unicoder-VL†[21]	73.0	94.1	85.4	57.8	88.9	76.3
CAAN (ours)	70.1	97.2	86.3	52.8	87.9	73.2
UNITER‡[4]	-	-	92.2	-	-	83.1
Unicoder-VL‡[21]	**86.2**	**99.0**	**93.8**	**71.5**	**94.9**	**85.8**

Table 3. Comparisons with BERT-based methods on the Flickr30k dataset. CAAN (ours) is the baseline model, which uses Faster R-CNN pre-trained on Visual Genome, without pre-training the language model. † indicates methods using both pre-trained visual features and language model (BERT) initialization with text-only data. ‡ indicates methods pre-trained with extra out-of-domain (Vision-Language) data.

4.3.2 Comparisons with BERT-based Methods

We additionally make comparisons with other BERT-based methods, which achieve the state-of-art performance on the Flickr30K and MS-COCO datasets. As shown in Table 3, our method has fairly comparable results compared with the BERT-based methods, even without introducing and fine-tuning on a pre-trained language model.

Besides, our method is much faster and smaller, compared to BERT-based ones. Taking ViLBERT as an example, computing similarity between a text-image pair takes around 0.5 s, while ours is around 45 μs, using 1 GTX1080Ti. ViLBERT has parameters of 275 M, while ours is only 11 M. Considering the speed and model size requirements of the real-world scenes, our method is more convenient and practical for deployment and application.

	Image Query		Sentence Query	
	R@1	R@10	R@1	R@10
baseline	58.1	90.0	42.0	79.7
baseline+IA	60.6	92.4	45.2	81.5
baseline+FA	62.3	93.2	46.6	83.0
baseline+SA	64.5	93.8	48.8	83.4
baseline+IA+FA	62.6	93.0	45.0	82.9
CAAN	**70.1**	**97.2**	**52.8**	**87.9**

Table 4. Results of ablation studies on the Flickr30K test set.

4.4. Ablation Studies on Attention Mechanism

In this section, we perform ablation studies to quantify the effect of our proposed attention mechanism, including intra-modal and inter-modal attention. We first provide the baseline model with bottom-up attention [1], denoted as "baseline" in Table 4. It takes the average of all local features as final representations. We can see that it achieves a fairly competitive result compared to the methods extracting global features shown in Table 1. It shows the reasonability to focus on local fragments rather than using a rough overview of a whole image or a full sentence.

Baseline with Inter-modal Attention. We implement inter-modal attention in the baseline model, denoted as "baseline+IA" in Table 4. It achieves R@1=60.6 and 45.2 with image and text as queries, respectively. Compared with "baseline", CAAN demonstrates its effectiveness of considering full alignments between region-word pairs.

Baseline with Intra-modal Attention. Table 4 illustrates the impact of preforming intra-modal attention. Both "baseline+FA" and "baseline+SA" use only relations of fragments in a single modality. The difference between them is the way to measure fragment affinities. Although "baseline+FA" introduces additional parameters M_1 and M_2 to fit data, "baseline+SA" still achieves better results, which shows the superior of inferring semantic correlations by

(a) Inter-modal Alignments (b) Intra-modal Correlations

Figure 4. Visualization of the attention weights of each image region with respect to sentence query on MS-COCO and Flickr30K datasets. The left sub-figure (a) shows the qualitative examples of text-to-image retrieval with different sentences. The right sub-figure (b) compares "baseline+IA+FA" and our CAAN, which shows that the similar semantics shared by different objects affect the attention process. It is beneficial to consider both inter-modal alignments and intra-modal correlations in an interactive way. (Best viewed in color)

adaptively measuring the distance of response vectors instead of original features.

Baseline with both Inter-modal and Intra-modal Attention. We further integrate inter-modal and intra-modal attention into the baseline modal. Results are denoted as "baseline+IA+FA" and "CAAN" shown in Table 4. "baseline+IA+FA" even has a slightly worse result compared to "baseline+FA". It shows that without careful designs, combing inter-modal alignments and intra-modal correlations might hurt the performance. While "CAAN" outperforms "baseline+IA+FA" and "baseline+SA", indicating that it is a better solution to consider the global context and conduct semantic correlation mining in an interactive way.

5. Visualization

To better understand the effectiveness of our proposed model, we visualize the attention assignment of the text-to-image retrieval process in Figure 4. For the qualitative examples in Figure 4(a), we can observe that attention weights are assigned to different regions for different image-text pairs. As shown in the first row of Figure 4(a), the region "bottle" receives more attention in the left sub-figure while the region "bags" is the focus in the right sub-figure. It indicates that our model infers inter-modal alignments based on the global context. For the qualitative examples in Figure 4(b), we provide comparisons with "baseline+FA+IA". As shown in the second row of Figure 4(b), the region "boy" is assigned more attention weight with the proposed CAAN compared with the model "baseline+IA+FA". It is notable that different objects with similar semantics affect the matching process.

6. Conclusion

In this paper, we propose a unified Context-Aware Attention Network (CAAN) to formulate the image-text retrieval as an attention process to selectively focus on the most informative local fragments. By incorporating intra-modal and inter-modal attention, our model aggregates the context information of alignments between word-region pairs (inter-modal context) and semantic correlations between fragments in a single modality (intra-modal context). Furthermore, we perform the semantic-based attention to model intra-modal correlations, which is the interpretable second-order attention of region-word alignments. The model demonstrates its effectiveness by achieving fairly competitive results on the Flickr30K and MS-COCO datasets.

7. Acknowledgements

This work has been partially supported by the Chinese National Natural Science Foundation Projects #61872367, #61876178, #61806196, #61806203, #61976229.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *ICCV*, 2015.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[6] Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*, 2016.

[7] Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. In *CVPR*, 2017.

[8] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017.

[9] Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *CVPR*, 2015.

[10] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013.

[11] Jiuxiang Gu, Jianfei Cai, Shafiq R. Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal LSTM. In *CVPR*, 2017.

[14] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *CVPR*, 2018.

[15] Andrej Karpathy, Armand Joulin, and Fei-Fei Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NeurIPS*, 2014.

[16] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[19] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, 2018.

[20] Kuang-Huei Lee, Hamid Palangi, Xi Chen, Houdong Hu, and Jianfeng Gao. Learning visual relation priors for image-text matching and image captioning with neural scene graph generators. *arXiv preprint arXiv:1909.09953*, 2019.

[21] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[22] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. Identity-aware textual-visual matching with latent co-attention. In *ICCV*, 2017.

[23] Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. Person search with natural language description. In *CVPR*, 2017.

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[25] Xiao Lin and Devi Parikh. Leveraging visual question answering for image-caption ranking. In *ECCV*, 2016.

[26] Chunxiao Liu, Zhendong Mao, An-An Liu, Tianzhu Zhang, Bin Wang, and Yongdong Zhang. Focus your attention: A bidirectional focal attention network for image-text matching. In *ACMMM*, 2019.

[27] Yu Liu, Yanming Guo, Erwin M. Bakker, and Michael S. Lew. Learning a recurrent residual fusion network for multimodal matching. In *ICCV*, 2017.

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[29] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NeurIPS*, 2016.

[30] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, 2015.

[31] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *NeurIPS*, 2014.

[32] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017.

[33] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal LSTM for dense visual-semantic embedding. In *ICCV*, 2017.

[34] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[35] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 1997.

[36] Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016.

[37] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *CVPR*, 2019.

[38] Marijn F. Stollenga, Jonathan Masci, Faustino J. Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NeurIPS*, 2014.

[39] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016.

[40] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *arXiv preprint arXiv:1704.03470*, 2017.

[41] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. CAMP: cross-modal adaptive message passing for text-image retrieval. *arXiv preprint arXiv:1909.05506*, 2019.

[42] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016.

[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[44] Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. In *CVPR*, 2015.

[45] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[46] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.

[47] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. Dual-path convolutional image-text embedding. *arXiv preprint arXiv:1711.0553*, 2017.

SIFT Meets CNN: A Decade Survey of Instance Retrieval

Liang Zheng, Yi Yang, and Qi Tian, *Fellow, IEEE*

Abstract—In the early days, content-based image retrieval (CBIR) was studied with global features. Since 2003, image retrieval based on local descriptors (*de facto* SIFT) has been extensively studied for over a decade due to the advantage of SIFT in dealing with image transformations. Recently, image representations based on the convolutional neural network (CNN) have attracted increasing interest in the community and demonstrated impressive performance. Given this time of rapid evolution, this article provides a comprehensive survey of instance retrieval over the last decade. Two broad categories, SIFT-based and CNN-based methods, are presented. For the former, according to the codebook size, we organize the literature into using large/medium-sized/small codebooks. For the latter, we discuss three lines of methods, i.e., using pre-trained or fine-tuned CNN models, and hybrid methods. The first two perform a single-pass of an image to the network, while the last category employs a patch-based feature extraction scheme. This survey presents milestones in modern instance retrieval, reviews a broad selection of previous works in different categories, and provides insights on the connection between SIFT and CNN-based methods. After analyzing and comparing retrieval performance of different categories on several datasets, we discuss promising directions towards generic and specialized instance retrieval.

Index Terms—Instance retrieval, SIFT, convolutional neural network, literature survey

1 Introduction

Content-based image retrieval (CBIR) has been a long-standing research topic in the computer vision society. In the early 1990s, the study of CBIR truly started. Images were indexed by the visual cues, such as texture and color, and a myriad of algorithms and image retrieval systems have been proposed. A straightforward strategy is to extract global descriptors. This idea dominated the image retrieval community in the 1990s and early 2000s. Yet, a well-known problem is that global signatures may fail the invariance expectation to image changes such as illumination, translation, occlusion and truncation. These variances compromise the retrieval accuracy and limit the application scope of global descriptors. This problem has given rise to local feature based image retrieval.

The focus of this survey is instance-level image retrieval. In this task, given a query image depicting a particular object/scene/architecture, the aim is to retrieve images containing the same object/scene/architecture that may be captured under different views, illumination, or with occlusions. Instance retrieval departs from class retrieval [1] in that the latter aims at retrieving images of the same class with the query. In the following, if not specified, we use "image retrieval" and "instance retrieval" interchangeably.

The milestones of instance retrieval in the past years are presented in Fig. 1, in which the times of the SIFT-based and CNN-based methods are highlighted. The majority of traditional methods can be considered to end in 2000 when Smeulders et al. [2] presented a comprehensive survey of CBIR "at the end of the early years". Three years later (2003) the Bag-of-Words (BoW) model was introduced to the image retrieval community [3], and in 2004 was applied to image classification [4], both relying on the SIFT descriptor [5]. The retrieval community has since witnessed the prominence of the BoW model for over a decade during which many improvements were proposed. In 2012, Krizhevsky et al. [6] with the AlexNet achieved the state-of-the-art recognition accuracy in ILSRVC 2012, exceeding previous best results by a large margin. Since then, research focus has begun to transfer to deep learning based methods [7], [8], [9], [10], especially the convolutional neural network (CNN).

The SIFT-based methods mostly rely on the BoW model. BoW was originally proposed for modeling documents because the text is naturally parsed into words. It builds a word histogram for a document by accumulating word responses into a global vector. In the image domain, the introduction of the scale-invariant feature transform (SIFT) [5] makes the BoW model feasible [3]. Originally, SIFT is comprised of a detector and descriptor, but which are used in isolation now; in this survey, if not specified, SIFT usually refers to the 128-dim descriptor, a common practice in the community. With a pre-trained codebook (vocabulary), local features are quantized to visual words. An image can thus be represented in a similar form to a document, and classic weighting and indexing schemes can be leveraged.

- *L. Zheng and Y. Yang are with the Centre for AI, University of Technology at Sydney, Ultimo, NSW 2007, Australia. E-mail: {liang.zheng, yi.yang}@uts.edu.au.*
- *Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78256. E-mail: qitian@cs.utsa.edu.*

Manuscript received 5 Aug. 2016; revised 20 May 2017; accepted 22 May 2017. Date of publication 29 May 2017; date of current version 10 Apr. 2018. (Corresponding author: Yi Yang.)
Recommended for acceptance by S. Lazebnik.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2709749

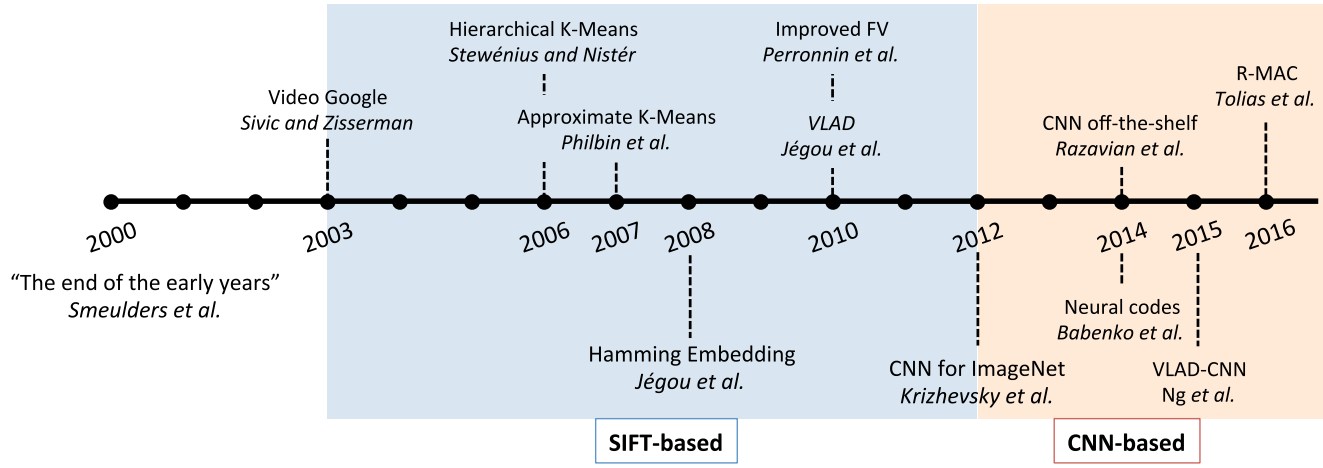


Fig. 1. Milestones of instance retrieval. After a survey of methods before the year 2000 by Smeulders et al. [2], Sivic and Zisserman [3] proposed Video Google in 2003, marking the beginning of the BoW model. Then, the hierarchical k-means and approximate k-means were proposed by Stewénius and Nistér [11] and Philbin et al. [12], respectively, marking the use of large codebooks in retrieval. In 2008, Jégou et al. [13] proposed Hamming Embedding, a milestone in using medium-sized codebooks. Then, compact visual representations for retrieval were proposed by Perronnin et al. [14] and Jégou et al. [15] in 2010. Although SIFT-based methods were still moving forward, CNN-based methods began to gradually take over, following the pioneering work of Krizhevsky et al. [6]. In 2014, Razavian et al. [7] proposed a hybrid method extracting multiple CNN features from an image. Babenko et al. [8] were the first to fine-tune a CNN model for generic instance retrieval. Both [9], [10] employ the column features from pre-trained CNN models, and [10] inspires later state-of-the-art methods. These milestones are the representative works of the categorization scheme in this survey.

In recent years, the popularity of SIFT-based models seems to be overtaken by the convolutional neural network, a hierarchical structure that has been shown to outperform hand-crafted features in many vision tasks. In retrieval, competitive performance compared to the BoW models has been reported, even with short CNN vectors [10], [16], [17]. The CNN-based retrieval models usually compute compact representations and employ the Euclidean distance or some approximate nearest neighbor (ANN) search methods for retrieval. Current literature may directly employ the pre-trained CNN models or perform fine-tuning for specific retrieval tasks. A majority of these methods feed the image into the network only once to obtain the descriptor. Some are based on patches which are passed to the network multiple times, a similar manner to SIFT; we classify them into hybrid methods in this survey.

1.1 Organization of This Paper

Upon the time of change, this paper provides a comprehensive literature survey of both the SIFT-based and CNN-based instance retrieval methods. We first present the categorization methodology in Section 2. We then describe the two major method types in Sections 3 and 4, respectively. On several benchmark datasets, Section 5 summarizes the comparisons between SIFT- and CNN-based methods. In Section 6, we point out two possible future directions. This survey will be concluded in Section 7.

2 Categorization Methodology

According to the different visual representations, this survey categorizes the retrieval literature into two broad types: SIFT-based and CNN-based. The SIFT-based methods are further organized into three classes: using large, medium-sized or small codebooks. We note that the codebook size is closely related to the choice of encoding methods. The CNN-based methods are categorized into using pre-trained or fine-tuned CNN models, as well as hybrid methods. Their similarities and differences are summarized in Table 1.

The SIFT-based methods had been predominantly studied before 2012 [6] (good works also appear in recent years [18], [19]). This line of methods usually use one type of detector, e.g., Hessian-Affine, and one type of descriptor, e.g., SIFT. Encoding maps a local feature into a vector. Based on the size of the codebook used during encoding, we classify SIFT-based methods into three categories as below.

TABLE 1
Major Differences between Various Types of Instance Retrieval Models

method type		detector	descriptor	encoding	dim.	indexing
SIFT-based	Large voc.	DoG, Hessian-Affine, dense patches, etc.	Local invariant descriptors such as SIFT	Hard, soft	High	Inverted index
	Mid voc.			Hard, soft, HE	Medium	Inverted index
	Small voc.			VLAD, FV	Low	ANN methods
CNN-based	Hybrid	Image patches	CNN features	VLAD, FV, pooling	Varies	ANN methods
	Pre-trained, single-pass	Column feat. or FC of pre-trained CNN models.		VLAD, FV, pooling	Low	ANN methods
	Fine-tuned, single-pass	A global feat. is end-to-end extracted from fine-tuned CNN models.			Low	ANN methods

For SIFT-based methods, hand-crafted local invariant features are extracted, and according to the codebook sizes, different encoding and indexing strategies are leveraged. For CNN-based methods, pre-trained, fine-tuned CNN models and hybrid methods are the primary types; fixed-length compact vectors are usually produced, combined with approximate nearest neighbor (ANN) methods.

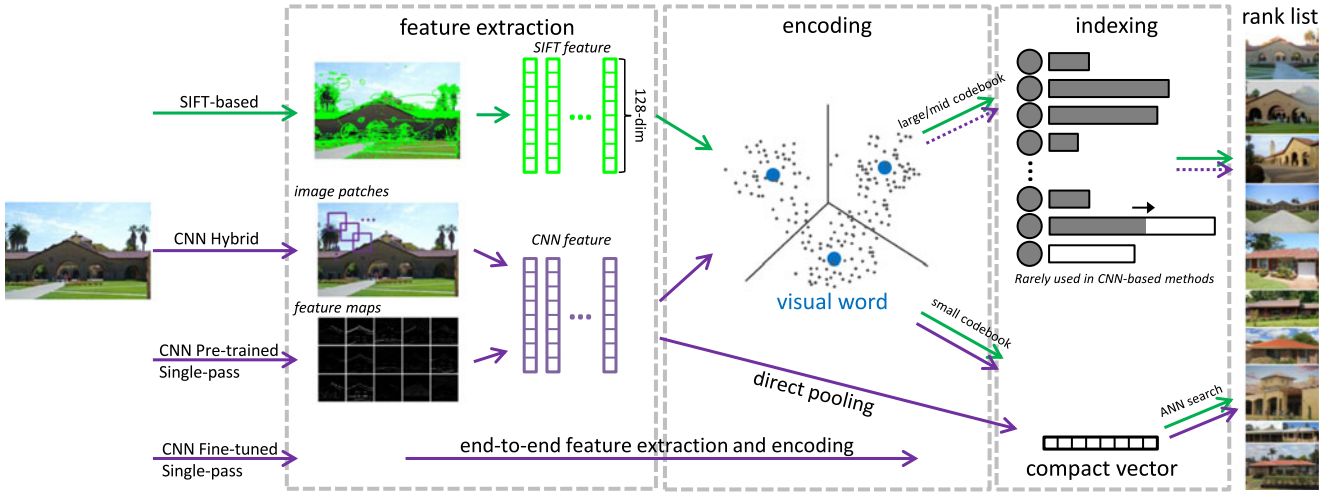


Fig. 2. A general pipeline of SIFT- and CNN-based retrieval models. Features are computed from hand-crafted detectors for SIFT, and densely applied filters or image patches for CNN. In both methods, under small codebooks, encoding/pooling is employed to produce compact vectors. In SIFT-based methods, the inverted index is necessary under large/medium-sized codebooks. The CNN features can also be computed in an end-to-end way using fine-tuned CNN models.

- *Using small codebooks.* The visual words are fewer than several thousand. Compact vectors are generated [14], [15] before dimension reduction and coding.
- *Using medium-sized codebooks.* Given the sparsity of BoW and the low discriminative ability of visual words, the inverted index and binary signatures are used [13]. The trade-off between accuracy and efficiency is a major influencing factor [20].
- *Using large codebooks.* Given the sparse BoW histograms and the high discriminative ability of visual words, the inverted index and memory-friendly signatures are used [21]. Approximate methods are used in codebook generation and encoding [11], [12].

The CNN-based methods extract features using CNN models. Compact (fixed-length) representations are usually built. There are three classes:

- *Hybrid methods.* Image patches are fed into CNN multiple times for feature extraction [7]. Encoding and indexing are similar to SIFT-based methods [22].
- *Using pre-trained CNN models.* Features are extracted in a single pass using CNN pre-trained on some large-scale datasets like ImageNet [23]. Compact Encoding/pooling techniques are used [9], [10].
- *Using fine-tuned CNN models.* The CNN model (e.g., pre-trained on ImageNet) is fine-tuned on a training set in which the images share similar distributions with the target database [8]. CNN features can be extracted in an end-to-end manner through a single pass to the CNN model. The visual representations exhibit improved discriminative ability [17], [24].

3 SIFT-Based Image Retrieval

3.1 Pipeline

The pipeline of SIFT-based retrieval is introduced in Fig. 2.

Local Feature Extraction. Suppose we have a gallery $\mathcal{G}$ consisting of N images. Given a feature detector, we extract local descriptors from the regions around the sparse interest points or dense patches. We denote the local descriptors of D detected regions in an image as $\{f_i\}_{i=i}^{D}, f_i \in \mathbb{R}^p$.

Codebook Training. SIFT-based methods train a codebook offline. Each visual word in the codebook lies in the center of a subspace, called the "Voronoi cell". A larger codebook corresponds to a finer partitioning, resulting in more discriminative visual words and vice versa. Suppose that a pool of local descriptors $\mathcal{F} \equiv \{f_i\}_{i=1}^{M}$ are computed from an unlabeled training set. The baseline approach, i.e., k-means, partitions the M points into K clusters; the K visual words thus constitute a codebook of size K.

Feature Encoding. A local descriptor $f_i \in \mathbb{R}^p$ is mapped into a feature embedding $g_i \in \mathbb{R}^l$ through the feature encoding process, $f_i \rightarrow g_i$. When k-means clustering is used, f_i can be encoded according to its distances to the visual words. For large codebooks, hard [11], [12] and soft quantization [25] are good choices. In the former, the resulting embedding g_i has only one non-zero entry; in the latter, f_i can be quantized to a small number of visual words. A global signature is produced after a sum-pooling of all the embeddings of local features. For medium-sized codebooks, additional binary signatures can be generated to preserve the original information. When using small codebooks, popular encoding schemes include vector of locally aggregated descriptors (VLAD) [15], Fisher vector (FV) [14], etc.

3.2 Local Feature Extraction

Local invariant features aim at accurate matching of local structures between images [26]. SIFT-based methods usually share a similar feature extraction step composed of a feature detector and a descriptor.

Local Detector. The *interest point detectors* aim to reliably localize a set of stable local regions under various imaging conditions. In the retrieval community, finding affine-covariant regions has been preferred. It is called "covariant" because the shapes of the detected regions change with the affine transformations, so that the region content (descriptors) can be invariant. This kind of detectors are different from keypoint-centric detectors such as the Hessian detector [27], and from those focusing on scale-invariant regions such as the difference of Gaussians (DoG) [5] detector. Elliptical regions which are adapted to the local intensity patterns

are produced by affine detectors. This ensures that the same local structure is covered under deformations caused by viewpoint variances, a problem often encountered in instance retrieval. In the milestone work [3], the Maximally Stable Extremal Region (MSER) detector [28] and the affine extended Harris-Laplace detector are employed, both of which are affine-invariant region detectors. MSER is used in several later works [11], [29]. Starting from [12], the Hessian-affine detector [30] has been widely adopted in retrieval. It has been shown to be superior to the difference of Gaussians detector [13], [31], due to its advantage in reliably detecting local structures under large viewpoint changes. To fix the orientation ambiguity of these affine-covariant regions, the gravity assumption is made [32]. The practice which dismisses the orientation estimation is employed by later works [33], [34] and demonstrates consistent improvement on architecture datasets where the objects are usually upright. Other non-affine detectors have also been tested in retrieval, such as the Laplacian of Gaussian (LOG) and Harris detectors used in [35]. For objects with smooth surfaces [36], few interest points can be detected, so the object boundaries are good candidates for local description.

On the other hand, some employ the *dense region detectors*. In the comparison between densely sampled image patches and the detected patches, Sicre et al. [37] report the superiority of the former. To recover the rotation invariance of dense sampling, the dominant angle of patches is estimated in [38]. A comprehensive comparison of various dense sampling strategies, the interest point detectors, and those in between can be accessed in [39].

Local Descriptor. With a set of detected regions, descriptors encode the local content. SIFT [5] has been used as the default descriptor. The 128-dim vector has been shown to outperform competing descriptors in matching accuracy [40]. In an extension, PCA-SIFT [41] reduces the dimension from 128 to 36 to speed up the matching process at the cost of more time in feature computation and loss of distinctiveness. Another improvement is RootSIFT [33], calculated by two steps: 1) ℓ_1 normalize the SIFT descriptor, 2) square root each element. RootSIFT is now used as a routine in SIFT-based retrieval. Apart from SIFT, SURF [42] is also widely used. It combines the Hessian-Laplace detector and a local descriptor of the local gradient histograms. The integral image is used for acceleration. SURF has a comparable matching accuracy with SIFT and is faster to compute. See [43] for comparisons between SIFT, PCA-SIFT, and SURF. To further accelerate the matching speed, binary descriptors [44] replace Euclidean distance with Hamming distance during matching.

Apart from hand-crafted descriptors, some also propose learning schemes to improve the discriminative ability of local descriptors. For example, Philbin et al. [45] proposes a non-linear transformation so that the projected SIFT descriptor yields smaller distances for true matches. Simoyan et al. [34] improve this process by learning both the pooling region and a linear descriptor projection.

3.3 Retrieval Using Small Codebooks

A small codebook has several thousand, several hundred or fewer visual words, so the computational complexity of codebook generation and encoding is moderate. Representative works include BoW [3], VLAD [15] and FV [14]. We mainly discuss VLAD and FV and refer readers to [46] for a comprehensive evaluation of the BoW compact vectors.

3.3.1 Codebook Generation

Clustering complexity depends heavily on the codebook size. In works based on VLAD [15] or FV [14], the codebook sizes are typically small, e.g., 64, 128, 256. For VLAD, flat k-means is employed for codebook generation. For FV, the Gaussian mixture model (GMM), i.e., $u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x)$, where K is the number of Gaussian mixtures, is trained using the maximum likelihood estimation. GMM describes the feature space with a mixture of K Gaussian distributions, and can be denoted as $\lambda = \{w_i, \mu_i, \sum_i, i = 1, \ldots, K\}$, where w_i, μ_i and $\sum_i$ represent the mixture weight, the mean vector and the covariance matrix of Gaussian u_i, respectively.

3.3.2 Encoding

Due to the small codebook size, relative complex and information-preserving encoding techniques can be applied. We mainly describe FV, VLAD and their improvements in this section. With a pre-trained GMM model, FV describes the averaged first and second order difference between local features and the GMM centers. Its dimension is $2pK$, where p is the dimension of the local descriptors and K is the codebook size of GMM. FV usually undergoes power normalization [47], [14] to suppress the burstiness problem (to be described in Section 3.4.3). In this step, each component of FV undergoes non-linear transformation featured by parameter α, $x_i := \text{sign}(x_i)\|x_i\|^\alpha$. Then ℓ_2 normalization is employed. Later, FV is improved from different aspects. For example, Koniusz et al. [48] augment each descriptor with its spatial coordinates and associated tunable weights. In [49], larger codebooks (up to 4,096) are generated and demonstrate superior classification accuracy to smaller codebooks, at the cost of computational efficiency. To correct the assumption that local regions are identically and independently distributed (iid), Cinbis et al. [50] propose non-iid models that discount the burstiness effect and yield improvement over the power normalization.

The VLAD encoding scheme proposed by Jégou et al. [15] can be thought of as a simplified version of FV. It quantizes a local feature to its nearest visual word in the codebook and records the difference between them. Nearest neighbor search is performed because of the small codebook size. The residual vectors are then aggregated by sum pooling followed by normalizations. The dimension of VLAD is pK. Comparisons of some important encoding techniques are presented in [51], [52]. Again, the improvement of VLAD comes from multiple aspects. In [53], Jégou and Chum suggest the usage of PCA and whitening (denoted as PCA_w in Table 5) to de-correlate visual word co-occurrences, and the training of multiple codebooks to reduce quantization loss. In [54], Arandjelović et al. extend VLAD in three aspects: 1) normalize the residual sum within each coarse cluster, called intra-normalization, 2) vocabulary adaptation to address the dataset transfer problem and 3) multi-VLAD for small object discovery. Concurrent to [54], Delhumeau et al. [55] propose to normalize each residual vector instead of the residual sums; they also advocate for local PCA within each Voronoi cell which does not perform

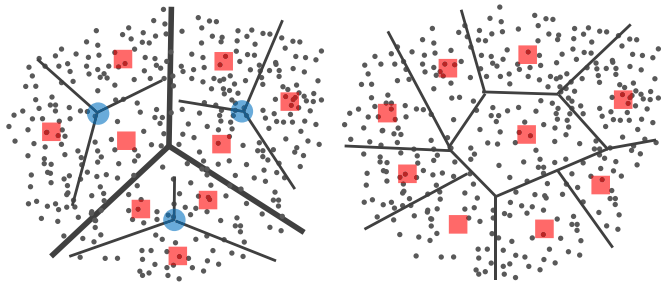
dimension reduction as [52]. A recent work [56] employs soft assignment and empirically learns optimal weights for each rank to improve over the hard quantization.

Note that some general techniques benefit various embedding methods, such as VLAD, FV, BoW, locality-constrained linear coding (LLC) [57] and monomial embeddings. To improve the discriminative ability of embeddings, Tolias et al. [58] propose the orientation covariant embedding to encode the dominant orientation of the SIFT regions jointly with the SIFT descriptor. It achieves a similar covariance property to weak geometric consistency (WGC) [13] by using geometric cues within regions of interest so that matching points with similar dominant orientations are up-weighted and vice versa. The triangulation embedding [18] only considers the direction instead of the magnitude of the input vectors. Jégou et al. [18] also present a democratic aggregation that limits the interference between the mapped vectors. Baring a similar idea with democratic aggregation, Murray and Perronnin [59] propose the generalized max pooling (GMP) optimized by equalizing the similarity between the pooled vector and each coding representation.

The computational complexity of BoW, VLAD and FV is similar. We neglect the offline training and SIFT extraction steps. During visual word assignment, each feature should compute its distance (or soft assignment coefficient) with all the visual words (or Gaussians) for VLAD (or FV). So this step has a complexity of $\mathcal{O}(pK)$. In the other steps, complexity does not exceed $\mathcal{O}(pK)$. Considering the sum-pooling of the embeddings, the encoding process has an overall complexity of $\mathcal{O}(pKD)$, where D is the number of features in an image. Triangulation embedding [18], a variant of VLAD, has a similar complexity. The complexity of multi-VLAD [54] is $\mathcal{O}(pKD)$, too, but it has a more costly matching process. Hierarchical VLAD [60] has a complexity of $\mathcal{O}(pKK'D)$, where K' is the size of the secondary codebook. In the aggregation stage, both GMP [59] and democratic aggregation [18] have high complexity. The complexity of GMP is $\mathcal{O}(\frac{P^2}{K})$, where P is the dimension of the feature embedding, while the computational cost of democratic aggregation comes from the Sinkhorn algorithm.

3.3.3 ANN Search

Due to the high dimensionality of the VLAD/FV embeddings, efficient compression and ANN search methods have been employed [61], [62]. For example, the principle component analysis (PCA) is usually adapted to for dimension reduction, and it is shown that retrieval accuracy even increases after PCA [53]. For hashing-based ANN methods, Perronnin et al. [47] use standard binary encoding techniques such as locality sensitive hashing [63] and spectral hashing [64]. Nevertheless, when being tested on the SIFT and GIST feature datasets, spectral hashing is shown to be outperformed by Product Quantization (PQ) [61]. In these quantization-based ANN methods, PQ is demonstrated to be better than other popular ANN methods such as FLANN [62] as well. A detailed discussion of VLAD and PQ can be viewed in [65]. PQ has since then been improved in a number of works. In [66], Douze et al. propose to re-order the cluster centroids so that adjacent centroids have small Hamming distances. This method is compatible with Hamming distance based ANN search, which offers significant speedup for PQ. We refer readers to [67] for a survey of ANN approaches.



(a) HKM (b) AKM

Fig. 3. Two milestone clustering methods (a) hierarchical k-means (HKM) [11] and (b) approximate k-means (AKM) [12] for large codebook generation. Bold borders and blue discs are the clustering boundaries and centers of the first layer of HKM. Slim borders and red squares are the final clustering results in both methods.

We also mention an emerging ANN technique, i.e., group testing [68], [69], [70]. In a nutshell, the database is decomposed into groups, each represented by a group vector. Comparisons between the query and group vectors reveal how likely a group contains a true match. Since group vectors are much fewer than the database vectors, search time is reduced. Iscen et al. [69] propose to directly find the best group vectors summarizing the database without explicitly forming the groups, which reduces the memory consumption.

3.4 Retrieval Using Large Codebooks

A large codebook may contain 1 million [11], [12] visual words or more [71], [72]. Some major steps undergo important changes compared with using small codebooks.

3.4.1 Codebook Generation

Approximate methods are critical in assigning data into a large number of clusters. In the retrieval community, two representative works are hierarchical k-means (HKM) [11] and approximate k-means (AKM) [12], as illustrated in Figs. 1 and 3. Proposed in 2006, HKM applies standard k-means on the training features hierarchically. It first partitions the points into a few clusters (e.g., $\bar{k} \ll K$) and then recursively partitions each cluster into further clusters. In every recursion, each point should be assigned to one of the $\bar{k}$ clusters, with the depth of the cluster tree being $\mathcal{O}(\log K)$, where K is the target cluster number. The computational cost of HKM is therefore $\mathcal{O}(\bar{k}M \log K)$, where M is the number of training samples. It is much smaller than the complexity of flat k-means $\mathcal{O}(MK)$ when K is large (a large codebook).

The other milestone in large codebook generation is AKM [12]. This method indexes the K cluster centers using a forest of random k-d trees so that the assignment step can be performed efficiently with ANN search. In AKM, the cost of assignment can be written as $\mathcal{O}(K \log K + vM \log K) = \mathcal{O}(vM \log K)$, where v is the number of nearest cluster candidates to be accessed in the k-d trees. So the computational complexity of AKM is on par with HKM and is significantly smaller than flat k-means when K is large. Experiments show that AKM is superior to HKM [12] due to its lower quantization error (see Section 3.4.2). In most AKM-based methods, the default choice for ANN search is FLANN [62].

3.4.2 Feature Encoding (Quantization)

Feature encoding is interleaved with codebook clustering, because ANN search is critical in both components. The ANN techniques implied in some classic methods like AKM and HKM can be used in both clustering and encoding steps. Under a large codebook, the key trade-off is between quantization error and computational complexity. In the encoding step, information-preserving encoding methods such as FV [14], sparse coding [73] are mostly infeasible due to their computational complexity. It therefore remains a challenging problem how to reduce the quantization error while keeping the quantization process efficient.

Fro the ANN methods, the earliest solution is to quantize a local feature along the hierarchical tree structure [11]. Quantized tree nodes in different levels are assigned different weights. However, due to the highly imbalanced tree structure, this method is outperformed by k-d tree based quantization method [12]: one visual word is assigned to each local feature, using a k-d tree built from the codebook for fast ANN search. In an improvement to this hard quantization scheme, Philbin et al. [25] propose soft quantization by quantizing a feature into several nearest visual words. The weight of each assigned visual word relates negatively to its distance from the feature by $\exp(-\frac{d^2}{2\sigma^2})$, where d is the distance between the descriptor and the cluster center. While soft quantization is based on the Euclidean distance, Mikulik et al. [71] propose to find relevant visual words for each visual word through an unsupervised set of matching features. Built on a probabilistic model, these alternative words tend to contain descriptors of matching features. To reduce the memory cost of soft quantization [25] and the number of query visual words, Cai et al. [74] suggest that when a local feature is far away from even the nearest visual word, this feature can be discarded without a performance drop. To further accelerate quantization, scalar quantization [75] suggests that local features be quantized without an explicitly trained codebook. A floating-point vector is binarized, and the first dimensions of the resulting binary vector are directly converted to a decimal number as a visual word. In the case of large quantization error and low recall, scalar quantization uses bit-flop to generate hundreds of visual words for a local feature.

3.4.3 Feature Weighting

TF-IDF. The visual words in codebook $\mathcal{C}$ are typically assigned specific weights, called the term frequency and inverse document frequency (TF-IDF), which are integrated with the BoW encoding. TF is defined as

$$\mathrm{TF}(c_i^j) = o_i^j, \tag{1}$$

where o_i^j is the number of occurrences of a visual word c_i within an image j. TF is thus a local weight. IDF, on the other hand, determines the contribution of a given visual word through global statistics. The classic IDF weight of visual word c_i is calculated as

$$\mathrm{IDF}(c_i) = \log\frac{N}{n_i}, \text{where } n_i = \sum_{j\in\mathcal{G}} \mathbf{1}(o_i^j > 0), \tag{2}$$

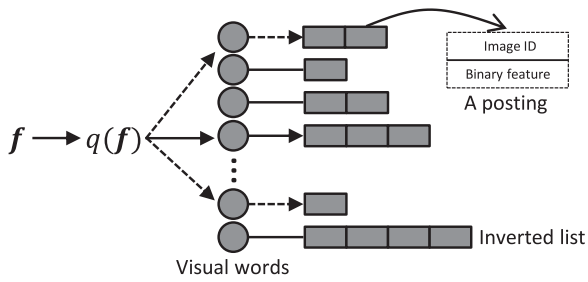


Fig. 4. The data structure of the inverted index. It physically contains K inverted lists, each consisting of some postings, which index the image ID and some binary signatures. During retrieval, a quantized feature will traverse the inverted list corresponding to its assigned visual word. Dashed line denotes soft quantization, in which multiple inverted lists are visited.

where N is the number of gallery images, and n_i encodes the number of images in which word c_i appears. The TF-IDF weight for visual word c_i in image j is

$$w(c_i^j) = \mathrm{TF}(c_i^j)\mathrm{IDF}(c_i). \tag{3}$$

Improvements. A major problem associated with visual word weighting is burstiness [76]. It refers to the phenomenon whereby repetitive structures appear in an image. This problem tends to dominate image similarity. Jégou et al. [76] propose several TF variants to deal with burstiness. An effective strategy consists in exerting a square operation on TF. Instead of grouping features with the same word index, Revaud et al. [77] propose detecting keypoint groups frequently happening in irrelevant images which are down-weighted in the scoring function. While the above two methods detect bursty groups after quantization, Shi et al. [19] propose detecting them in the descriptor stage. The detected bursty descriptors undergo average pooling and are fed in the BoW architectures. From the aspect of IDF, Zheng et al. [78] propose the $\mathcal{L}_p$-norm IDF to tackle burstiness and Murata et al. [79] design the exponential IDF which is later incorporated into the BM25 formula. When most works try to suppress burstiness, Torii et al. [80] view it as a distinguishing feature for architectures and design new similarity measurement following burstiness detection.

Another feature weighting strategy is feature augmentation on the database side [33], [81]. Both methods construct an image graph offline, with edges indicating whether two images share a same object. For [81], only features that pass the geometric verification are preserved, which reduces the memory cost. Then, the feature of the base image is augmented with all the visual words of its connecting images. This method is improved in [33] by only adding those visual words which are estimated to be visible in the augmented image, so that noisy visual words can be excluded.

3.4.4 The Inverted Index

The inverted index is designed to enable efficient storage and retrieval and is usually used under large/medium-sized codebooks. Its structure is illustrated in Fig. 4. The inverted index is a one-dimensional structure where each entry corresponds to a visual word in the codebook. An inverted list is attached to each word entry, and those indexed in the each inverted list are called indexed features or postings. The inverted index takes advantages of the sparse nature of the visual word histogram under a large codebook.

In literature, it is required that new retrieval methods be adjustable to the inverted index. In the baseline [11], [12], the image ID and term frequency (TF) are stored in a posting. When other information is integrated, they should be small in size. For example, in [82], the metadata are quantized, such as descriptor contextual weight, descriptor density, mean relative log scale and the mean orientation difference in each posting. Similarly, quantized spatial information such as the orientation can also be stored [21], [83]. In co-indexing [72], when the inverted index is enlarged with globally consistent neighbors, semantically isolated images are deleted to reduce memory consumption. In [84], the original one-dimensional inverted index is expanded to two-dimensional for ANN search, which learns a codebook for each SIFT sub-vector. Later, it is applied to instance retrieval by [31] to fuse local color and SIFT descriptors.

3.5 Retrieval Using Medium-Sized Codebooks

Medium-sized codebooks refer to those having 10-200k visual words. The visual words exhibit medium discriminative ability, and the inverted index is usually constructed.

3.5.1 Codebook Generation and Quantization

Considering the relatively small computational cost compared with large codebooks (Section 3.4.1), flat k-means can be adopted for codebook generation [20], [85]. It is also shown in [31], [86] that using AKM [12] for clustering also yields very competitive retrieval accuracy.

For quantization, nearest neighbor search can be used to find the nearest visual words in the codebook. Practice may tell that using some strict ANN algorithms produces competitive retrieval results. So comparing with the extensive study on quantization under large codebooks (Section 3.4.2) [25], [71], [74], relatively fewer works focus on the quantization problem under a medium-sized codebook.

3.5.2 Hamming Embedding and Its Improvements

The discriminative ability of visual words in medium-sized codebooks lies in between that of small and large codebooks. So it is important to compensate the information loss during quantization. To this end, a milestone work, i.e., Hamming embedding (HE) has been dominantly employed.

Proposed by Jégou et al. [13], HE greatly improves the discriminative ability of visual words under medium-sized codebooks. HE first maps a SIFT descriptor $f \in \mathbb{R}^p$ from the p-dimensional space to a p_b-dimensional space

$$x = P \cdot f = (x_1, \dots x_{p_b}), \tag{4}$$

where $P \in \mathbb{R}_b^p \times p$ is a projecting matrix, and x is a low-dimensional vector. By creating a matrix of random Gaussian values and applying a QR factorization to it, matrix P is taken as the first p_b rows of the resulting orthogonal matrix. To binarize x, Jegou et al. propose to compute the median vector $\overline{x_i} = (\overline{x_{1,i}}, \dots, \overline{x_{p_b,i}})$ of the low-dimensional vector using descriptors falling in each Voronoi cell c_i. Given descriptor f and its projected vector x, HE computes its visual word c_t, and the HE binary vector is computed as

$$b_j(x) = \begin{cases} 1 \text{ if } x_j > \overline{x_{j,t}}, \\ 0 \text{ otherwise}, \end{cases} \tag{5}$$

where $b(x) = (b_1(x), \dots, b_{p_b}(x))$ is the resulting HE vector of dimension p_b. The binary feature $b(x)$ serves as a secondary check for feature matching. A pair of local features are a true match when two criteria are satisfied: 1) identical visual words and 2) small Hamming distance between their HE signatures. The extension of HE [85] estimates the matching strength between feature f_1 and f_2 reversely to the Hamming distance by an exponential function

$$w_{\mathrm{HE}}(f_1, f_2) = \exp\left(-\frac{\mathcal{H}(b(x_1), b(x_2))}{2\gamma^2}\right), \tag{6}$$

where $b(x_1)$ and $b(x_2)$ are the HE binary vector of f_1 and f_2, respectively, $\mathcal{H}(\cdot,\cdot)$ computes the Hamming distance between two binary vectors, and γ is a weighting parameter. As shown in Fig. 6, HE [13] and its weighted version [85] improves accuracy considerably in 2008 and 2010.

Applications of HE include video copy detection [87], image classification [88] and re-ranking [89]. For example, in image classification, patch matching similarity is efficiently estimated by HE which is integrated into linear kernel-based SVM [88]. In image re-ranking, Tolias et al. [89] use lower HE thresholds to find strict correspondences which resemble those found by RANSAC, and the resulting image subset is more likely to contain true positives for query reformulation.

The improvement over HE has been observed in a number of works, especially from the view of match kernel [20]. To reduce the information loss on the query side, Jain et al. [90] propose a vector-to-binary distance comparison. It exploits the vector-to-hyperplane distance while retaining the efficiency of the inverted index. Further, Qin et al. [91] design a higher-order match kernel within a probabilistic framework and adaptively normalize the local feature distances by the distance distribution of false matches. This method is in the spirit similar to [92], in which the word-word distance, instead of the feature-feature distance [91], is normalized, according to the neighborhood distribution of each visual word. While the average distance between a word to its neighbors is regularized to be almost constant in [92], the idea of democratizing the contribution of individual embeddings has later been employed in [18]. In [20], Tolias et al. show that VLAD and HE share similar natures and propose a new match kernel which trades off between local feature aggregation and feature-to-feature matching, using a similar matching function to [91]. They also demonstrate that using more bits (e.g., 128) in HE is superior to the original 64 bits scheme at the cost of decreased efficiency. Even more bits (256) are used in [75], but this method may be prone to relatively low recall.

3.6 Other Important Issues

3.6.1 Feature Fusion

Local-Local Fusion. A problem with the SIFT feature is that only local gradient description is provided. Other discriminative information encoded in an image is still not leveraged. In Fig. 5B, a pair of false matches cannot be rejected by HE due to their similarity in the SIFT space, but the fusion of other local (or regional) features may correct this problem. A good choice for local-local fusion is to couple SIFT with color descriptors. The usage of color-SIFT descriptors can partially

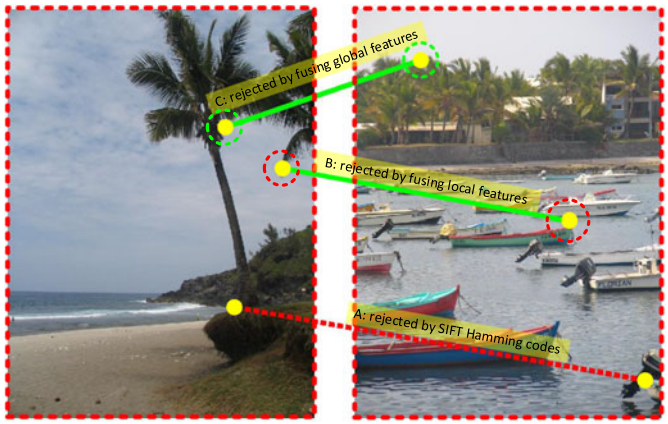


Fig. 5. False match removal by (A) HE [13], (B) local-local feature fusion, and (C) local-global feature fusion.

address the trade-off between invariance and discriminative ability. Evaluation has been conducted on several recognition benchmarks [93] of the descriptors such as HSV-SIFT [94], HueSIFT [95] and OpponentSIFT [93]. Both HSV-SIFT and HueSIFT are scale-invariant and shift-invariant. OpponentSIFT describes all the channels in the opponent color space using the SIFT descriptor and is largely robust to the light color changes. In [93], OpponentSIFT is recommended when no prior knowledge about the datasets is available. In more recent works, the binary color signatures are stored in the inverted index [31], [96]. Despite the good retrieval accuracy on some datasets, the potential problem is that intensive variation in illumination may compromise the effectiveness of colors.

Local-Global Fusion. Local and global features describe images from different aspects and can be complementary. In Fig. 5C, when local (and regional) cues are not enough to reject a false match pair, it would be effective to further incorporate visual information from a larger context scale. Early and late fusion are two possible ways. In early fusion, the image neighborhood relationship mined by global features such as FC8 in AlexNet [6] is fused in the SIFT-based inverted index [72]. In late fusion, Zhang et al. [97] build an offline graph for each type of feature, which is subsequently fused during the online query. In an improvement of [97], Deng et al. [98] add weakly supervised anchors to aid graph fusion. Both works on the rank level. For score-level fusion, automatically learned category-specific attributes are combined with pre-trained category-level information [99]. Zheng et al. [86] propose the query-adaptive late fusion by extracting a number of features (local or global, good or bad) and weighting them in a query-adaptive manner.

3.6.2 Geometric Matching

A frequent concern with the BoW model is the lack of geometric constraints among local features. Geometric verification can be used as a critical pre-processing step various scenarios, such as query expansion [100], [101], feature selection [81], database-side feature augmentation [33], [81], large-scale object mining [102], etc. The most well-known method for global spatial verification is RANSAC [12]. It calculates affine transformations for each correspondence repeatedly which are verified by the number of inliers that fit the transformation. RANSAC is effective in re-ranking a subset of top-ranked images but has efficiency problems. As a result, how to efficiently and accurately incorporate spatial cues in the SIFT-based framework has been extensively studied.

A good choice is to discover the spatial context among local features. For example, visual phrases [103], [104], [105], [106] are generated among individual visual words to provide more strict matching criterion. Visual word co-occurrences in the entire image are estimated [107] and aggregated [108], while in [29], [109], [110] visual word clusters within local neighborhoods are discovered. Visual phrases can also be constructed from adjacent image patches [103], random spatial partitioning [106], and localized stable regions [29] such as MSER [28].

Another strategy uses voting to check geometric consistency. In the voting space, a bin with a larger value is more likely to represent the true transformation. An important work is weak geometrical consistency [13], which focuses on the difference in scale and orientation between matched features. The space of difference is quantized into bins. Hough voting is used to locate the subset of correspondences similar in scale or orientation differences. Many later works can be viewed as extensions of WGC. For example, the method of Zhang et al. [21] can be viewed as WGC using x, y offsets instead of scale and orientation. This method is invariant to object translations, but may be sensitive to scale and rotation changes due to the rigid coordinate quantization. To regain the scale and the rotation variance, Shen et al. [111] quantize the angle and scale of the query region after applying several transformations. A drawback of [111] is that query time and memory cost are both increased. To enable efficient voting and alleviate quantization artifacts, Hough pyramid matching (HPM) [112] distributes the matches over a hierarchical partition of the transformation space. HPM trades off between flexibility and accuracy and is very efficient. Quantization artifact can also be reduced by allowing a single correspondence to vote for multiple bins [113]. HPM and [113] are much faster than RANSAC and can be viewed as extensions in the rotation and the scale invariance to the weak geometry consistency proposed along with Hamming Embedding [13]. In [114], a rough global estimate of orientation and scale changes is made by voting, which is used to verify the transformation obtained by the matched features. A recent method [115] combines the advantage of hypothesis-based methods such as RANSAC [12] and voting-based methods [21], [112], [113], [114]. Possible hypothesises are identified by voting and later verified and refined. This method inherits efficiency from voting and supports query expansion since it outputs an explicit transformation and a set of inliers.

3.6.3 Query Expansion

As a post-processing step, query expansion (QE) significantly improves the retrieval accuracy. In a nutshell, a number of top-ranked images from the original rank list are employed to issue a new query which is in turn used to obtain a new rank list. QE allows additional discriminative features to be added to the original query, thus improving recall.

In instance retrieval, Chum et al. [100] are the first to exploit this idea. They propose the average query expansion (AQE) which averages features of the top-ranked images to issue the new query. Usually, spatial verification [12] is employed for re-ranking and obtaining the ROIs from

which the local features undergo average pooling. AQE is used by many later works [10], [17], [24] as a standard tool. The recursive AQE and the scale-band recursive QE are effective improvement but incur more computational cost [100]. Four years later, Chum et al. [101] improve QE from the perspectives of learning background confusers, expanding the query region and incremental spatial verification. In [33], a linear SVM is trained online using the top-ranked and bottom-ranked images as positive and negative training samples, respectively. The learned weight vector is used to compute the average query. Other important extensions include "hello neighbor" based on reciprocal neighbors [116], QE with rank-based weighting [111], Hamming QE [89] (see Section 3.5), etc.

3.6.4 Small Object Retrieval

Retrieving objects that cover a small portion of images is a challenging task due to 1) the few detected local features and 2) the large amount of background noise. The Instance Search task in the TRECVID campaign [117] and the task of logo retrieval are important venues/applications for this task.

Generally speaking, both TRECVID and logo retrieval can be tackled with similar pipelines. For keypoint-based methods, the spatial context among the local features is important to discriminative target objects from others, especially in cases of rigid objects. Examples include [118], [119], [120]. Other effective methods include burstiness handling [77] (discussed in Section 3.4.3), considering the different inlier ratios between the query and target objects [121], etc. In the second type of methods, effective region proposals [122] or multi-scale image patches [123] can be used as object region candidates. In [123], a recent state-of-the-art method, a regional diffusion mechanism based on neighborhood graphs is proposed to further improve the recall of small objects.

4 CNN-Based Image Retrieval

CNN-based retrieval methods have constantly been proposed in recent years and are gradually replacing the hand-crafted local detectors and descriptors. In this survey, CNN-based methods are classified into three categories: using pre-trained CNN models, using fine-tuned CNN models and hybrid methods. The first two categories compute the global feature with a single network pass, and the hybrid methods may require multiple network passes (see Fig. 2).

4.1 Retrieval Using Pre-Trained CNN Models

This type of methods is efficient in feature computation due to the single-pass mode. Given the transfer nature, its success lies in the feature extraction and encoding steps. We will first describe some commonly used datasets and networks for pre-training, and then the feature computation process.

4.1.1 Pre-Trained CNN Models

Popular CNN Architectures. Several CNN models serve as good choices for extracting features, including AlexNet [6], VGGNet [124], GoogleNet [125] and ResNet [126], which are listed in Table 2. Briefly, CNN can be viewed as a set of non-linear functions and is composed of a number of layers such as convolution, pooling, non-linearities, etc. CNN has a hierarchical structure. From bottom to top layers, the image undergoes convolution with filters, and the receptive field of these image filters increases. Filters in the same layer have the same size but different parameters. AlexNet [6] was proposed the earliest among these networks, which has five convolutional layers and three fully connected (FC) layers. It has 96 filters in the first layer of sizes $11 \times 11 \times 3$ and has 256 filters of size $3 \times 3 \times 192$ in the 5th layer. Zeiler et al. [127] observe that the filters are sensitive to certain visual patterns and that these patterns evolve from low-level bars in bottom layers to high-level objects in top layers. For low-level and simple visual stimulus, the CNN filters act as the detectors in the local hand-crafted features, but for the high-level and complex stimulus, the CNN filters have distinct characteristics that depart from SIFT-like detectors. AlexNet has been shown to be outperformed by newer ones such as VGGNet, which has the largest number of parameters. ResNet and GoogleNet won the ILSVRC 2014 and 2015 challenges, respectively, showing that CNNs are more effective with more layers. A full review of these networks is beyond the scope of this paper, and we refer readers to [6], [128], [124] for details.

TABLE 2
Pre-Trained CNN Models That Can Be Used

models	size	# layers	training Set	used in
OverFeat [132]	144M	6+3	ImageNet	[7]
AlexNet [6]			ImageNet	[22], [133]
PlacesNet [129]	60M	5+3	Places	[130], [131]
HybridNet [129]			ImageNet+Places	[130], [131]
VGGNet [124]	138M	13+3	ImageNet	[10]
GoogleNet [125]	11M	22	ImageNet	[9]
ResNet [126]	44.6M	101	ImageNet	n.a

Datasets for Pre-Training. Several large-scale recognition datasets are used for CNN pre-training. Among them, the ImageNet dataset [23] is mostly commonly used. It contains 1.2 million images of 1,000 semantic classes and is usually thought of as being generic. Another data source for pre-training is the Places-205 dataset [129] which is twice as large as ImageNet but has five times fewer classes. It is a scene-centric dataset depicting various indoor and outdoor scenes. A hybrid dataset combining the Places-205 and the ImageNet datasets has also been used for pre-training [129]. The resulting HybridNet is evaluated in [125], [126], [130], [131] for instance retrieval.

The Transfer Issue. Comprehensive evaluation of various CNNs on instance retrieval has been conducted in several recent works [130], [131], [134]. The transfer effect is mostly concerned. It is considered in [130] that instance retrieval, as a target task, lies farthest from the source, i.e., ImageNet. Studies reveal some critical insights in the transfer process. First, during model transfer, features extracted from different layers exhibit different retrieval performance. Experiments confirm that the top layers may exhibit lower generalization ability than the layer before it. For example, for AlexNet pre-trained on ImageNet, it is shown that FC6, FC7, and FC8 are in descending order regarding retrieval accuracy [130]. It is also shown in [10], [134] that the pool5 feature of AlexNet and VGGNet is even superior to FC6

when proper encoding techniques are employed. Second, the source training set is relevant to retrieval accuracy on different datasets. For example, Azizpour et al. [130] report that HybridNet yields the best performance on Holidays after PCA. They also observe that AlexNet pre-trained on ImageNet is superior to PlacesNet and HybridNet on the Ukbench dataset [11] which contains common objects instead of architectures or scenes. So the similarity of the source and target plays a critical role in instance retrieval when using a pre-trained CNN model.

4.1.2 Feature Extraction

FC Descriptors. The most straightforward idea is to extract the descriptor from the fully-connected layer of the network [7], [8], [135], e.g., the 4,096-dim FC6 or FC7 descriptor in AlexNet. The FC descriptor is generated after layers of convolutions with the input image, has a global receptive field, and thus can be viewed as a global feature. It yields fair retrieval accuracy under Euclidean distance and can be improved with power normalization [14].

Intermediate Local Features. Many recent retrieval methods [9], [10], [134] focus on local descriptors in the intermediate layers. In these methods, lower-level convolutional filters (kernels) are used to detect local visual patterns. Viewed as local detectors, these filters have a smaller receptive field and are densely applied on the entire image. Compared with the global FC feature, local detectors are more robust to image transformations such as truncation and occlusion, in ways that are similar to the local invariant detectors (Section 3.2).

Local descriptors are tightly coupled with these intermediate local detectors, i.e., they are the responses of the input image to these convolution operations. In other words, after the convolutions, the resulting activation maps can be viewed as a feature ensemble, which is called the "column feature" in this survey. For example in AlexNet [6], there are $n = 96$ detectors (convolutional filters) in the 1st convolutional layer. These filters produces $n = 96$ heat maps of size 27×27 (after max pooling). Each pixel in the maps has a receptive field of 19×19 and records the response of the image w.r.t. the corresponding filter [9], [10], [134]. The column feature is therefore of size $1 \times 1 \times 96$ (Fig. 2) and can be viewed as a description of a certain patch in the original image. Each dimension of this descriptor denotes the level of activation of the corresponding detector and resembles the SIFT descriptor to some extent. The column feature initially appears in [133], where Razavian et al. first do max-pooling over regularly partitioned windows on the feature maps and then concatenate them across all filter responses, yielding column-like features. In [136], column features from multiple layers of the networks are concatenated, forming the "hypercolumn" feature.

4.1.3 Feature Encoding and Pooling

When column features are extracted, an image is represented by a set of descriptors. To aggregate these descriptors into a global representation, currently two strategies are adopted: encoding and direct pooling (Fig. 2).

Encoding. A set of column features resembles a set of SIFT features. So standard encoding schemes can be directly employed. The most commonly used methods are VLAD [15] and FV [14]. A brief review of VLAD and FV can be seen in Section 3.3.2. A milestone work is [9], in which the column features are encoded into VLAD for the first time. This idea was later extended to CNN model fine-tuning [137]. The BoW encoding can also be leveraged, as the case in [138]. The column features within each layer are aggregated into a BoW vector which is then concatenated across the layers. An exception to these fix-length representations is [139], in which the column features are quantized with a codebook of size 25k and an inverted index is employed for efficiency.

Pooling. A major difference between the CNN column feature and SIFT is that the former has an explicit meaning in each dimension, i.e., the response of a particular region of the input image to a filter. Therefore, apart from the encoding schemes mentioned above, direct pooling techniques can produce discriminative features as well.

A milestone work in this direction consists in the Maximum activations of convolutions (MAC) proposed by Tolias et al. [10]. Without distorting or cropping images, MAC computes a global descriptor with a single forward pass. Specifically, MAC calculates the maximum value of each intermediate feature map and concatenates all these values within a convolutional layer. In its multi-region version, the integral image and an approximate maximum operator are used for fast computation. The regional MAC descriptors are subsequently sum-pooled along with a series of normalization and PCA-whitening operations [53]. We also note in this survey that several other works [133], [134], [140] also employ similar ideas with [10] in employing max or average pooling on the intermediate feature maps and that Razavian et al. [133] are the first. It has been observed that the last convolutional layer (e.g., pool5 in VGGNet), after pooling usually yields superior accuracy to the FC descriptors and the other convolutional layers [134].

Apart from direct feature pooling, it is also beneficial to assign some specific weights to the feature maps within each layer before pooling. In [140], Babenko et al. propose the injection of the prior knowledge that objects tend to be located toward image centers, and impose a 2-D Gaussian mask on the feature maps before sum pooling. Xie et al. [141] improve the MAC representation [10] by propagating the high-level semantics and spatial context to low-level neurons for improving the descriptive ability of these bottom-layer activations. With a more general weighting strategy, Kalantidis et al. [16] perform both feature map-wise and channel-wise weighing, which aims to highlight the highly active spatial responses while reducing burstiness effects.

4.2 Image Retrieval with Fine-Tuned CNN Models

Although pre-trained CNN models have achieved impressive retrieval performance, a hot topic consists in fine-tuning the CNN model on specific training sets. When a fine-tuned CNN model is employed, the image-level descriptor is usually generated in an end-to-end manner, i.e., the network will produce a final visual representation without additional explicit encoding or pooling steps.

4.2.1 Datasets for Fine-Tuning

The nature of the datasets used in fine-tuning is the key to learning discriminative CNN features. ImageNet [23] only provides images with class labels. So the pre-trained CNN model is competent in discriminating images of different

TABLE 3
Statistics of Instance-Level Datasets Having Been Used in Fine-Tuning

name	# images	# classes	content
Landmarks [8]	213,678	672	Landmark
3D Landmark [24]	163,671	713	Landmark
Tokyo TM [137]	112,623	n.a	Landmark
MV RGB-D [142]	250,000	300	House. object
Product [143]	101,945×2	n.a	Furniture

object/scene classes, but may be less effective to tell the difference between images that fall in the same class (e.g., architecture) but depict different instances (e.g., "Eiffel Tower" and "Notre-Dame"). Therefore, it is important to fine-tune the CNN model on task-oriented datasets.

The datasets having been used for fine-tuning in recent years are shown in Table 3. Buildings and common objects are the focus. The milestone work on fine-tuning is [8]. It collects the *Landmarks dataset* by a semi-automated approach: automated searching for the popular landmarks in Yandex search engine, followed by a manual estimation of the proportion of relevant image among the top ranks. This dataset contains 672 classes of various architectures, and the fine-tuned network produces superior features on landmark related datasets such as Oxford5k [12] and Holidays [13], but has decreased performance on Ukbench [11] where common objects are presented. Babenko et al. [8] have also fine-tuned CNNs on the *Multi-view RGB-D dataset* [142] containing turntable views of 300 household objects, in order to improve performance on Ukbench. The Landmark dataset is later used by Gordo et al. [17] for fine-tuning, after an automatic cleaning approach based on SIFT matching. In [24], Radenović et al. employ the retrieval and Structure-From-Motion methods to build *3D landmark* models so that images depicting the same architecture can be grouped. Using this labeled dataset, the linear discriminative projections (denoted as L_w in Table 5) outperform the previous whitening technique [53]. Another dataset called *Tokyo Time Machine* is collected using Google Street View Time Machine which provides images depicting the same places over time [137]. While most of the above datasets focus on landmarks, Bell et al. [143] build a *Product dataset* consisting of furniture by developing a crowd-sourced pipeline to draw connections between in-situ objects and the corresponding products. It is also feasible to fine-tune on the query sets suggested in [144], but this method may not be adaptable to new query types.

4.2.2 Networks in Fine-Tuning

The CNN architectures used in fine-tuning mainly fall into two types: the classification-based network and the verification-based network. The classification-based network is trained to classify architectures into pre-defined categories. Since there is usually no class overlap between the training set and the query images, the learned embedding e.g., FC6 or FC7 in AlexNet, is used for Euclidean distance based retrieval. This train/test strategy is employed in [8], in which the last FC layer is modified to have 672 nodes corresponding to the number of classes in the Landmark dataset.

The verification network may either use a siamese network with pairwise loss or use a triplet loss and has been more widely employed for fine-tuning. A standard siamese network based on AlexNet and the contrastive loss is employed in [143]. In [24], Radenović et al. propose to replace the FC layers with a MAC layer [10]. Moreover, with the 3D architecture models built in [24], training pairs can be mined. Positive image pairs are selected based on the number of co-observed 3D points (matched SIFT features), while hard negatives are defined as those with small distances in their CNN descriptors. These image pairs are fed into the siamese network, and the contrastive loss is calculated from the ℓ_2 normalized MAC features. In a concurrent work to [24], Gordo et al. [17] fine-tune a triplet-loss network and a region proposal network on the Landmark dataset [8]. The superiority of [17] consists in its localization ability, which excludes the background in feature learning and extraction. In both works, the fine-tuned models exhibit state-of-the-art accuracy on landmark retrieval datasets including Oxford5k, Paris6k and Holidays, and also good generalization ability on Ukbench (Table 5). In [137], a VLAD-like layer is plugged in the network at the last convolutional layer which is amenable to training via back-propagation. Meanwhile, a new triplet loss is designed to make use of the weakly supervised Google Street View Time Machine data.

4.3 Hybrid CNN-Based Methods

For the hybrid methods, multiple network passes are performed. A number of image patches are generated from an input image, which are fed into the network for feature extraction before an encoding/pooling stage. Since the manner of "detector + descriptor" is similar to SIFT-based methods, we call this method type "hybrid". It is usually less efficient than the single-pass methods.

4.3.1 Feature Extraction

In hybrid methods, the feature extraction process consists of patch detection and description steps. For the first step, the literature has seen three major types of region detectors. The first is grid image patches. For example, in [22], a two-scale sliding window strategy is employed to generate patches. In [7], the dataset images are first cropped and rotated, and then divided into patches of different scales, the union of which covers the whole image. The second type is invariant keypoint/region detectors. For instance, the difference of Gaussian feature points are used in [145]; the MSER region detector is leveraged in [146]. Third, region proposals also provide useful information on the locations of the potential objects. Mopuri et al. [147] employ selective search [148] to generate image patches, while EdgeBox [149] is used in [150]. In [144], the region proposal network (RPN) [151] is applied to locate the potential objects in an image.

The use of CNN as region descriptors is validated in [146], showing that CNN is superior to SIFT in image matching except on blurred images. Given the image patches, the hybrid CNN method usually employs the FC or pooled intermediate CNN features. Examples using the FC descriptors include [7], [22], [147], [152]. In these works, the 4,096-dim FC features are extracted from the multi-scale image regions [7], [22], [152] or object proposals [147]. On the other hand, Razavian et al. [133] also uses the intermediate descriptors after max-pooling as region descriptors.

The above methods use pre-trained models for patch feature extraction. Based on the hand-crafted detectors, patch descriptors can also be learned through CNN in either supervised [153] or unsupervised manner [145], which improves over the previous works on SIFT descriptor learning [34], [45]. Yi et al. [154] further propose an end-to-end learning method integrating region detector, orientation estimator and feature descriptor in a single pipeline.

4.3.2 Feature Encoding and Indexing

The encoding/indexing procedure of hybrid methods resembles SIFT-based retrieval, e.g., VLAD/FV encoding under a small codebook or the inverted index under a large codebook.

The VLAD/FV encoding, such as [22], [147], follow the standard practice in the case of SIFT features [14], [15], so we do not detail here. On the other hand, several works exploit the inverted index on the patch-based CNN features [139], [155], [156]. Again, standard techniques in SIFT-based methods such as HE are employed [156]. Apart from the above-mentioned strategies, we notice that several works [7], [133], [152] extract several region descriptors per image to do a many-to-many matching, called "spatial search" [7]. This method improves the translation and scale invariance of the retrieval system but may encounter efficiency problems. A reverse strategy to applying encoding on top of CNN activations is to build a CNN structure (mainly consisting of FC layers) on top of SIFT-based representations such as FV. By training a classification model on natural images, the intermediate FC layer can be used for retrieval [157].

4.4 Discussions

4.4.1 Relationship between SIFT- and CNN-Based Methods

In this survey, we categorize current literature into six fine-grained classes. The differences and some representative works of the six categories are summarized in Tables 1 and 5. Our observation goes below.

First, the hybrid method can be viewed as a transition zone from SIFT- to CNN-based methods. It resembles the SIFT-based methods in all the aspects except that it extracts CNN features as the local descriptor. Since the network is accessed multiple times during patch feature extraction, the efficiency of the feature extraction step may be compromised.

Second, the single-pass CNN methods tend to combine the individual steps in the SIFT-based and hybrid methods. In Table 5, the "pre-trained single-pass" category integrates the feature detection and description steps; in the "fine-tuned single-pass" methods, the image-level descriptor is usually extracted in an end-to-end mode, so that no separate encoding process is needed. In [17], a "PCA" layer is integrated for discriminative dimension reduction, making a further step towards end-to-end feature learning.

Third, fixed-length representations are gaining more popularity due to efficiency considerations. It can be obtained by aggregating local descriptors (SIFT or CNN) [9], [15], [18], [22], direct pooling [10], [147], or end-to-end feature computation [8], [17]. Usually, dimension reduction methods such as PCA can employed on top of the fixed-length representations, and ANN search methods such as PQ [15] or hashing [47] can be used for fast retrieval.

4.4.2 Hashing and Instance Retrieval

Hashing is a major solution to the approximate nearest neighbor problem. It can be categorized into locality sensitive hashing (LSH) [63] and learning to hash. LSH is data-independent and is usually outperformed by learning to hash, a data-dependent hashing approach. For learning to hash, a recent survey [67] categorizes it into quantization and pairwise similarity preserving. The quantization methods are briefly discussed in Section 3.3.2. For the pairwise similarity preserving methods, some popular hand-crafted methods include Spectral hashing [64], LDA hashing [158], etc.

Recently, hashing has seen a major shift from hand-crafted to supervised hashing with deep neural networks. These methods take the original image as input and produce a learned feature before binarization [159], [160]. Most of these methods, however, focus on class-level image retrieval, a different task with instance retrieval discussed in this survey. For instance retrieval, when adequate training data can be collected, such as architecture and pedestrians, the deep hashing methods may be of critical importance.

5 Experimental Comparisons

5.1 Image Retrieval Datasets

Five popular instance retrieval datasets are used in this survey. Statistics of these datasets can be accessed in Table 4.

Holidays [13] is collected by Jégou et al. from personal holiday albums, so most of the images are of various scene types. The database has 1,491 images composed of 500 groups of similar images. Each image group has 1 query, totaling 500 query images. Most SIFT-based methods employ the original images, except [32], [71] which manually rotate the images into upright orientations. Many recent CNN-based methods [16], [137], [140] also use the rotated version of Holidays. In Table 5, results of both versions of Holidays are shown (separated by "/"). Rotating the images usually brings 2-3 percent mAP improvement.

Ukbench [11] consists of 10,200 images of various content, such as objects, scenes, and CD covers. All the images are divided into 2,550 groups. Each group has four images depicting the same object/scene, under various angles, illuminations, translations, etc. Each image in this dataset is taken as the query in turn, so there are 10,200 queries.

Oxford5k [12] is collected by crawling images from Flickr using the names of 11 different landmarks in Oxford. A total of 5,062 images form the image database. The dataset defines five queries for each landmark by hand-drawn bounding boxes, so that 55 query Regions of Interest (ROI) exist in total. Each database image is assigned one of four labels, *good*, *OK*, *junk*, or *bad*. The first two labels are true matches to the query ROIs, while "bad" denotes the distractors. In junk images, less than 25 percent of the objects are visible, or they undergo severe occlusion or distortion, so these images have zero impact on retrieval accuracy.

Flickr100k [25] contains 99,782 high resolution images crawled from Flickr's 145 most popular tags. In literature, this dataset is typically added to Oxford5k to test the scalability of retrieval algorithms.

Paris6k [25] is featured by 6,412 images crawled from 11 queries on specific Paris architecture. Each landmark has five queries, so there are also 55 queries with bounding

TABLE 4
Statistics of Popular Instance-Level Datasets

name	# images	# queries	content
Holidays [13]	1,491	500	scene
Ukbench [11]	10,200	10,200	common objects
Paris6k [25]	6,412	55	buildings
Oxford5k [12]	5,062	55	buildings
Flickr100k [25]	99,782	-	from Flickr's popular tags

boxes. The database images are annotated with the same four types of labels as Oxford5k. Two major evaluation protocols exist for Oxford5k and Paris6k. For SIFT-based methods, the cropped regions are usually used as query. For CNN-based methods, some employ the full-sized query images [8], [137]; some follow the standard cropping protocol, either by cropping the ROI and feeding it into CNN [16] or extracting CNN features using the full image and selecting those falling in the ROI [144]. Using the full image may lead to mAP improvement. These protocols are used in Table 5.

5.2 Evaluation Metrics

Precision-Recall. Recall denotes the ratio of returned true matches to the total number or true matches in the database, while precision refers to the fraction of true matches in the returned images. Given a subset of n returned images, assuming there are n_p true matches among them, and a total of N_p true matches exist in the whole database, then recall@n ($r@n$) and precision@n ($p@n$) are calculated as $\frac{n_p}{N_p}$ and $\frac{n_p}{n}$, respectively. In image retrieval, given a query image and its rank list, a precision-recall curve can be drawn on the (precision, recall) points $(r@1, p@1), (r@2, p@2), \ldots, (r@N, p@N)$, where N is the number of images in the database.

Average Precision and Mean Average Precision. To more clearly record the retrieval performance, average precision (AP) is used, which amounts to the area under the precision-recall curve. Typically, a larger AP means a higher precision-recall curve and thus better retrieval performance. Since retrieval datasets typically have multiple query images, their respective APs are averaged to produce a final performance evaluation, i.e., the mean average precision (mAP). Conventionally, we use mAP to evaluate retrieval accuracy on the Oxford5k, Paris6k, and Holidays datasets.

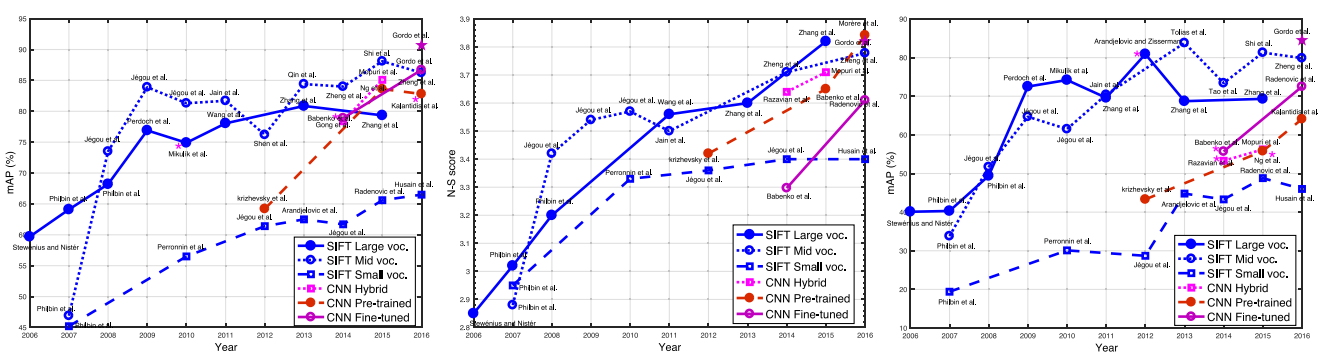
N-S Score. The N-S score is specifically used on the Ukbench dataset and is named after David **N**istér and Henrik **S**tewénius [11]. It is equivalent to precision@4 or recall@4 because every query in Ukbench has four true matches in the database. The N-S score is calculated as the average number of true matches in the top-4 ranks across all the rank lists.

5.3 Comparison and Analysis

5.3.1 Performance Improvement Over the Years

We present the improvement in retrieval accuracy over the past ten years in Fig. 6 and the numbers of some representative methods in Table 5. The results are computed using codebooks trained on independent datasets [13]. We can clearly observe that the field of instance retrieval has constantly been improving. The baseline approach (HKM) proposed over ten years ago only yields a retrieval accuracy of 59.7 percent, 2.85, 44.3 percent, 26.6, and 46.5 percent on Holidays, Ukbench, Oxford5k, Oxford5k+Flickr100k, and Paris6k, respectively. Starting from the baseline approaches [11], [12], methods using large codebooks improve steadily when more discriminative codebooks [71], spatial constraints [21], [82], and complementary descriptors [72], [163] are introduced. For medium-sized codebooks, the most significant accuracy advance has been witnessed in the years 2008-2010 with the introduction of Hamming Embedding [13], [85] and its improvements [76], [85], [90]. From then on, major improvements come from the strength of feature fusion [31], [163], [135] with the color and CNN features, especially on the Holidays and Ukbench datasets.

On the other hand, CNN-based retrieval models have quickly demonstrated their strengths in instance retrieval. In the year 2012 when the AlexNet [6] was introduced, the performance of the off-the-shelf FC features is still far from satisfactory compared with SIFT models during the same period. For example, the FC descriptor of AlexNet pre-trained on ImageNet yields 64.2, 3.42, and 43.3 percent in mAP, N-S score, and mAP, respectively, on the Holidays, Ukbench, and Oxford5k datasets. These numbers are lower than [82] by 13.85 percent, 0.14 on Holidays and Ukbench, respectively, and lower than [111] by 31.9 percent on Oxford5k. However, with the advance in CNN architectures and fine-tuning strategies, the performance of the CNN-based



(a) Holidays, *mAP* (b) Ukbench, *N-S* (c) Oxford5k, *mAP*

Fig. 6. The state of the art over the years on the (a) Holidays, (b) Ukbench, and (c) Oxford5k datasets. Six fine-grained categories are summarized (see Section 2). For each year, the best accuracy of each category is reported. For the compact representations, results of 128-bit vectors are preferentially selected. The purple star denotes the results produced by 2,048-dim vectors [17], the best performance in fine-tuned CNN methods. Methods with a pink asterisk denote using rotated images on Holidays, full-sized queries on Oxford5k, or spatial verification and QE on Oxford5k (see Table 5).

TABLE 5
Performance Summarization of Some Representative Methods of the Six Categories on the Benchmarks

	method type	methods	detector	descriptor	encoding	indexing	voc	dim	Holidays	Ukbench	Oxford5k	+100k	Paris6k	mem/img
SIFT-based	Large Voc.	HKM [11]	MSER	SIFT	hier. soft, BoW	inv. index	1M		59.7	2.85	44.3†	26.6†	46.5†	9.8kb
		AKM [12]	hes-aff	SIFT	hard, BoW	inv. index	1M		64.1†	3.02†	49.3	34.3	50.2†	9.8kb
		Fine Voc. [71]	hes-aff	SIFT	alt. word, BoW	inv. index	16M		-/74.9 (75.8)	-	74.2 (84.9)	67.4 (79.5)	74.9 (82.4)	9.8kb
		Three Things [33]	hes-aff	rootSIFT	hard, BoW	inv. index	1M		-	-	80.9*	72.2*	76.5*	22.0kb
		Co-index [72]	DoG	SIFT, CNN	hard, BoW	co-index	1M		80.86	3.60	68.72	-	-	21.6kb
	Mid Voc.	HE+WGC [13], [85]	hes-aff	SIFT	hard, BoW, HE	inv. index	20k		81.3 (84.2)	3.42 (3.55)	61.5 (74.7)	51.6 (68.7)	-	36.8kb
		Burst [76]	hes-aff	SIFT	burst, BoW, HE	inv. index	20k		83.9 (84.8)	3.54 (3.64)	64.7 (68.5)	58§ (63§)	62.8†	36.8kb
		Q.ada [91]	hes-aff	rootSIFT	MA, BoW, HE	inv. index	20k		78.0	-	82.1	72.8	73.6	36.8kb
		ASMK [20]	hes-aff	rootSIFT	MA, BoW, agg.HE	inv. index	65k		81.0	-	80.4	75.0 (85.0)	77.0	43.2kb
		c-MI [31]	hes-aff	rootSIFT, HS	MA, BoW, HE	2D index	20k×200		84.0	3.71	58.2†	35.2†	55.1†	45.2kb
	Small Voc.	VLAD [15]	hes-aff	SIFT	VLAD	PCA, PQ	64	4,096	55.6	3.18†	37.8	27.2†	38.6†	16kb
		FV [47], [52]	hes-aff	PCA-SIFT	FV, pw.	LSH, SH	64	4,096	59.5	3.35	41.8	33.1†	43.0	16kb
		All A. VLAD [54]	hes-aff	SIFT	Improved VLAD	PCA	64	128	62.5	-	44.8	-	-	0.5kb
		NE [53]	hes-aff	rootSIFT	NE, multi. voc, VLAD	PCA_w	4×256	128	61.4	3.36	-	-	-	0.5kb
		Triangulation [18]	hes-aff	rootSIFT	triang+democ	PCA, pw.	16	128	61.7	3.40	43.3	35.3	-	0.5kb
CNN-based	Hybrid	Off the Shelf [7]	den. patch	OFeat 1st FC	-	-	-	-	84.3	3.64	-/68.0	-	-/79.5	4-15kb
		MSS [133]	den. patch	vgg conv5	MP	-	-	-	88.1	3.72	-/84.4	-	-/85.3	16kb
		CKN [153]	hes-aff	CKN	VLAD, power	-	256	65k	79.3	3.76	56.5	-	-	256kb
		MOP [22]	den. patch	alex FC7	multi-scale VLAD	PCA_w	100	2,048	80.2	-	-	-	-	8kb
		OLDFP[147]	sel. search	alex FC7	MP	ITQ [161]	-	512	88.5	3.81	60.7	-	66.2	2kb
	Pre-trained single-pass	BLCF [139]	vgg16, conv5		hard, BoW	inv. index	25k		-	-	73.9 (78.8)	59.3 (65.1)	82.0 (84.8)	0.67kb
		R-MAC [10]	vgg16, conv5		region MP, SP	PCA_w	-	512	85.2/86.9	-	66.9 (77.3)	61.6 (73.2)	83.0 (86.5)	1kb
		CroW [16]	vgg16, conv5		cross-dim pool.	PCA_w	-	512	-/85.1	-	70.8 (74.9)	65.3 (70.6)	79.7 (84.8)	2kb
		SPoC [140]	vgg16, conv5		SP, center prior	PCA_w	-	256	-/80.2	3.65	53.1‡/58.9	50.1‡/57.8	-	1kb
		VLAD-CNN [9]	google, various incept.		intra-norm, VLAD	PCA	100	128	83.6	-	-/55.8	-	-/58.3	0.5kb
	Fine-tuned single-pass	Faster R-CNN [144]	vgg16-reg-query, conv5		SP or MP	-	-	512	-	-	71.0‡(78.6)	-	79.8‡(84.2)	2kb
		Neural Codes [8]	alexnet-classification loss-Landmark, FC6			PCA	-	128	-/78.9	3.29	-/55.7	-/52.3	-	0.5kb
		NetVLAD [137]	vgg16-trip. loss-Tokyo TM, VLAD layer			PCA_w	64	512	81.7/86.1	-	65.6/67.6	-	73.4/74.9	8kb
		SiaMAC [24]	vgg16-pair loss-3D Landmark, MAC layer			L_w	-	512	-/82.5	3.61†	77.0 (82.9)	69.2 (77.9)	83.8 (85.6)	2kb
		Deep Retrieval [17], [162]	vgg16-trip. loss-cleaned Landmark, PCA layer used				-	512	86.7/89.1	3.62	83.1 (89.1)	78.6 (87.3)	87.1 (91.2)	2kb
			ResNet101-trip. loss-cleaned Landmark, PCA layer used				-	2,048	90.3/94.8	3.84	86.1 (90.6)	82.8 (89.4)	94.5 (96.0)	8kb

*"+100k" –> the addition of Flickr100k into Oxford5k. "pw." –> power low normalization [14]. "MP" –> max pooling. "SP" –> sum pooling. * and parentheses –> results are obtained with post-processing steps such as spatial verification or QE. § –> numbers are estimated from the curves. † numbers are reported by our implementation. For Holidays, results using the rotated images are presented after "/". For Oxford5k (+100k) and Paris6k, results using the full-sized queries are shown after "/". ‡ –> the full query image is fed into the network, but only the features whose centers fall into the query region of interest are aggregated. Note that in many fixed-length representations, ANN algorithms such as PQ are not used to report the results, but ANN can be readily applied after PCA during indexing.*

TABLE 6
A Summary of Efficiency and Accuracy Comparison between Different Categories

method type	efficiency				accuracy	
	feat. ext.	retr.	mem.	train	generic	specific
SIFT large voc.	fair	high	fair	fair	high	fair
SIFT mid voc.	fair	low	low	fair	high	high
SIFT small voc.	fair	high	high	high	low	Low
CNN hybrid	low	varies	varies	varies	fair	fair
CNN pre-trained	high	high	high	high	high	fair
CNN fine-tuned	high	high	high	low	high	high

Note: feature extraction time is estimated using CPUs and GPUs (as is usually done) for SIFT and CNN, resp. *When using GPUs for SIFT extraction, the efficiency could be high as well.*

methods is improving fast, being competitive on the Holidays and Ukbench datasets [17], [164], and slightly lower on Oxford5k but with much smaller memory cost [24].

5.3.2 Accuracy Comparisons

The retrieval accuracy of different categories on different datasets can be viewed in Fig. 6, Tables 5 and 6. From these results, we arrive at three observations.

First, among the SIFT-based methods, those with medium-sized codebooks [13], [31], [19] usually lead to superior (or competitive) performance, while those based on small codebook (compact representations) [15], [18], [56] exhibit inferior accuracy. On the one hand, the visual words in the medium-sized codebooks lead to relatively high matching recall due to the large Voronoi cells. The further integration of HE methods largely improves the discriminative ability, achieving a desirable trade-off between matching recall and precision. On the other hand, although the visual words in small codebooks have the highest matching recall, their discriminative ability is not significantly improved due to the aggregation procedure and the small dimensionality. So its performance can be compromised.

Second, among the CNN-based categories, the fine-tuned category [8], [17], [24] is advantageous in specific tasks (such as landmark/scene retrieval) which have similar data distribution with the training set. While this observation is within expectation, we find it interesting that the fine-tuned model proposed in [17] yields very competitive performance on generic retrieval (such as Ukbench) which has distinct data distribution with the training set. In fact, Babenko et al. [8] show that the CNN features fine-tuned on Landmarks compromise the accuracy on Ukbench. The generalization ability of [17] could be attributed to the effective training of the region proposal network. In comparison, using pre-trained models may exhibit high accuracy on Ukbench, but only yields moderate performance on landmarks. Similarly, the hybrid methods have fair performance on all the tasks, when it may still encounter efficiency problems [7], [152].

Third, comparing all the six categories, the "CNN fine-tuned" and "SIFT mid voc." categories have the best overall accuracy, while the "SIFT small voc." category has a relatively low accuracy.

5.3.3 Efficiency Comparisons

Feature Computation Time. For the SIFT-based methods, the dominating step is local feature extraction. Usually, it takes 1-2s for a CPU to extract the Hessian-Affine region based SIFT descriptors for a 640×480 image, depending on the complexity (texture) of the image. For the CNN-based method, it takes 0.082 and 0.347 s for a single forward pass of a 224×224 and 1,024×768 image through VGG16 on a TitanX card, respectively. It is reported in [17] that four images (with largest side of 724 pixels) can be processed in 1 second. The encoding (VLAD or FV) time of the pre-trained column features is very fast. For the CNN Hybrid methods, extracting CNN features out of tens of regions may take seconds. Overall speaking, the CNN pre-trained and fine-tuned models are efficient in feature computation using GPUs. Yet it should be noted that when using GPUs for SIFT extraction, high efficiency could also be achieved.

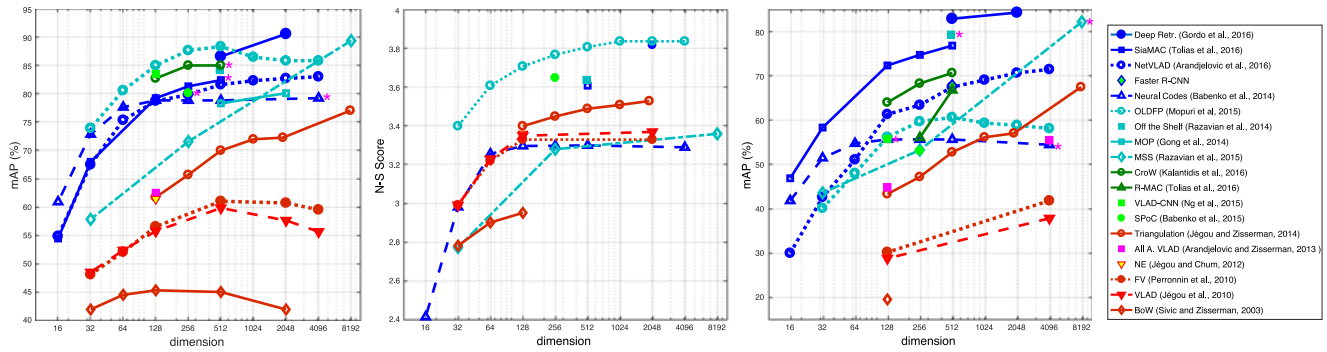
Retrieval Time. The efficiency of nearest neighbor search is high for "SIFT large voc.", "SIFT small voc.", "CNN pre-trained" and "CNN fine-tuned", because the inverted lists are short for a properly trained large codebook, and because the latter three have a compact representation to be accelerated by ANN search methods like PQ [61]. Efficiency for the medium-sized codebook is low because the inverted list contains more postings compared to a large codebook, and the filtering effect of HE methods can only correct this problem to some extent. The retrieval complexity for hybrid methods, as mentioned in Section 4.3, may suffer from the expensive many-to-many matching strategy [7], [133], [152].

Training Time. Training a large or medium-sized codebook usually takes several hours with AKM or HKM. Using small codebooks reduces the codebook training time. For the fine-tuned model, Gordo et al. [17] report using five days on a K40 GPU for the triplet-loss model. It may take less time for the siamese [24] or the classification models [8], but should still much longer than SIFT codebook generation. Therefore, in terms of training, those using direct pooling [10], [134] or small codebooks [15], [9] are more time efficient.

Memory Cost. Table 5 and Fig. 8 show that the SIFT methods with large codebooks and the compact representations are both efficient in memory cost. But the compact representations can be compressed into compact codes [53] using PQ or other competing quantization/hashing methods, so their memory consumption can be further reduced. In comparison, the methods using medium-sized codebooks are the most memory-consuming because the binary signatures should be stored in the inverted index. The hybrid methods somehow have mixed memory cost because the many-to-many strategy requires storing a number of region descriptors per image [7], [152] while some others employ efficient encoding methods [22], [147].

Spatial Verification and Query Expansion. Spatial verification which provides refined rank lists is often used in conjunction with QE. The RANSAC verification proposed in [12] has a complexity of $\mathcal{O}(z^2)$, where z is the number of matched features. So this method is computationally expensive. The ADV approach [113] is less expensive with $\mathcal{O}(z \log z)$ complexity due to its ability to avoid unrelated Hough votes. The most efficient methods consist in [112], [115] which has a complexity of $\mathcal{O}(z)$, and [115] further outputs the transformation and inliers for QE.

From the perspective of query expansion, since new queries are issued, search efficiency is compromised. For example, AQE [100] almost doubles the search time due to



(a) Holidays, *mAP* (b) Ukbench, *N-S* (c) Oxford5k, *mAP* (d) Legend

Fig. 7. The impact of feature dimension on retrieval accuracy. Compact (fixed-length) representations are shown, i.e., SIFT small voc., hybrid CNN methods, pre-trained CNN methods, and fine-tuned CNN methods. Curves with a pink asterisk on the end indicates using rotated images or full-sized queries on Holidays and Oxford5k (see Table 5), *resp.*

the new query. For the recursive AQE and the scale-band recursive QE [100], the search time is much longer because several new searches are conducted. For other QE variants [33], [101], the proposed improvements only add marginal cost compared to performing another search, so their complexity is similar to basic QE methods.

5.3.4 Important Parameters

We summarize the impact of codebook size on SIFT methods using large/medium-sized codebooks, and the impact of dimensionality on compact representations including SIFT small codebooks and CNN-based methods.

Codebook Size. The mAP results on Oxford5k are drawn in Fig. 9, and methods using large/medium-sized codebooks are compared. Two observations can be made. First, mAP usually increases with the codebook size but may reach saturation when the codebook is large enough. This is because a larger codebook improves the matching precision, but if it is too large, matching recall is lower, leading to saturated or even compromised performance [12]. Second, methods using the medium-sized codebooks have more stable performance when codebook size changes. This can be attributed to HE [13], which contributes more for a smaller codebook, compensating the lower baseline performance.

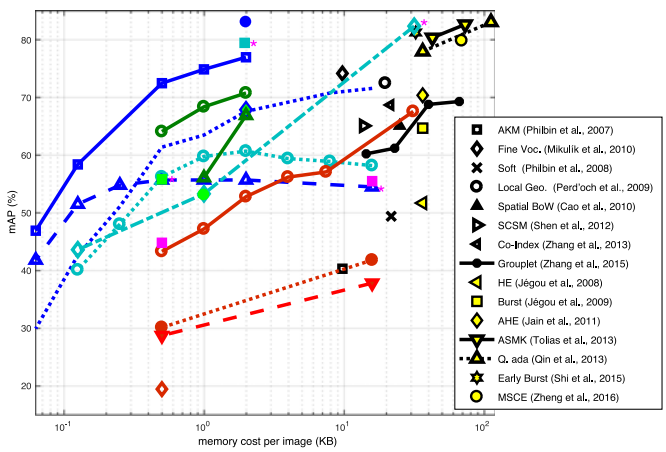


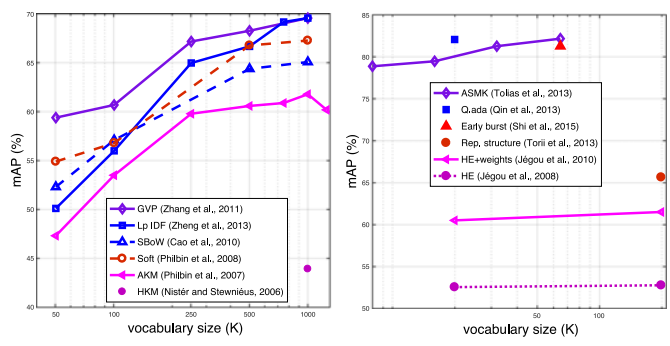
Fig. 8. Memory cost versus retrieval accuracy on Oxford5k. In the legend, the first 8 methods are based on large codebooks, while the last 7 use medium-sized codebooks. This figure shares the same legend with Fig. 7c except the newly added numbers (in black).

Dimensionality. The impact of dimensionality on compact vectors is presented in Fig. 7. Our finding is that the retrieval accuracy usually remains stable under larger dimensions, and drops quickly when the dimensionality is below 256 or 128. Our second finding favors the methods based on region proposals [17], [147]. These methods demonstrate very competitive performance under various feature lengths, probably due to their superior ability in object localization.

5.3.5 Discussions

We provide a brief discussion on when to use CNN over SIFT and the other way around. The above discussions provide comparisons between the two features. On the one hand, CNN-based methods with fixed-length representations have advantages in nearly all the benchmarking datasets. Specifically, in two cases, CNN-based methods can be assigned with higher priority. First, for specific object retrieval (e.g., buildings, pedestrians) when sufficient training data is provided, the ability of CNN embedding learning can be fully utilized. Second, for common object retrieval or class retrieval, the pre-trained CNN models are competitive.

On the other hand, despite the usual advantages of CNN-based methods, we envision that the SIFT feature still has merits in some cases. For example, when the query or some target images are gray-scale, CNN may be less effective than SIFT because SIFT is computed on gray-scale images without resorting to color information. A similar



(a) SIFT Large voc. (b) SIFT Mid voc.

Fig. 9. The impact of codebook size on SIFT-based methods using (a) large codebooks and (b) medium-sized codebooks on the Oxford5k dataset.

situation involves when object color change is highly intense. In another example, for small object retrieval or when the queried object undergoes severe occlusions, the usage of local features like SIFT is favored. In applications like book/CD cover retrieval, we can also expect good performance out of SIFT due to the rich textures.

6 Future Research Directions

6.1 Towards Generic Instance Retrieval

A critical direction is to make the search engine applicable to generic search purpose. Towards this goal, two important issues should be addressed. First, large-scale instance-level datasets are to be introduced. While several instance datasets have been released as shown in Table 3, these datasets usually contain a particular type of instances such as landmarks or indoor objects. Although the RPN structure used by Gordo et al. [17] has proven competitive on Ukbench in addition to the building datasets, it remains unknown if training CNNs on more generic datasets will bring further improvement. Therefore, the community is in great need of large-scale instance-level datasets or efficient methods for generating such a dataset in either a supervised or unsupervised manner.

Second, designing new CNN architectures and learning methods are important in fully exploiting the training data. Previous works employ standard classification [8], pairwise-loss [24] or Triplet-loss [17], [165] CNN models for fine-tuning. The introduction of Faster R-CNN to instance retrieval is a promising starting point towards more accurate object localization [17]. Moreover, transfer learning methods are also important when adopting a fine-tuned model in another retrieval task [166].

6.2 Towards Specialized Instance Retrieval

To the other end, there are also increasing interests in specialized instance retrieval. Examples include place retrieval [167], pedestrian retrieval [168], vehicle retrieval [169], logo retrieval [77], etc. Images in these tasks have specific prior knowledge that can be made use of. For example in pedestrian retrieval, the recurrent neural network (RNN) can be employed to pool the body part or patch descriptors. In vehicle retrieval, the view information can be inferred during feature learning, and the license plate can also provide critical information when being captured within a short distance.

Meanwhile, the process of training data collection can be further explored. For example, training images of different places can be collected via Google Street View [137]. Vehicle images can be accessed either through surveillance videos or internet images. Exploring new learning strategies in these specialized datasets and studying the transfer effect would be interesting. Finally, compact vectors or short codes will also become important in realistic retrieval settings.

7 Concluding Remarks

This survey reviews instance retrieval approaches based on the SIFT and CNN features. According to the codebook size, we classify the SIFT-based methods into three classes: using large, medium-sized, and small codebook. According to the feature extraction process, the CNN-based methods are categorized into three classes, too: using pre-trained models, fine-tuned models, and hybrid methods. A comprehensive survey of the previous approaches is conducted under each of the defined categories. The category evolution suggests that the hybrid methods are in the transition position between SIFT and CNN-based methods, that compact representations are getting popular, and that instance retrieval is working towards end-to-end feature learning and extraction.

Through the collected experimental results on several benchmark datasets, comparisons are made between the six method categories. Our findings favor the usage of CNN fine-tuning strategy, which yields competitive accuracy on various retrieval tasks and has advantages in efficiency. Future research may focus on learning more generic feature representations or more specialized retrieval tasks.

Acknowledgments

The authors would like to thank the pioneer researchers in image retrieval and other related fields. This work was partially supported by the Google Faculty Award and the Data to Decisions Cooperative Research Centre. This work was supported in part to Dr. Qi Tian by ARO grant W911NF-15-1-0290, Faculty Research Gift Awards by NEC Laboratories of America and Blippar, and National Science Foundation of China (NSFC) 61429201.

References

[1] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 776–789.
[2] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.
[3] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Comput. Vis.*, 2003, Art. no. 1470.
[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop Statist. Learn. Comput. Vis.*, 2004, pp. 1–22.
[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[7] A. Sharif Razavian , H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 512–519.
[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 584–599.
[9] J. Ng, F. Yang, and L. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2015, pp. 53–61.
[10] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. Int. Conf. Learn. Representations*, 2016.
[11] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2161–2168.
[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[13] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.
[14] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3304–3311.
[16] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 685–701.
[17] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 241–257.
[18] H. Jégou and A. Zisserman, "Triangulation embedding and democratic aggregation for image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3310–3317.
[19] M. Shi, Y. Avrithis, and H. Jégou, "Early burst detection for memory-efficient image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 605–613.
[20] G. Tolias, Y. Avrithis, and H. Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 1401–1408.
[21] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 809–816.
[22] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 392–407.
[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.
[24] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–20.
[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[26] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. Trends Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, 2008.
[27] P. R. Beaudet, "Rotationally invariant image operators," in *Proc. 4th Int. Joint Conf. Pattern Recognit*, 1978, pp. 579–583.
[28] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
[29] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 25–32.
[30] K. Mikolajczyk, et al., "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1/2, pp. 43–72, 2005.
[31] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1947–1954.
[32] M. Perdóch, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 9–16.
[33] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2911–2918.
[34] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, Aug. 2014.
[35] T. Ge, Q. Ke, and J. Sun, "Sparse-coded features for image retrieval," in *Proc. British Mach. Vis. Conf.*, 2013, pp. 1–11.
[36] R. Arandjelović and A. Zisserman, "Smooth object retrieval using a bag of boundaries," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 375–382.
[37] R. Sicre and T. Gevers, "DENSE sampling of features for image retrieval," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 3057–3061.
[38] W.-L. Zhao, H. Jégou, and G. Gravier, "Oriented pooling for dense and non-dense rotation-invariant features," in *Proc. 24th British Mach. Vis. Conf.*, 2013, pp. 99.1–99.11.
[39] A. Iscen, G. Tolias, P.-H. Gosselin, and H. Jégou, "A comparison of dense region detectors for image search and fine-grained classification," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2369–2381, Aug. 2015.
[40] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[41] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. II-506–II-513.
[42] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[43] L. Juan and O. Gwun, "A comparison of SIFT, PCA-SIFT and SURF," *Int. J. Image Process.*, vol. 3, no. 4, pp. 143–152, 2009.
[44] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[45] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Descriptor learning for efficient retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 677–691.
[46] F. Radenović, H. Jégou, and O. Chum, "Multiple measurements and joint dimensionality reduction for large scale image search with short vectors," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 587–590.
[47] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3384–3391.
[48] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Comput. Vis. Image Understanding*, vol. 117, no. 5, pp. 479–492, 2013.
[49] P.-H. Gosselin, N. Murray, H. Jégou, and F. Perronnin, "Revisiting the Fisher vector for fine-grained classification," *Pattern Recognit. Lett.*, vol. 49, pp. 92–98, 2014.
[50] R. G. Cinbis, J. Verbeek, and C. Schmid, "Approximate Fisher kernels of non-iid image models for image categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1084–1098, Jun. 2015.
[51] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. British Mach. Vis. Conf.*, 2011, pp. 76.1–76.12.
[52] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[53] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 774–787.
[54] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1578–1585.
[55] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 653–656.
[56] S. S. Husain and M. Bober, "Improving large-scale image retrieval through robust aggregation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, Doi: 10.1109/TPAMI.2016.2613873.
[57] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.
[58] G. Tolias, T. Furon, and H. Jégou, "Orientation covariant aggregation of local descriptors with embeddings," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 382–397.
[59] N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2473–2480.
[60] Z. Liu, S. Wang, and Q. Tian, "Fine-residual VLAD for image retrieval," *Neurocomputing*, vol. 173, pp. 1183–1191, 2016.
[61] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[62] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.
[63] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. Annu. ACM Symp. Theory Comput.*, 1998, pp. 604–613.
[64] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[65] E. Spyromitros-Xioufis, S. Papadopoulos, I. Y. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "A comprehensive study over VLAD and product quantization in large-scale image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1713–1728, Oct. 2014.
[66] M. Douze, H. Jégou, and F. Perronnin, "Polysemous codes," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 785–801.
[67] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, Doi: 10.1109/TPAMI.2017.2699960.

[68] M. Shi, T. Furon, and H. Jégou, "A group testing framework for similarity search in high-dimensional spaces," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 407–416.

[69] A. Iscen, M. Rabbat, and T. Furon, "Efficient large-scale similarity search using matrix factorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2073–2081.

[70] A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou, "Memory vectors for similarity search in high-dimensional spaces," *IEEE Trans. Big Data*, 2017, Doi: 10.1109/TBDATA.2017.2677964.

[71] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.

[72] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1673–1680.

[73] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1794–1801.

[74] Y. Cai, W. Tong, L. Yang, and A. G. Hauptmann, "Constrained keypoint quantization: Towards better bag-of-words model for large-scale multimedia retrieval," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2012, Art. no. 16.

[75] W. Zhou, Y. Lu, H. Li, and Q. Tian, "Scalar quantization for large scale image search," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 169–178.

[76] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1169–1176.

[77] J. Revaud, M. Douze, and C. Schmid, "Correlation-based burstiness for logo retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 965–968.

[78] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Lp-norm IDF for large scale image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1626–1633.

[79] M. Murata, H. Nagano, R. Mukai, K. Kashino, and S. Satoh, "BM25 with exponential IDF for instance search," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1690–1699, Oct. 2014.

[80] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 883–890.

[81] P. Turcot and D. G. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 2109–2116.

[82] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proc. IEEE Conf. Comput. Vis.*, 2011, pp. 209–216.

[83] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1794–1801.

[84] A. Babenko and V. Lempitsky, "The inverted multi-index," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3069–3076.

[85] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 316–336, 2010.

[86] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1741–1750.

[87] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 257–266, Jun. 2010.

[88] M. Jain, R. Benmokhtar, H. Jégou, and P. Gros, "Hamming embedding similarity-based image classification," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2012, Art. no. 19.

[89] G. Tolias and H. Jégou, "Visual query expansion with or without geometry: Refining local descriptors by feature aggregation," *Pattern Recognit.*, vol. 47, no. 10, pp. 3466–3476, 2014.

[90] M. Jain, H. Jégou, and P. Gros, "Asymmetric hamming embedding: Taking the best of our bits for large scale image search," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1441–1444.

[91] D. Qin, C. Wengert, and L. Gool, "Query adaptive similarity for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1610–1617.

[92] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 2–11, Jan. 2010.

[93] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[94] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.

[95] J. Van de Weijer, T. Gevers, and A. D. Bagdanov, "Boosting color saliency in image feature detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 150–156, Jan. 2006.

[96] C. Wengert, M. Douze, and H. Jégou, "Bag-of-colors for improved image search," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1437–1440.

[97] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 660–673.

[98] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao, "Visual reranking through weakly supervised multi-graph learning," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 2600–2607.

[99] R. Tao, A. W. Smeulders, and S.-F. Chang, "Attributes and categories for generic instance search from one example," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 177–186.

[100] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[101] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 889–896.

[102] J. Philbin, J. Sivic, and A. Zisserman, "Geometric latent dirichlet allocation on a matching graph for large-scale image datasets," *Int. J. Comput. Vis.*, vol. 95, no. 2, pp. 138–153, 2011.

[103] Q.-F. Zheng, W.-Q. Wang, and W. Gao, "Effective and efficient object-based image retrieval using visual phrases," in *Proc. ACM Int. Conf. Multimedia*, 2006, pp. 77–80.

[104] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li, "Descriptive visual words and visual phrases for image applications," in *Proc. ACM Int. Conf. Multimedia*, 2009, pp. 75–84.

[105] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1745–1752.

[106] Y. Jiang, J. Meng, and J. Yuan, "Randomized visual phrases for object search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3100–3107.

[107] L. Torresani, M. Szummer, and A. Fitzgibbon, "Learning query-dependent prefilters for scalable image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2615–2622.

[108] P. Koniusz, F. Yan, P.-H. Gosselin, and K. Mikolajczyk, "Higher-order occurrence pooling for bags-of-words: Visual concept detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 313–326, Feb. 2017.

[109] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[110] C. L. Zitnick, J. Sun, R. Szeliski, and S. Winder, "Object instance recognition using triplets of feature symbols," Microsoft Research, Cambridge, U.K., Tech. Rep. MSR-TR-200753, 2007.

[111] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3013–3020.

[112] Y. Avrithis and G. Tolias, "Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval," *Int. J. Comput. Vis.*, vol. 107, no. 1, pp. 1–19, 2014.

[113] X. Wu and K. Kashino, "Adaptive dither voting for robust spatial verification," in *Proc. IEEE Conf. Comput. Vis.*, 2015, pp. 1877–1885.

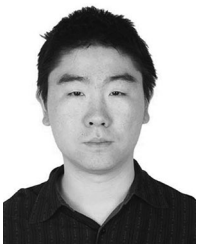
[114] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5153–5161.

[115] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 321–337.

[116] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 777–784.

[117] G. Awad, et al., "TRECVID 2016: Evaluating video search, video event detection, localization, and hyperlinking," in *Proc. 20th Int. Workshop Video Retrieval Eval.*, 2016, pp. 1–18.
[118] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 17–24.
[119] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, "Scalable triangulation-based logo recognition," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2011, Art. no. 20.
[120] S. Romberg and R. Lienhart, "Bundle min-hashing for logo recognition," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2013, pp. 113–120.
[121] C.-Z. Zhu, H. Jégou, and S. Satoh, "Query-adaptive asymmetrical dissimilarities for visual object retrieval," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 1705–1712.
[122] Z. Chen, W. Zhang, B. Hu, X. Can, S. Liu, and D. Meng, "Retrieving objects by partitioning," *IEEE Trans. Big Data*, vol. 3, no. 1, pp. 44–54, Jan.-Mar. 2017.
[123] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[124] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
[125] C. Szegedy, et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.
[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.
[127] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[128] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. British Mach. Vis. Conf.*, 2014, pp. 1–12.
[129] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 487–495.
[130] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, Sep. 2016.
[131] V. Chandrasekhar, J. Lin, O. Morère, H. Goh, and A. Veillard, "A practical guide to CNNs and Fisher vectors for image instance retrieval," *Signal Process.*, vol. 128, pp. 426–439, 2016.
[132] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2014.
[133] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," in *Proc. Int. Conf. Learn. Representations Workshops*, 2015.
[134] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in CNN feature transfer," *arXiv:1604.00133*, 2016.
[135] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *Int. J. Comput. Vis.*, vol. 120, pp. 1–13, 2016.
[136] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 447–456.
[137] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 5297–5307.
[138] P. Kulkarni, J. Zepeda, F. Jurie, P. Perez, and L. Chevallier, "Hybrid multi-layer deep CNN/aggregator feature for image classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 1379–1383.
[139] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marqués, and X. Giró-i Nieto, "Bags of local convolutional features for scalable instance search," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 327–331.
[140] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1269–1277.
[141] L. Xie, L. Zheng, J. Wang, A. Yuille, and Q. Tian, "Interactive: Inter-layer activeness propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 270–279.
[142] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1817–1824.
[143] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 98.
[144] A. Salvador, X. Giró-i Nieto, F. Marqués, and S. Satoh, "Faster R-CNN features for instance search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2016, pp. 394–401.
[145] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 4353–4361.
[146] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: A comparison to sift," *arXiv:1405.5769*, 2014.
[147] K. Mopuri and R. Babu, "Object level deep feature pooling for compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 201, pp. 62–70.
[148] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, pp. 154–171, 2013.
[149] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
[150] T. Uricchio, M. Bertini, L. Seidenari, and A. Bimbo, "Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 1020–1026.
[151] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[152] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are ONE," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 3–10.
[153] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 91–99.
[154] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–483.
[155] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "Deepindex for accurate and efficient image retrieval," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 43–50.
[156] Y. Li, X. Kong, L. Zheng, and Q. Tian, "Exploiting hierarchical activations of neural network for image retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 132–136.
[157] F. Perronnin and D. Larlus, "Fisher vectors meet neural networks: A hybrid classification architecture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3743–3752.
[158] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.
[159] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2015, pp. 27–35.
[160] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1556–1564.
[161] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 817–824.
[162] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *arXiv:1610.07940*, 2016.
[163] S. Zhang, X. Wang, Y. Lin, and Q. Tian, "Cross indexing with grouplets," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1969–1979, Nov. 2015.
[164] O. Morère, A. Veillard, J. Lin, J. Petta, V. Chandrasekhar, and T. Poggio, "Group invariant deep representations for image instance retrieval," in *Proc. AAAI Symp. Sci. Intell.*, 2017.
[165] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[166] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[167] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1808–1817.
[168] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1116–1124.
[169] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2016, pp. 1–6.



Liang Zheng received the PhD degree in electronic engineering from Tsinghua University, China, in 2015, and the BE degree in life science from Tsinghua University, China, in 2010. He was a postdoc researcher in University of Texas, San Antonio, Texas. He is now a postdoc researcher in the Center of Artificial Intelligence, University of Technology Sydney, Australia. His research interests include image retrieval, person re-identification, and deep learning.

Yi Yang received the PhD degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently a professor with University of Technology Sydney, Australia. He was a post-doctoral research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. His current research interest includes machine learning and its applications to multimedia content analysis and computer vision.

Qi Tian (M'96-SM'03-F'16) received the BE degree in electronic engineering from Tsinghua University, China, the MS degree in electrical and computer engineering from Drexel University, and the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1992, 1996, and 2002, respectively. He is currently a professor with the Department of Computer Science, University of Texas, San Antonio (UTSA). His research interests include multimedia information retrieval and computer vision. He is fellow of the IEEE.

BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition

Boyan Zhou[1] Quan Cui[1,2] Xiu-Shen Wei[1*] Zhao-Min Chen[1,3]
[1]Megvii Technology [2]Waseda University [3]Nanjing University

Abstract

*Our work focuses on tackling the challenging but natural visual recognition task of long-tailed data distribution (*i.e.*, a few classes occupy most of the data, while most classes have rarely few samples). In the literature, class re-balancing strategies (*e.g.*, re-weighting and re-sampling) are the prominent and effective methods proposed to alleviate the extreme imbalance for dealing with long-tailed problems. In this paper, we firstly discover that these re-balancing methods achieving satisfactory recognition accuracy owe to that they could significantly promote the classifier learning of deep networks. However, at the same time, they will unexpectedly damage the representative ability of the learned deep features to some extent. Therefore, we propose a unified Bilateral-Branch Network (BBN) to take care of both representation learning and classifier learning simultaneously, where each branch does perform its own duty separately. In particular, our BBN model is further equipped with a novel cumulative learning strategy, which is designed to first learn the universal patterns and then pay attention to the tail data gradually. Extensive experiments on four benchmark datasets, including the large-scale iNaturalist ones, justify that the proposed BBN can significantly outperform state-of-the-art methods. Furthermore, validation experiments can demonstrate both our preliminary discovery and effectiveness of tailored designs in BBN for long-tailed problems. Our method won the first place in the iNaturalist 2019 large scale species classification competition, and our code is open-source and available at* https://github.com/Megvii-Nanjing/BBN.

1. Introduction

With the advent of research on deep Convolutional Neural Networks (CNNs), the performance of image classification has witnessed incredible progress. The success is undoubtedly inseparable to available and high-quality large-scale datasets, *e.g.*, ImageNet ILSVRC 2012 [24], MS COCO [18] and Places Database [37], *etc*. In contrast with these visual recognition datasets exhibiting roughly uniform distributions of class labels, real-world datasets always have skewed distributions with *a long tail* [15, 26], *i.e.*, a few classes (a.k.a. *head class*) occupy most of the data, while most classes (a.k.a. *tail class*) have rarely few samples, cf. Figure 1. Moreover, more and more long-tailed datasets reflecting the realistic challenges are constructed and released by the computer vision community in very recent years, *e.g.*, iNaturalist [6], LVIS [10] and RPC [29]. When dealing with

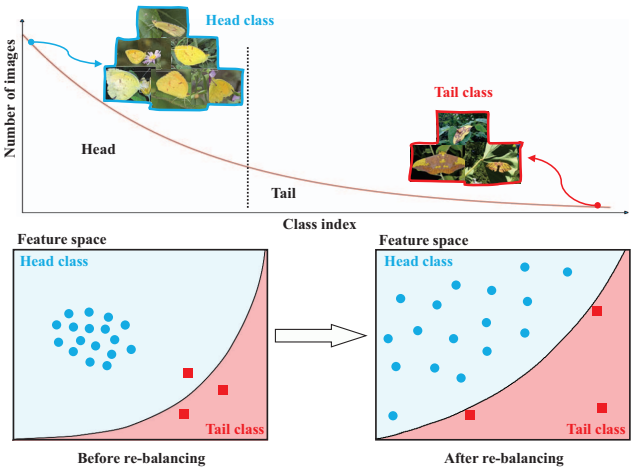


Figure 1. Real-world large-scale datasets often display the phenomenon of long-tailed distributions. The extreme imbalance causes tremendous challenges on the classification accuracy, especially for the tail classes. Class re-balancing strategies can yield better classification accuracy for long-tailed problems. In this paper, we reveal that the mechanism of these strategies is to significantly promote classifier learning but will unexpectedly damage the representative ability of the learned deep features to some extent. As conceptually demonstrated, after re-balancing, the decision boundary (*i.e.*, black solid arc) tends to accurately classify the tail data (*i.e.*, red squares). However, the intra-class distribution of each class becomes more separable. Quantitative results are presented in Figure 2, and more analyses can be found in the supplementary materials.

*X.-S. Wei is the corresponding author (weixs.gm@gmail.com). Q. Cui and Z.-M. Chen's contribution was made when they were interns in Megvii Research Nanjing, Megvii Technology, China. This research was supported by National Key R&D Program of China (No. 2017YFA0700800).

2575-7075/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPR42600.2020.00974

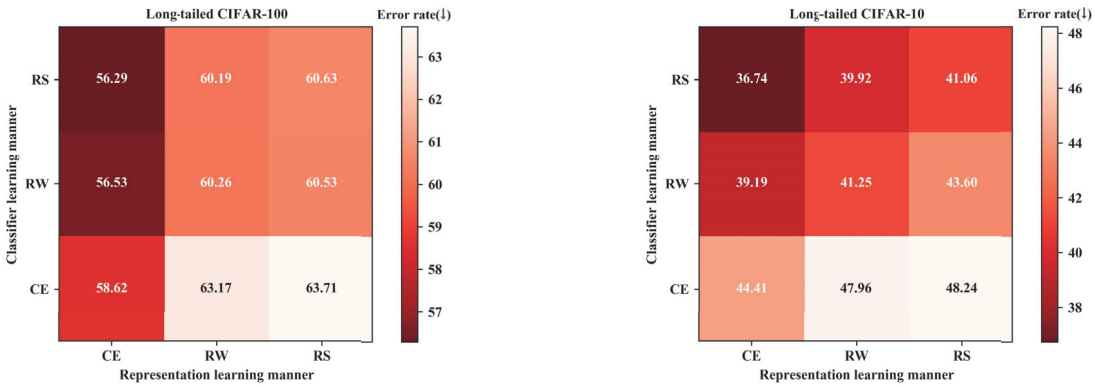


Figure 2. Top-1 error rates of different manners for representation learning and classifier learning on two long-tailed datasets CIFAR-100-IR50 and CIFAR-10-IR50 [3]. "CE" (Cross-Entropy), "RW" (Re-Weighting) and "RS" (Re-Sampling) are the conducted learning manners. As observed, when fixing the representation (comparing error rates of three blocks in the vertical direction), the error rates of classifiers trained with RW/RS are reasonably lower than CE. While, when fixing the classifier (comparing error rates in the horizontal direction), the representations trained with CE surprisingly get lower error rates than those with RW/RS. Experimental details can be found in Section 3.

such visual data, deep learning methods are not feasible to achieve outstanding recognition accuracy due to both the data-hungry limitation of deep models and also the extreme class imbalance trouble of long-tailed data distributions.

In the literature, the prominent and effective methods for handling long-tailed problems are class re-balancing strategies, which are proposed to alleviate the extreme imbalance of the training data. Generally, class re-balancing methods are roughly categorized into two groups, *i.e.*, re-sampling [25, 1, 14, 1, 11, 2, 7, 21, 4] and cost-sensitive re-weighting [13, 28, 5, 23]. These methods can adjust the network training, by re-sampling the examples or re-weighting the losses of examples within mini-batches, which is in expectation closer to the test distributions. Thus, class re-balancing is effective to directly influence the classifier weights' updating of deep networks, *i.e.*, promoting the classifier learning. That is the reason why re-balancing could achieve satisfactory recognition accuracy on long-tailed data.

However, although re-balancing methods have good eventual predictions, we argue that these methods still have adverse effects, *i.e.*, they will also unexpectedly damage the representative ability of the learned deep features (*i.e.*, the representation learning) to some extent. In concretely, re-sampling has the risks of over-fitting the tail data (by over-sampling) and also the risk of under-fitting the whole data distribution (by under-sampling), when data imbalance is extreme. For re-weighting, it will distort the original distributions by directly changing or even inverting the data presenting frequency.

As a preliminary of our work, by conducting validation experiments, we justify our aforementioned argumentations. Specifically, to figure out how re-balancing strategies work, we divide the training process of deep networks into two stages, *i.e.*, to separately conduct the representation learning and the classifier learning. At the former stage for representation learning, we employ plain training (conventional cross-entropy), re-weighting and re-sampling as three learning manners to obtain their corresponding learned representations. Then, at the latter stage for classifier learning, we first fix the parameters of representation learning (*i.e.*, backbone layers) converged at the former stage and then retrain the classifiers of these networks (*i.e.*, fully-connected layers) *from scratch*, also with the three aforementioned learning manners. In Figure 2, the prediction error rates on two benchmark long-tailed datasets [3], *i.e.*, CIFAR-100-IR50 and CIFAR-10-IR50, are reported. Obviously, when fixing the representation learning manner, re-balancing methods reasonably achieve lower error rates, indicating they can promote classifier learning. On the other side, by fixing the classifier learning manner, plain training on original imbalanced data can bring better results according to its better features. Also, the worse results of re-balancing methods prove that they will hurt feature learning.

Therefore, in this paper, for exhaustively improving the recognition performance of long-tailed problems, we propose a unified Bilateral-Branch Network (BBN) model to take care of *both representation learning and classifier learning* simultaneously. As shown in Figure 3, our BBN model consists of two branches, termed as the "conventional learning branch" and the "re-balancing branch". In general, each branch of BBN separately performs its own duty for representation learning and classifier learning, respectively. As the name suggests, the conventional learning branch equipped with the typical uniform sampler w.r.t. the original data distribution is responsible for learning universal patterns for recognition. While, the re-balancing branch coupled with a reversed sampler is designed to model the tail data. After that, the predicted outputs of these bilateral branches are aggregated in the cumulative learning part by an adaptive trade-off parameter α. α is automatically generated by the

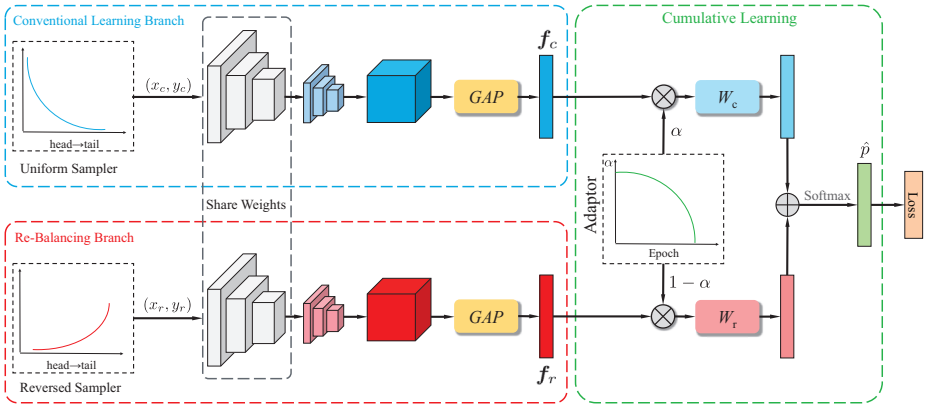


Figure 3. Framework of our Bilateral-Branch Network (BBN). It consists of three key components: 1) The *conventional learning branch* takes input data from a uniform sampler, which is responsible for learning universal patterns of original distributions. While, 2) the *re-balancing branch* takes inputs from a reversed sampler and is designed for modeling the tail data. The output feature vectors $\boldsymbol{f}_c$ and $\boldsymbol{f}_r$ of two branches are aggregated by 3) our *cumulative learning strategy* for computing training losses. "*GAP*" is short for global average pooling.

"Adaptor" according to the number of training epochs, which adjusts the whole BBN model to firstly learn the universal features from the original distribution and then pay attention to the tail data gradually. More importantly, α could further control the parameter updating of each branch, which, for example, avoids damaging the learned universal features when emphasizing the tail data at the later periods of training.

In experiments, empirical results on four benchmark long-tailed datasets show that our model obviously outperforms existing state-of-the-art methods. Moreover, extensive validation experiments and ablation studies can prove the aforementioned preliminary discovery and also validate the effectiveness of our tailored designs for long-tailed problems.

The main contributions of this paper are as follows:

- We explore the mechanism of the prominent class re-balancing methods for long-tailed problems, and further discover that these methods can significantly promote classifier learning and meanwhile will affect the representation learning w.r.t. the original data distribution.
- We propose a unified Bilateral-Branch Network (BBN) model to take care of both representation learning and classifier learning for exhaustively boosting long-tailed recognition. Also, a novel cumulative learning strategy is developed for adjusting the bilateral learnings and coupled with our BBN model's training.
- We evaluate our model on four benchmark long-tailed visual recognition datasets, and our proposed model consistently achieves superior performance over previous competing approaches.

2. Related work

Class re-balancing strategies: *Re-sampling* methods as one of the most important class re-balancing strategies could be divided into two types: 1) Over-sampling by simply repeating data for minority classes [25, 1, 2] and 2) under-sampling by abandoning data for dominant classes [14, 1, 11]. But sometimes, with re-sampling, duplicated tailed samples might lead to over-fitting upon minority classes [4, 5], while discarding precious data will certainly impair the generalization ability of deep networks.

Re-weighting methods are another series of prominent class re-balancing strategies, which usually allocate large weights for training samples of tail classes in loss functions [13, 28]. However, re-weighting is not capable of handling the large-scale, real-world scenarios of long-tailed data and tends to cause optimization difficulty [20]. Consequently, Cui *et al.* [5] proposed to adopt the effective number of samples [5] instead of proportional frequency. Thereafter, Cao *et al.* [3] explored the margins of the training examples and designed a label-distribution-aware loss to encourage larger margins for minority classes.

In addition, recently, some two-stage fine-tuning strategies [3, 6, 22] were developed to modify re-balancing for effectively handling long-tailed problems. Specifically, they separated the training process into two single stages. In the first stage, they trained networks as usual on the original imbalanced data and only utilized re-balancing at the second stage to fine-tune the network with a small learning rate.

Beyond that, other methods of different learning paradigms were also proposed to deal with long-tailed problems, *e.g.*, metric learning [34, 13], meta-learning [19] and knowledge transfer learning [28, 36], which, however, are not within the scope of this paper.

Mixup: Mixup [33] was a general data augmentation algorithm, *i.e.*, convexly combining random pairs of training images and their associated labels, to generate *additional* samples when training deep networks. Also, manifold mixup [27] conducted mixup operations on random

pairs of samples in the manifold feature space for augmentation. The mixed ratios in mixup were sampled from the β-distribution to increase the randomness of augmentation. Although mixup is clearly far from our unified end-to-end trainable model, in experiments, we still compared with a series of mixup algorithms to validate our effectiveness.

3. How class re-balancing strategies work?

In this section, we attempt to figure out the working mechanism of these class re-balancing methods. More concretely, we divide a deep classification model into two essential parts: 1) the feature extractor (*i.e.*, frontal base/backbone networks) and 2) the classifier (*i.e.*, last fully-connected layers). Accordingly, the learning process of a deep classification network could be separated into representation learning and classifier learning. Since class re-balancing strategies could boost the classification accuracy by altering the training data distribution closer to test and paying more attention to the tail classes, we propose a conjecture that the way these strategies work is to promote classifier learning significantly but might damage the universal representative ability of the learned deep features due to distorting original distributions.

In order to justify our conjecture, we design a two-stage experimental fashion to separately learn representations and classifiers of deep models. Concretely, in the first stage, we train a classification network with plain training (*i.e.*, cross-entropy) or re-balancing methods (*i.e.*, re-weighting/re-sampling) as learning manners. Then, we obtain different kinds of feature extractors corresponding to these learning manners. When it comes to the second stage, we fix the parameters of the feature extractors learned in the former stage, and retrain classifiers *from scratch* with the aforementioned learning manners again. In principle, we design these experiments to fairly compare the quality of representations and classifiers learned by different manners by following the control variates method.

The CIFAR [16] datasets are a collection of images that are commonly used to assess computer vision approaches. Previous work [5, 3] created long-tailed versions of CIFAR datasets with different imbalance ratios, *i.e.*, the number of the most frequent class divided by the least frequent class, to evaluate the performance. In this section, following [3], we also use long-tailed CIFAR-10/CIFAR-100 as the test beds.

As shown in Figure 2, we conduct several contrast experiments to validate our conjecture on CIFAR-100-IR50 (long-tailed CIFAR-100 with imbalance ratio 50). As aforementioned, we separate the whole network into two parts: the feature extractor and classifier. Then, we apply three manners for the feature learning and the classifier learning respectively according to our two-stage training fashion. Thus, we can obtain nine groups of results based on different permutations: (1) Cross-Entropy (CE): We train the networks as usual on the original imbalanced data with the conventional cross-entropy loss. (2) Re-Sampling (RS): We first sample a class uniformly and then collect an example from that class by sampling with replacement. By repeating this process, a balanced mini-batch data is obtained. (3) Re-Weighting (RW): We re-weight all the samples by the inverse of the sample size of their classes. The error rate is evaluated on the validation set. As shown in Figure 2, we have the observations from two perspectives:

- **Classifiers:** When we apply the same representation learning manner (comparing error rates of three blocks in the vertical direction), it can be reasonably found that RW/RS always achieve lower classification error rates than CE, which owes to their re-balancing operations adjusting the classifier weights' updating to match test distributions.
- **Representations:** When applying the same classifier learning manner (comparing error rates of three blocks in the horizontal direction), it is a bit of surprise to see that error rates of CE blocks are consistently lower than error rates of RW/RS blocks. The findings indicate that training with CE achieves better classification results since it obtains better features. The worse results of RW/RS reveal that they lead to inferior discriminative ability of the learned deep features.

Furthermore, as shown in Figure 2 (left), by employing CE on the representation learning and employing RS on the classifier learning, we can achieve the lowest error rate on the validation set of CIFAR-100-IR50. Additionally, to evaluate the generalization ability for representations produced by three manners, we utilize pre-trained models trained on CIFAR-*100*-IR50 as the feature extractor to obtain the representations of CIFAR-*10*-IR50, and then perform the classifier learning experiments as the same as aforementioned. As shown in Figure 2 (right), on CIFAR-10-IR50, it can have the identical observations, even in the situation that the feature extractor is trained on another long-tailed dataset.

4. Methodology

4.1. Overall framework

As shown in Figure 3, our BBN consists of three main components. Concretely, we design two branches for representation learning and classifier learning, termed "*conventional learning branch*" and "*re-balancing branch*", respectively. Both branches use the same residual network structure [12] and share all the weights except for the last residual block. Let $\mathbf{x}_\cdot$ denote a training sample and $y_\cdot \in \{1, 2, \ldots, C\}$ is its corresponding label, where C is the number of classes. For the bilateral branches, we apply uniform and reversed samplers to each of them separately and obtain two samples $(\mathbf{x}_c, y_c)$ and $(\mathbf{x}_r, y_r)$ as the input data, where $(\mathbf{x}_c, y_c)$ is for the conventional learning branch and $(\mathbf{x}_r, y_r)$ is for the re-balancing branch. Then, two samples

are fed into their own corresponding branch to acquire the feature vectors $\boldsymbol{f}_c \in \mathbb{R}^D$ and $\boldsymbol{f}_r \in \mathbb{R}^D$ by global average pooling.

Furthermore, we also design a specific cumulative learning strategy for shifting the learning "attention" between two branches in the training phase. In concretely, by controlling the weights for $\boldsymbol{f}_c$ and $\boldsymbol{f}_r$ with an adaptive trade-off parameter α, the weighted feature vectors $\alpha \boldsymbol{f}_c$ and $(1-\alpha)\boldsymbol{f}_r$ will be sent into the classifiers $\boldsymbol{W}_c \in \mathbb{R}^{D\times C}$ and $\boldsymbol{W}_r \in \mathbb{R}^{D\times C}$ respectively and the outputs will be integrated together by element-wise addition. The output logits are formulated as

$$\mathbf{z} = \alpha \boldsymbol{W}_c^\top \boldsymbol{f}_c + (1-\alpha)\boldsymbol{W}_r^\top \boldsymbol{f}_r, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^C$ is the predicted output, *i.e.*, $[z_1, z_2, \ldots, z_C]^\top$. For each class $i \in \{1, 2, \ldots, C\}$, the softmax function calculates the probability of the class by

$$\hat{p}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}. \tag{2}$$

Then, we denote $E(\cdot,\cdot)$ as the cross-entropy loss function and the output probability distribution as $\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_C]^\top$. Thus, the weighted cross-entropy classification loss of our BBN model is illustrated as

$$\mathcal{L} = \alpha E(\hat{\boldsymbol{p}}, y_c) + (1-\alpha) E(\hat{\boldsymbol{p}}, y_r), \tag{3}$$

and the whole network is end-to-end trainable.

4.2. Proposed bilateral-branch structure

In this section, we elaborate the details of our unified bilateral-branch structure shown in Figure 3. As aforementioned, the proposed conventional learning branch and re-balancing branch do perform their own duty (*i.e.*, representation learning and classifier learning, respectively). There are two unique designs for these branches.

Data samplers. The input data for the conventional learning branch comes from a uniform sampler, where each sample in the training dataset is sampled only once with equal probability in a training epoch. The uniform sampler retains the characteristics of original distributions, and therefore benefits the representation learning. While, the re-balancing branch aims to alleviate the extreme imbalance and particularly improve the classification accuracy on tail classes [26], whose input data comes from a reversed sampler. For the reversed sampler, the sampling possibility of each class is proportional to the reciprocal of its sample size, *i.e.*, the more samples in a class, the smaller sampling possibility that class has. In formulations, let denote that the number of samples for class i is N_i and the maximum sample number of all the classes is N_{max}. There are three sub-procedures to construct the reversed sampler: 1) Calculate the sampling possibility P_i for class i according to the number of samples as

$$P_i = \frac{w_i}{\sum_{j=1}^{C} w_j}, \tag{4}$$

where $w_i = \frac{N_{max}}{N_i}$; 2) Randomly sample a class according to P_i; 3) Uniformly pick up a sample from class i with replacement. By repeating this reversed sampling process, training data of a mini-batch is obtained.

Weights sharing. In BBN, both branches economically share the same residual network structure, as illustrated in Figure 3. We use ResNets [12] as our backbone network, *e.g.*, ResNet-32 and ResNet-50. In details, two branch networks, except for the last residual block, share the same weights. There are two benefits for sharing weights: On the one hand, the well-learned representation by the conventional learning branch can benefit the learning of the re-balancing branch. On the other hand, sharing weights will largely reduce computational complexity in the inference phase.

4.3. Proposed cumulative learning strategy

Cumulative learning strategy is proposed to shift the learning focus between the bilateral branches by controlling both the weights for features produced by two branches and the classification loss $\mathcal{L}$. It is designed to first learn the universal patterns and then pay attention to the tail data gradually. In the training phase, the feature $\boldsymbol{f}_c$ of the conventional learning branch will be multiplied by α and the feature $\boldsymbol{f}_r$ of the re-balancing branch will be multiplied by $1-\alpha$, where α is automatically generated according to the training epoch. Concretely, the number of total training epochs is denoted as T_{max} and the current epoch is T. α is calculated by

$$\alpha = 1 - \left(\frac{T}{T_{max}}\right)^2, \tag{5}$$

which α will gradually decrease as the training epochs increasing.

In intuition, we design the adapting strategy for α based on the motivation that discriminative feature representations are the foundation for learning robust classifiers. Although representation learning and classifier learning deserve equal attentions, the learning focus of our BBN should gradually change from feature representations to classifiers, which can exhaustively improve long-tailed recognition accuracy. With α decreasing, the main emphasis of BBN turns from the conventional learning branch to the re-balancing branch. Different from two-stage fine-tuning strategies [3, 6, 22], our α ensures that both branches for different goals can be constantly updated in the whole training process, which could avoid the affects on one goal when it performs training for the other goal.

In experiments, we also provide the qualitative results of this intuition by comparing different kinds of adaptors, cf. Section 5.5.2.

Table 1. Top-1 error rates of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. (Best results are marked in bold.)

Dataset	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
CE	29.64	25.19	13.61	61.68	56.15	44.29
Focal [17]	29.62	23.28	13.34	61.59	55.68	44.22
Mixup [33]	26.94	22.18	12.90	60.46	55.01	41.98
Manifold Mixup [27]	27.04	22.05	12.97	61.75	56.91	43.45
Manifold Mixup (two samplers)	26.90	20.79	13.17	63.19	57.95	43.54
CE-DRW [3]	23.66	20.03	12.44	58.49	54.71	41.88
CE-DRS [3]	24.39	20.19	12.62	58.39	54.52	41.89
CB-Focal [5]	25.43	20.73	12.90	60.40	54.83	42.01
LDAM-DRW [3]	22.97	18.97	11.84	57.96	53.38	41.29
Our BBN	**20.18**	**17.82**	**11.68**	**57.44**	**52.98**	**40.88**

4.4. Inference phase

During inference, the test samples are fed into both branches and two features $\boldsymbol{f}_c'$ and $\boldsymbol{f}_r'$ are obtained. Because both branches are equally important, we simply fix α to 0.5 in the test phase. Then, the equally weighted features are fed to their corresponding classifiers (*i.e.*, $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$) to obtain two prediction logits. Finally, both logits are aggregated by element-wise addition to return the classification results.

5. Experiments

5.1. Datasets and empirical settings

Long-tailed CIFAR-10 and CIFAR-100. Both CIFAR-10 and CIFAR-100 contain 60,000 images, 50,000 for training and 10,000 for validation with category number of 10 and 100, respectively. For fair comparisons, we use the long-tailed versions of CIFAR datasets as the same as those used in [3] with controllable degrees of data imbalance. We use an imbalance factor β to describe the severity of the long tail problem with the number of training samples for the most frequent class and the least frequent class, *e.g.*, $\beta = \frac{N_{max}}{N_{min}}$. Imbalance factors we use in experiments are 10, 50 and 100.

iNaturalist 2017 and iNaturalist 2018. The iNaturalist species classification datasets are large-scale real-world datasets that suffer from extremely imbalanced label distributions. The 2017 version of iNaturalist contains 579,184 images with 5,089 categories and the 2018 version is composed of 437,513 images from 8,142 categories. Note that, besides the extreme imbalance, the iNaturalist datasets also face the fine-grained problem [32, 35, 30, 31]. In this paper, the official splits of training and validation images are utilized for fair comparisons.

5.2. Implementation details

Implementation details on CIFAR. For long-tailed CIFAR-10 and CIFAR-100 datasets, we follow the data augmentation strategies proposed in [12]: randomly crop a 32×32 patch from the original image or its horizontal flip with 4 pixels padded on each side. We train the ResNet-32 [12] as our backbone network for all experiments by standard mini-batch stochastic gradient descent (SGD) with momentum of 0.9, weight decay of 2×10^{-4}. We train all the models on a single NVIDIA 1080Ti GPU for 200 epochs with batch size of 128. The initial learning rate is set to 0.1 and the first five epochs is trained with the linear warm-up learning rate schedule [8]. The learning rate is decayed at the 120^{th} and 160^{th} epoch by 0.01 for our BBN.

Implementation details on iNaturalist. For fair comparisons, we utilize ResNet-50 [12] as our backbone network in all experiments on iNaturalist 2017 and iNaturalist 2018. We follow the same training strategy in [8] with batch size of 128 on four GPUs of NVIDIA 1080Ti. We firstly resize the image by setting the shorter side to 256 pixels and then take a 224×224 crop from it or its horizontal flip. During training, we decay the learning rate at the 60^{th} and 80^{th} epoch by 0.1 for our BBN, respectively.

5.3. Comparison methods

In experiments, we compare our BBN model with three groups of methods:

- **Baseline methods.** We employ plaining training with cross-entropy loss and focal loss [17] as our baselines. Note that, we also conduct experiments with a series of mixup algorithms [33, 27] for comparisons.
- **Two-stage fine-tuning strategies.** To prove the effectiveness of our cumulative learning strategy, we also compare with the two-stage fine-tuning strategies proposed in previous state-of-the-art [3]. We train networks with cross-entropy (CE) on imbalanced data in the first stage, and then conduct class re-balancing training in the second stage. "CE-DRW" and "CE-DRS" refer to the two-stage baselines using re-weighting and re-sampling at the second stage.
- **State-of-the-art methods.** For state-of-the-art methods, we compare with the recently proposed LDAM [3] and CB-Focal [5] which achieve good classification accuracy on these four aforementioned long-tailed datasets.

Table 2. Top-1 error rates of ResNet-50 on large-scale long-tailed datasets iNaturalist 2018 and iNaturalist 2017. Our method outperforms the previous state-of-the-arts by a large margin, especially with 2× scheduler. "*" indicates original results in that paper.

Dataset	iNaturalist 2018	iNaturalist 2017
CE	42.84	45.38
CE-DRW [3]	36.27	40.48
CE-DRS [3]	36.44	40.12
CB-Focal [5]	38.88	41.92
LDAM-DRW* [3]	32.00	–
LDAM-DRW [3]	35.42	39.49
LDAM-DRW [3] (2×)	33.88	38.19
Our BBN	33.71	36.61
Our BBN (2×)	**30.38**	**34.25**

Table 3. Ablation studies for different samplers for the re-balancing branch of BBN on long-tailed CIFAR-10-IR50.

Sampler	Error rate
Uniform sampler	21.31
Balanced sampler	21.06
Reversed sampler (Ours)	**17.82**

Table 4. Ablation studies of different adaptor strategies of BBN on long-tailed CIFAR-10-IR50.

Adaptor	α	Error rate
Equal weight	0.5	21.56
β-distribution	$Beta(0.2, 0.2)$	21.75
Parabolic increment	$\left(\frac{T}{T_{max}}\right)^2$	22.70
Linear decay	$1 - \frac{T}{T_{max}}$	18.55
Cosine decay	$\cos(\frac{T}{T_{max}} \cdot \frac{\pi}{2})$	18.04
Parabolic decay (Ours)	$1 - \left(\frac{T}{T_{max}}\right)^2$	**17.82**

5.4. Main results

5.4.1 Experimental results on long-tailed CIFAR

We conduct extensive experiments on long-tailed CIFAR datasets with three different imbalanced ratios: 10, 50 and 100. Table 1 reports the error rates of various methods. We demonstrate that our BBN consistently achieves the best results across all the datasets, when comparing other comparison methods, including the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS), the series of mixup algorithms (*i.e.*, mixup, manifold mixup and manifold mixup with two samplers as the same as ours), and also previous state-of-the-arts (*i.e.*, CB-Focal [5] and LDAM-DRW [3]).

Especially for long-tailed CIFAR-10 with imbalanced ratio 100 (an extreme imbalance case), we get 20.18% error rate which is 2.79% lower than that of LDAM-DRW [3]. Additionally, it can be found from that table, the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS) are effective, since they could obtain comparable or even better results comparing with state-of-the-art methods.

5.4.2 Experimental results on iNaturalist

Table 2 shows the results on two large-scale long-tailed datasets, *i.e.*, iNaturalist 2018 and iNaturalist 2017. As shown in that table, the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS) also perform well, which have consistent observations with those on long-tailed CIFAR. Compared with other methods, on iNaturalist, our BBN still outperform competing approaches and baselines. Besides, since iNaturalist is large-scale, we also conduct network training with the 2× scheduler. Meanwhile, for fair comparisons, we further evaluate the previous state-of-the-art LDAM-DRW [3] with the 2× training scheduler. It is obviously to see that, with 2× scheduler, our BBN achieves significantly better results than BBN without 2× scheduler. Additionally, compared with LDAM-DRW (2×), we achieve $+3.50\%$ and $+3.94\%$ improvements on iNaturalist 2018 and iNaturalist 2017, respectively. In addition, even though we do not use the 2× scheduler, our BBN can still get the best results. For a detail, we conducted the experiments based on LDAM [3] with the source codes provided by the authors, but failed to reproduce the results reported in that paper.

5.5. Ablation studies

5.5.1 Different samplers for the re-balancing branch

For better understanding our proposed BBN model, we conduct experiments on different samplers utilized in the re-balancing branch. We present the error rates of models trained with different samplers in Table 3. For clarity, the uniform sampler maintains the original long-tailed distribution. The balanced sampler assigns the same sampling possibility to all classes, and construct a mini-batch training data obeying a balanced label distribution. As shown in that table, the reversed sampler (our proposal) achieves considerably better performance than the uniform and balanced samplers, which indicates that the re-balancing branch of BBN should pay more attention to the tail classes by enjoying the reversed sampler.

5.5.2 Different cumulative learning strategies

To facilitate the understanding of our proposed cumulative learning strategy, we explore several different strategies to generate the adaptive trade-off parameter α on CIFAR-10-IR50. Specifically, we test with both progress-relevant/irrelevant strategies, cf. Table 4. For clarity, progress-relevant strategies adjust α with the number of training epochs, *e.g.*, linear decay, cosine decay, *etc*. Progress-irrelevant strategies include the equal weight or generate from a discrete distribution (*e.g.*, the β-distribution).

Table 5. Feature quality evaluation for different learning manners.

Representation learning manner	Error rate
CE	**58.62**
RW	63.17
RS	63.71
BBN-CB	58.89
BBN-RB	61.09

As shown in Table 4, the decay strategies (*i.e.*, linear decay, cosine decay and our parabolic decay) for generating α can yield better results than the other strategies (*i.e.*, equal weight, β-distribution and parabolic increment). These observations prove our motivation that the conventional learning branch should be learned firstly and then the re-balancing branch. Among these strategies, the best way for generating α is the proposed parabolic decay approach. In addition, the parabolic increment, where re-balancing are attended before conventional learning, performs the worst, which validates our proposal from another perspective. More detailed discussions can be found in the supplementary materials.

5.6. Validation experiments of our proposals

5.6.1 Evaluations of feature quality

It is proven in Section 3 that learning with vanilla CE on original data distribution can obtain good feature representations. In this subsection, we further explore the representation quality of our proposed BBN by following the empirical settings in Section 3. Concretely, given a BBN model trained on CIFAR-100-IR50, firstly, we fix the parameters of representation learning of two branches. Then, we separately retrain the corresponding classifiers from scratch of two branches also on CIFAR-100-IR50. Finally, classification error rates are tested on these two branches independently.

As shown in Table 5, the feature representations obtained by the conventional learning branch of BBN ("BBN-CB") achieves comparable performance with CE, which indicates that our proposed BBN greatly preserves the representation capacity learned from the original long-tailed dataset. Note that, the re-balancing branch of BBN ("BBN-RB") also gets better performance than RW/RS and it possibly owes to the parameters sharing design of our model.

5.6.2 Visualization of classifier weights

Let denote $\boldsymbol{W} \in \mathbb{R}^{D\times C}$ as a set of classifiers $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_C\}$ for all the C classes, where $\boldsymbol{w}_i \in \mathbb{R}^D$ indicates the weight vector for class i. Previous work [9] has shown that the value of ℓ_2-norm $\{\|\boldsymbol{w}_i\|_2\}_{i=1}^{C}$ for different classes can demonstrate the preference of a classifier, *i.e.*, the classifier $\boldsymbol{w}_i$ with the largest ℓ_2-norm tends to judge one example as belonging to its class i. Following [9], we visualize the ℓ_2-norm of these classifiers.

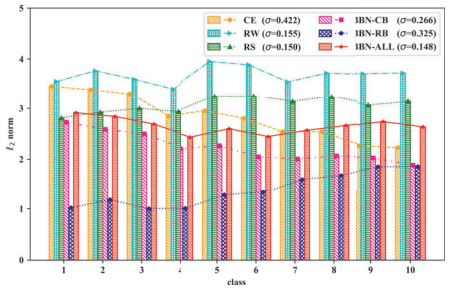


Figure 4. ℓ_2-norm of classifier weights for different learning manners. Specifically, "BBN-ALL" indicates the ℓ_2-norm of the combination of $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$ in our model. σ in the legend is the standard deviation of ℓ_2-norm for ten classes.

As shown in Figure 4, we visualize the ℓ_2-norm of ten classes trained on CIFAR-10-IR50. For our BBN, we visualize the classifier weights $\boldsymbol{W}_c$ of the conventional learning branch ("BBN-CB") and the classifier weights $\boldsymbol{W}_r$ of the re-balancing branch ("BBN-RB"), as well as their combined classifier weights ("BBN-ALL"). Additionally, the visualization results on classifiers trained with these learning manners in Section 3, *i.e.*, CE, RW and RS, are also provided.

Obviously, the ℓ_2-norm of ten classes' classifiers for our proposed model (*i.e.*, "BBN-ALL") are basically equal, and their standard deviation $\sigma = 0.148$ is the smallest one. For the classifiers trained by other learning manners, the distribution of the ℓ_2-norm of CE is consistent with the long-tailed distribution. The ℓ_2-norm distribution of RW/RS looks a bit flat, but their standard deviations are larger than ours. It gives an explanation why our BBN can outperform these methods. Additionally, by separately analyzing our model, its conventional learning branch ("BBN-CB") has a similar ℓ_2-norm distribution with CE's, which justifies its duty is focusing on universal feature learning. The ℓ_2-norm distribution of the re-balancing branch ("BBN-RB") has a reversed distribution w.r.t. original long-tailed distributions, which reveals it is able to model the tail.

6. Conclusions

In this paper, for studying long-tailed problems, we explored how class re-balancing strategies influenced representation learning and classifier learning of deep networks, and revealed that they can promote classifier learning significantly but also damage representation learning to some extent. Motivated by this, we proposed a Bilateral-Branch Network (BBN) with a specific cumulative learning strategy to take care of both representation learning and classifier learning for exhaustively improving the recognition performance of long-tailed tasks. By conducting extensive experiments, we proved that our BBN could achieve the best results on long-tailed benchmarks, including the large-scale iNaturalist. In the future, we attempt to tackle the long-tailed detection problems with our BBN model.

References

[1] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 2, 3

[2] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881, 2019. 2, 3

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pages 1–18, 2019. 2, 3, 4, 5, 6, 7

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 2, 3

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019. 2, 3, 4, 6, 7

[6] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, pages 4109–4118, 2018. 1, 3, 5

[7] Chris Drummond and Robert C Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. *Workshop on Learning From Imbalanced Datasets II*, 11:1–8, 2003. 2

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, pages 1–12, 2017. 6

[9] Yandong Guo and Lei Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, pages 1–12, 2017. 8

[10] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 1

[11] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. 2, 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 5, 6

[13] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016. 2, 3

[14] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. 2, 3

[15] Maurice George Kendall, Alan Stuart, John Keith Ord, Steven F Arnold, Anthony O'Hagan, and Jonathan Forster. *Kendall's advanced theory of statistics*, volume 1. 1987. 1

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 4

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 6

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[19] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 1–10, 2019. 3

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 3

[21] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, pages 1–7, 2016. 2

[22] Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *CVPR*, pages 864–873, 2016. 3, 5

[23] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 1–13, 2018. 2

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[25] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, pages 467–482, 2016. 2, 3

[26] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, pages 1–22, 2017. 1, 5

[27] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019. 3, 6

[28] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, pages 7029–7039, 2017. 2, 3

[29] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. RPC: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, pages 1–24, 2019. 1

[30] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Transactions on Image Processing*, 26(6):2868–2881, 2017. 6

[31] Xiu-Shen Wei, Peng Wang, Lingqiao Liu, Chunhua Shen, and Jianxin Wu. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Transactions on Image Processing*, 28(12):6116–6125, 2019. 6

[32] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069*, pages 1–7, 2019. 6

[33] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, pages 1–13, 2018. 3, 6

[34] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, pages 5409–5418, 2017. 3

[35] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017. 6

[36] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, pages 7812–7821, 2019. 3

[37] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for ccene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1

Large scale long-tailed product recognition system at Alibaba

Xiangzeng Zhou, Pan Pan, Yun Zheng, Yinghui Xu, Rong Jin
Machine Intelligence Technology Lab, Damo Academy
Alibaba Group, Hangzhou, China
xiangzeng.zxz,panpan.pp,zhengyun.zy@alibaba-inc.com
renji.xyh@taobao.com,jinrong.jr@alibaba-inc.com

ABSTRACT

A practical large scale product recognition system suffers from the phenomenon of long-tailed imbalanced training data under the E-commercial circumstance at Alibaba. Besides product images at Alibaba, plenty of image related side information (e.g. title, tags) reveal rich semantic information about images. Prior works mainly focus on addressing the long tail problem in visual perspective only, but lack of consideration of leveraging the side information. In this paper, we present a novel side information based large scale visual recognition co-training (SICoT) system to deal with the long tail problem by leveraging the image related side information. In the proposed co-training system, we firstly introduce a bilinear word attention module aiming to construct a semantic embedding over the noisy side information. A visual feature and semantic embedding co-training scheme is then designed to transfer knowledge from classes with abundant training data (head classes) to classes with few training data (tail classes) in an end-to-end fashion. Extensive experiments on four challenging large scale datasets, whose numbers of classes range from one thousand to one million, demonstrate the scalable effectiveness of the proposed SICoT system in alleviating the long tail problem. In the visual search platform Pailitao[1] at Alibaba, we settle a practical large scale product recognition application driven by the proposed SICoT system, and achieve a significant gain of unique visitor (UV) conversion rate.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

product recognition, long-tailed, attention, side information, co-training

ACM Reference Format:
Xiangzeng Zhou, Pan Pan, Yun Zheng, Yinghui Xu, Rong Jin. 2020. Large scale long-tailed product recognition system at Alibaba. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3340531.3417445

[1]http://www.pailitao.com

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3417445

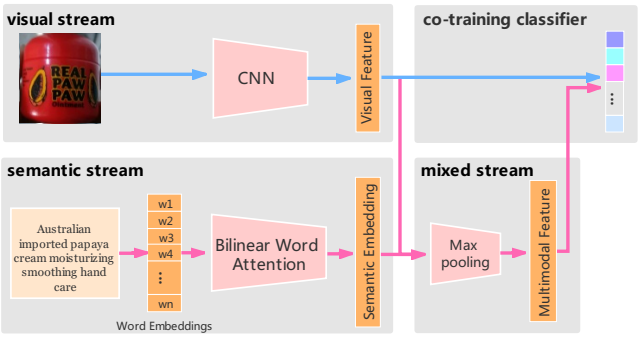


Figure 1: The overall architecture of our proposed side information based co-training (SICoT) system. The system contains four streams: (i) The visual stream is a visual feature extractor using a convolutional neural network. (ii) The semantic stream, which consists of a word2vec module and a proposed bilinear word attention module, aims to learn a semantic embedding from the noisy side information. (iii) The mixed stream takes charge of generating a multimodal feature. (iv) The shared classifier is co-trained by the visual stream and the mixed stream.

1 INTRODUCTION

Recent years have witnessed the remarkable progresses of wielding deep learning models in visual recognition task. With the aid of deep learning techniques, nowadays it is practicable to establish an industrial large scale visual recognition application based on huge volume of image data. Compared to the quantity of products and image data under the E-commercial circumstance at Alibaba, many popular so-called large scale visual recognition datasets, like ImageNet [6], WebVision2.0 [16], iMaterialist Product 2019 [13] and Open Images V4 [15], appear to be relatively small scale. Even though some datasets have reached several millions of training image data, the quantity of categories only ranges from hundreds to thousands.

On the basis of abundant image data of great value in the E-commercial scenario and powerful computing resources at Alibaba, it is still great challenging to establish a truly practical large scale visual recognition application. And these challenges are roughly reflected in following three aspects:

Enormous quantity of classes and images: There are about tens of million of daily active products and billions of image data

in the marketplace of Alibaba, which covers categories of clothing, shoe, bag, cosmetic, drink, snack and toy in general. The huge quantity of product classes and images brings difficulties to both the training and deployment of a large scale visual recognition model. For example, when the number of classes reaches about one million, the size of the last fully connected (FC) layer will exceed the maximum memory of a single block of Nvidia-V100-32G GPU. This requires a new training paradigm capable of training a huge FC layer in a distributed manner.

Scarce and noisy annotation: Unlike those well compiled small scale datasets (e.g. ImageNet [6]), it is impracticable to manually annotate the daily growing enormous quantity of image data in the E-commercial scenario. Training images with high-quality annotations are scarce and insufficient to build a practical large scale visual recognition system. Although there are lots of annotations provided by sellers or customers in the marketplace of Alibaba, it is of great difficulty to use these weakly and noisy annotations to assist in training a visual recognition model.

Long-tailed distribution of training data: The phenomenon of long-tailed imbalanced training data naturally occurs under the E-commercial circumstance. In the marketplace of Alibaba, a large amount of new arrival products emerge everyday, meanwhile, quite a large portion of products are low sales or even zero sale. The difficulty to acquire sufficient training images for these products challenges the performance of a large scale visual recognition application.

It is known that, without any special treatment of the classes with insufficient training data (tail classes), the classification boundary of a recognition model inclines toward those classes with abundant training data (head classes). At Alibaba, there are abundant image related data or side information, such as short titles and long text descriptions, coming from various sellers or customers. These side information that containing rich, yet weak and noisy annotations reveal underlying similarity among images and classes from a different perspective.

However, prior works [2, 7–9, 14, 20, 24, 31, 33] mainly focus on addressing the long tail problem in visual perspective only, but lack of consideration of leveraging the side information. On the other hand, most works [4, 12, 19] take advantage of the side information as a kind of weakly supervision in general, and are not meant to address the long tail problem. Inspired by the work of transferring knowledge or borrowing training examples between similar classes [17] in detection task, we attempt to address the problem of long-tailed distributed training data in the task of large scale product recognition by leveraging the noisy side information in this paper. Considering the following two facts observed on the data of the marketplace at Alibaba, the usage of image related side information has great potential to alleviate the long tail problem. a) Unlike the extreme imbalanced distribution of image data, the distribution of words from image related titles is relatively much balanced. b) About 12% words out of the entire vocabulary are shared between the head classes and the tail classes.

In this paper, we propose a novel side information based visual recognition co-training (SICoT) system, as shown in Fig. 1, which aims to deal with the long tail problem in a large scale product classification task. Moreover, we have launched a SKU level product recognition service driven by the proposed SICoT system in the visual search platform Pailitao [1, 32] at Alibaba, as shown in Fig. 2. We conclude our contributions as following:

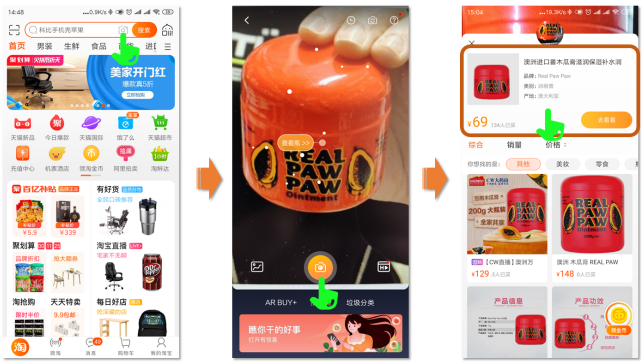


Figure 2: The scenario of large scale SKU level product recognition service on Pailitao at Alibaba: by taking a picture, Pailitao identifies the product with its short title and several associated tags in real time shown at the top of the page.

1) We introduce a bilinear word attention module to distinguish important words from the noisy side information of image short titles, followed by constructing a semantic embedding as a kind of distilled knowledge of the side information.

2) Considering the long tail problem in a large scale product recognition task, we propose a novel visual feature and semantic embedding co-training (SICoT) system to help transfer knowledge from the head classes to the tail classes in an end-to-end way. This co-training system aims to perform transfer learning across the head and the tail classes by deeply involving the side information in both feature learning and classifier training.

3) Extensive experiments on both open large scale datasets and our organized huge scale SKU level product datasets demonstrate the scalable effectiveness of the proposed side information based co-training system in relieving the long tail problem.

2 RELATED WORK

Taking the issue of long-tailed distributed training data into account, it still remains very challenging problems on how to establish a practical large scale product recognition system in the E-commercial scenario at Alibaba. Some prior works about addressing the long tail problem, taking advantaging of side information and large scale product recognition are roughly summarised in the following aspects.

Imbalanced learning: To alleviate the problem of long-tailed distributed training data, many traditional approaches have been extensively studied in the past [2, 7–9, 14, 20, 24, 31, 33]. Re-sampling methods [9, 20] aim to balance the numbers of training samples between multiple classes by under-sampling the head classes or over-sampling the tail classes. However, these methods often lead to removal of important samples or introduction of meaningless duplicated samples. Cost-sensitive methods [24, 33] try to make the standard classifiers more sensitive to the head classes by imposing higher misclassification cost to the head classes than to the tail classes. Most recently, deep neural networks are widely applied to performing imbalanced learning [2, 7, 8, 14, 31, 33]. Apart from the

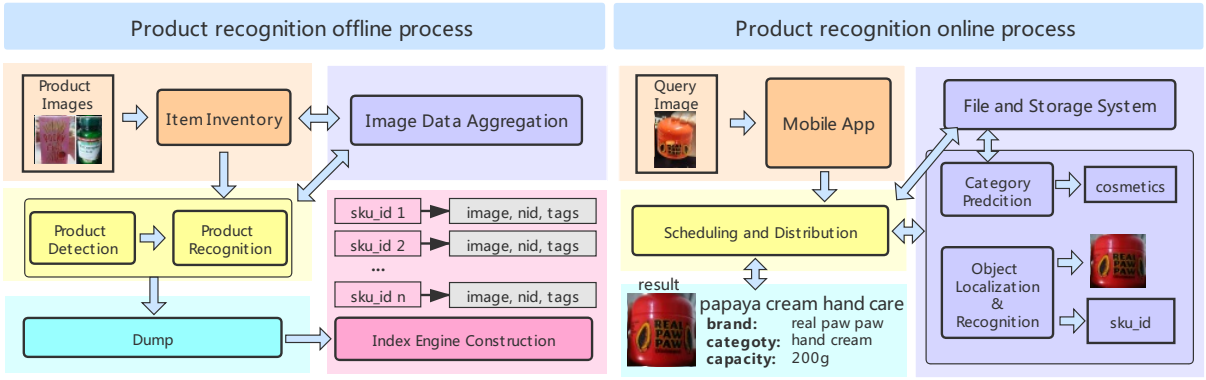


Figure 3: Overview of the overall product recognition architecture settled in Pailitao.

conventional cross entropy loss, several new objective loss function, like range loss [31], class rectification loss [7] and cluster-based large margin local embedding [8] are proposed to address the long tail problem by rectifying the classification boundaries dominated by the head classes. The works mentioned above devote major effort to addressing the long tail problem merely from the visual aspect. Considering that the side information can provide rich semantic information about image, in this paper the side information are involved in helping tackle the long tail problem.

Weakly supervised learning: In many popular datasets and practical scenarios, lots of auxiliary data or side information associated with images is provided, such as image titles and long text descriptions of in WebVision2.0 [16], wordnet in ImageNet [6] and so on. These side information normally come from heterogeneous data sources via web search, and naturally contain a lot of noise. In most of prior works, the noisy side information is mainly taken as a kind of weakly supervision in coordination with other kinds of learning tasks [4, 12, 19]. However, especially in a classification task, taking advantage of the side information as supervision in the one-hot fashion may not fully exploit the knowledge and be somewhat shallow. Besides, the usage of the side information is these works are not meant to address the long tail problem. In our proposed SICoT system, we explore a novel fashion to handle the long tail problem by taking advantage of the side information.

Knowledge distill and transfer: The basic principle of transfer learning are also introduced to transfer knowledge or even borrow training data from the head classes to the tail classes [17, 22, 34]. A series of works [27, 28, 30] present a learning using privileged information (LUPI) framework to transfer knowledge (e.g. similarity or margin) across multiple models which usually learned in different modalities. A teacher-teach-student scheme presented in the LUPI framework provides a relatively deeper way to use the side information to assist the original visual recognition task. Extending to this, [18, 29] unify the LUPI framework and the knowledge distillation paradigm [11] into a generalized distillation framework. These works provide a possible way to use the side information as a teacher model to help a visual recognition task (student model) in a teacher-teach-student or distillation scheme. However, the teacher-teach-student scheme requires the teacher model to be learned beforehand. And this two-stage approach is hard to be optimized globally. This scheme also implicitly demands the teacher model to be more powerful than the student model. In our proposed SICoT system, the side information participate into the visual classification task in conjunction with the visual feature in an end-to-end way. Moreover, no prior assumptions and constraints are made on the side information.

Large scale product recognition: Taking the applied techniques into account, Trax [3, 26] and MalongTech [25] have devoted their efforts to establish practical large scale visual recognition applications, especially the SKU (stock keeping unit [2]) level product visual recognition. MalongTech hosts a SKU level product dataset iMaterialist product 2019 and a corresponding competition in conjunction with FGVC6 workshop of CVPR2019 [13]. However, the iMaterialist product 2019 dataset covers only two thousands of product SKUs, and only provides image data without any side information for extending research. At Alibaba, in order to establish a scalable product recognition system, we organize a large scale SKU level product dataset that consists of about 60 million real-shot product images covering 1 million SKUs with the aid of the visual search engine Pailitao [1, 32]. Apart from images, the dataset contains abundant yet noisy side information (e.g. image titles, long descriptions and tags) provided by various sellers or customers in the marketplace of Alibaba. On the basis of the large scale SKU level product dataset, we settle a 30 million products recognition service driven by the proposed SICoT system in Pailitao. To our knowledge, this is the largest scale product recognition application in the E-commercial scenario so far.

3 APPROACH

In this section, we elaborate our proposed side information based co-training (SICoT) system in a large scale visual recognition task over long-tailed distributed training data. In Sec. 3.1, we firstly illustrate the overall product recognition process on the basis of the visual search service Pailitao [1, 32]. Considering that the side information of image titles from heterogeneous resources are often noisy, we then propose a bilinear word attention network in Sec. 3.2 to distinguish the important words from the noisy side information. Subsequently, a detailed illustration of the side information based

[2] https://en.wikipedia.org/wiki/Stock_keeping_unit

Figure 4: Two images of English setters from the Webvision2.0 [16] and two images of products from the marketplace of Alibaba. Along with each image, a short text description is also provided. These text descriptions are naturally noisy.

co-training system (SICoT) is given in Sec. 3.3. The SICoT system aims to leverage visual features and semantic embeddings to help transfer knowledge from the head classes to the tail classes in an end-to-end fashion.

3.1 Product Recognition Architecture

The entire product recognition architecture, as shown in Fig. 3, comprises an offline process and an online process, that following the present visual search architecture of Pailitao [1, 32] in general. The offline process mainly refers to the daily process of building product index using the proposed SICoT product recognition system. Unlike the index engine in the visual search service, the product index stores the SKU ids of products and corresponding product images, titles and tags for online retrieval. In the online process, the core function is a real time product recognition service in charge of predicting a SKU id for each query image. For the other modules in this architecture, we simply reuse the design of the visual search service, like category prediction and object localization. Once a query image is successfully recognized by the online service, a predicted SKU id will be obtained. By retrieving the index engine using the SKU id, the corresponding product image, title and tags will be obtained and presented to the customer, as shown in Fig. 2.

3.2 Bilinear word attention network

In many practical scenarios, besides images lots of related side information (e.g. image titles) can be obtained. These image related side information may reveal underlying similarity among images and classes, so as to it has great potential to improve a visual recognition task. In order to process both the visual information of images and the image related side information in an unified framework, we propose a bilinear word attention network, as shown in Fig. 5, to learn a semantic embedding from the noisy side information.

A conventional tokenization is firstly conducted on the side information of image titles, followed by using a word2vec model to generate a word embedding for each word after tokenization. In most of natural language processing (NLP) related tasks, such as language translation and image captioning, the order of words in a title matters in general. However, in our experimental observations, the order of words is less important and even harmful, especially when the side information is highly noisy. Here, we simply use an average pooling operation to generate a global embedding from the word embeddings instead of using a sequential model (e.g. recurrent neural networks). As for each word embedding, this global embedding can be regarded as a global context without consideration of the order of words in the side information.

Noise and meaningless words naturally occur in the image related side information due to the heterogeneous resources in both the marketplace of Alibaba and many open datasets. For example, as shown in Fig. 4, both the text descriptions of two dogs from Webvision2.0 and the titles of two soft drinks from the marketplace contain several words less relevant to the image content (e.g. *show*, *autumn*, *free shipping*). Considering this issue, we propose a soft attention sub-network to evaluate the importance of each word in the side information. A bilinear operation between the global context and all word embeddings is introduced to generate a second order feature map. An average pooling operation and a nonlinear transformation are then carried out above the feature map to output a word attention vector. A semantic embedding of the entire side information is achieved by the weighted sum of the word attention vector and all word embeddings at final.

3.3 Side information based co-training system

Given the bilinear word attention based semantic embedding presented in Sec. 3.2, we propose a visual feature and semantic embedding co-training scheme in this section. Due to the absence of image related side information once a recognition model has been deployed, the proposed co-training scheme is only involved in the training stage. The scheme is designed to take the semantic embedding from the side information as a bridge to transfer knowledge from the head classes to the tail classes. Unlike the teacher-teach-student paradigm used in LUPI [27], our approach makes no prior assumption about models, i.e. a teacher model should be more powerful than a student model. In fact, in our experimental observations, it is often inadequate to carry out a satisfactory classification by using the image related side information only, especially when the number of class are huge.

As illustrated in Fig. 1, we show the overall architecture of our proposed co-training scheme that consisting of three streams, i.e. a visual stream, a semantic stream and a mixed stream. The visual stream is a conventional visual recognition pipeline, which comprises a common convolutional neural network as a feature extractor and a plain softmax classifier optimized by a cross entropy loss. The visual stream is the target task that we attempt to improve. The semantic stream is simply the bilinear word attention based semantic embedding sub-network, in which the word embeddings are required initialized from a pretrained model like Word2vec [21]. Noted that it is assured that the visual feature and the semantic embedding take a same dimension through a deliberate design of

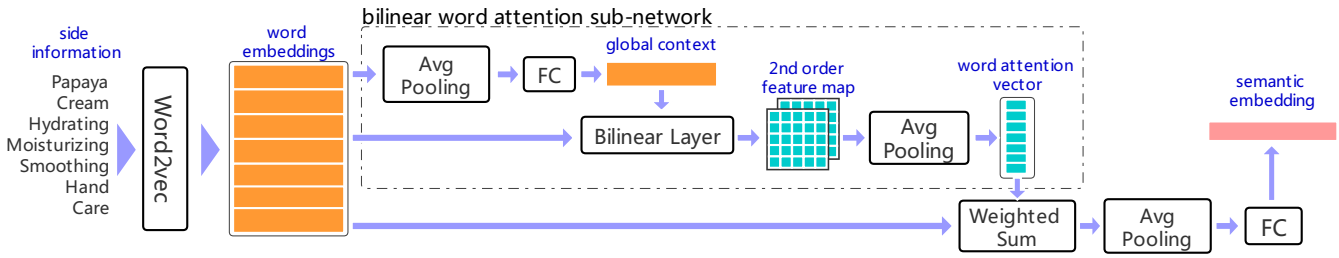


Figure 5: The semantic embedding network mainly consisting of a word2vec module and the proposed bilinear word attention module.

the network. A mixed or multi-modal feature is then achieved by a max-pooling operation over the visual features and the semantic embeddings. Unlike a conventional multi-task framework that consisting of multiple learning tasks driven by different objectives, the proposed co-training scheme makes both the visual features and the semantic embeddings to be learnt driven by a same classification task. As shown in Fig. 1, the visual feature x^{v} and the multi-modal feature x^{m} are followed by a shared co-training classifier optimized with the objective as Equ. 1, in which the $\hat{\mathrm{y}}_i^{\mathrm{v}}$ and $\hat{\mathrm{y}}_i^{\mathrm{m}}$ are the classifier output of the visual feature $\mathrm{x}_i^{\mathrm{v}}$ and the semantic embedding $\mathrm{x}_i^{\mathrm{m}}$, respectively.

$$Loss = -\frac{1}{N}(\sum_{i=1}^{N} \mathrm{y}_i \cdot \log(\hat{\mathrm{y}}_i^{\mathrm{v}})) - \lambda \cdot \frac{1}{N}(\sum_{i=1}^{N} \mathrm{y}_i \cdot \log(\hat{\mathrm{y}}_i^{\mathrm{m}})) \quad (1)$$

In the proposed training scheme, the shared classifier are learnt in a co-training fashion, in which the classification boundaries are directly affected both visually and semantically. It is well known that the classification boundaries in a classification task with long-tailed imbalanced training data are easily dominated by the head classes. The proposed co-training scheme may rectify the skewed classification boundaries by introducing the semantic knowledge into the classification training. Furthermore, it can be observed in Fig. 1 that the visual and semantic streams are tangled in not only the classifier training part, but also the feature learning part via the gradient backward procedure. Compared to the methods of taking the side information as weakly supervision and the two stage teacher-teach-student paradigm in the LUPI framework, our proposed co-training scheme provides a deeper way to tangle the visual and semantic knowledge in an end-to-end manner.

4 EXPERIMENTS

In this section, we evaluate our proposed side information based co-training approach on four large scale datasets that exhibiting long-tailed distribution, i.e. iMaterialist Product 2019 [13], Webvision2.0 selected 1k [16] and our proposed 34k and 1M SKU level product datasets, and demonstrate the positive effect of our approach on the long tail problem. We also report a relative gain of unique visitor (UV) conversion rate after settling our approach to the visual search application Pailitao [1] at Alibaba.

4.1 Datasets preparation

The WebVision2.0 [16] dataset is designed to facilitate the research on learning visual representation from noisy web data and it is attached with Google and Flickr retrieval results that containing titles and detailed text descriptions. In this paper, we extract the title of each image as the side information for experiments. We preprocess these side information by discarding the punctuation marks and 5 percent of very frequent and very infrequent words, and throwing away those images without any side information. Eventually, one thousand classes are randomly picked out from the entire 5 thousand of classes in Webvision2.0 [16]. iMaterialist Product 2019 [13] hosted by MalongTech [25] is a SKU level product dataset that consisting of 2019 product SKUs. Because there is no any image related side information provided by Materialist Product 2019, we take each image as an input query of the image search engine Pailitao [1] and collect the title of the top1 search result as the side information. Apart from the two open datasets, we also provide two SKU level product datasets in this paper, a facial cream and mask product dataset that containing 34 thousands classes and a huge scale product dataset that containing one million classes. The two proposed datasets come from the marketplace of Alibaba. An overview of statistics about the four datasets are illustrated in Table 1. And the long-tailed distribution of training data are shown in Fig. 6, in which the boundaries between the head and tail classes on training data are experimentally determined as 200, 5000, 100 and 200, respectively. About the testing sets in the four datasets, each class contains approximately equal number of images for a fair evaluation.

4.2 Evaluation metrics

Considering that many classes are conceptually overlapped and ambiguous, especially in the Webvision2.0 and the SKU level product datasets, we report the results of $top1$ and $top3$ predicted labels for the evaluation of recognition performance. In addition, the evaluation of overall $top1$ and $top3$ accuracies are also conducted over the head and tail classes, respectively, to demonstrate the effect on the long tail problem.

4.3 Implementation details

In the proposed co-training scheme, we use a Resnet-50 [10] initialized from a ImageNet pretrained model as the backbone convolutional neural network (CNN) in the visual stream. As shown

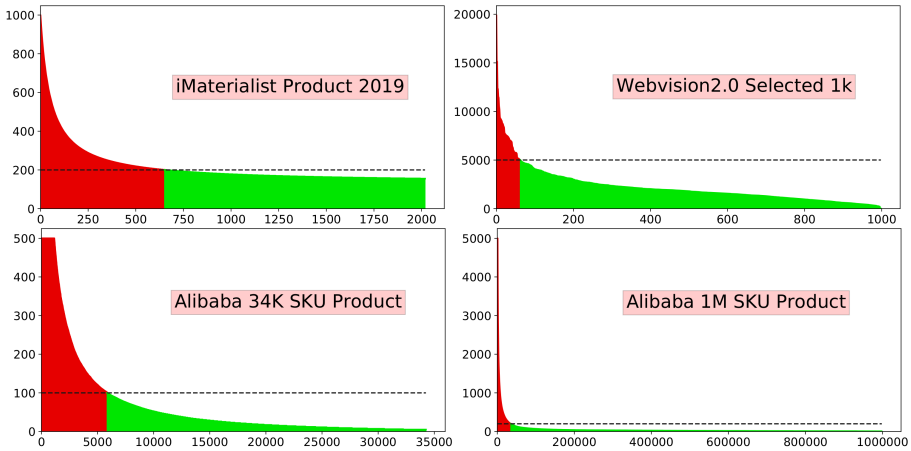


Figure 6: Long-tailed distribution of training set of the four datasets. The red and green parts in each panel represent the head and tail classes, respectively.

	Class	Trainset	Testset	Vocab
iMaterialist Product 2019	2019	440438	9986	175208
Webvision2.0 Selected 1k	1000	2230968	59040	162541
Alibaba 34K SKU Product	34258	2305853	171290	19762
Alibaba 1M SKU Product	998131	55776960	7675690	345656

Table 1: Statistics of the four datasets, in which the "Vocab" means the number of words extracted from the side information of image titles after a tokenization process.

in Fig. 1, the union of the visual stream and co-training classifier in the architecture represents the baseline that carrying out a conventional classifier using image data only. The word embeddings in the semantic stream are initialized using the *word_embedding*() API of Alibaba NLP toolbox (AliNLP) [5]. The word embeddings of AliNLP are trained using the product titles from the marketplace of Alibaba, and well performs on many natural language processing tasks under the E-commercial circumstance at Alibaba.

When training on such a large dataset as our proposed product dataset that containing one million classes, the size of the last fully connected (FC) layer will be larger than the memory size of a single block of GPU (e.g. Nvidia V100 32G). The proposed co-training system is implemented in a hybrid parallel training framework [23], in which the last FC layer is divided and sent to M GPUs for distributed training. The training is carried out in a distributed computing platform of Alibaba with 60 blocks of Nvidia P100 GPUs. For a fair comparison, the baseline and the co-training approach are trained using a stochastic gradient descent (SGD) optimizer with a same learning configuration of a batch size 256, an initial learning rate 0.1 and a step decay policy of step 1, gamma 0.8.

4.4 Experimental results

4.4.1 Overall classification accuracy. As illustrated in Table 2, we report the *top*1 and *top*3 (enclosed in parentheses) classification accuracies on the four datasets. It is clearly observed that our proposed co-training approach using the side information of image titles shows performance improvements against the baselines with significant *top*1 (and *top*3) accuracies gain by 1.01%(1.68%) in iMaterialist Product 2019, 3.27%(1.91%) in Webvision2.0 Selected 1k, 1.87%(1.85%) in Alibaba 34K SKU Product and 2.00% (0.86%) in Alibaba 1M SKU Product. When the number of classes ranging from 1 thousands to 1 millions, our approach achieves consistent and significant accuracy gains all the time. The consistent improvement of classification performance demonstrates an attractive scalability for setting out practical visual recognition applications.

4.4.2 Effect of the bilinear word attention based semantic embedding. For a qualitative evaluation of the proposed bilinear word attention based semantic embedding, as illustrated in Fig. 7, we visualize the learned word attention vectors of several title samples. The titles of the four examples are demonstrated in original Chinese and translated English at the same time. These words are displayed in the descending order of the corresponding value in the attention vector. In each example, three most important words are highlighted in green and three most unimportant words are highlighted in red. It is observed that the words with larger attention weights generally refer to product names and brand names, and the words with smaller attention weights are often less helpful to distinguish the product from other products. For example, the promotion words of "discount" and "new arrival" are irrelevant to identifying a product.

	Baseline	Co-training	Gain
iMaterialist Product 2019	57.29 (83.77)	58.30 (85.45)	**1.01 (1.68)**
Webvision2.0 Selected 1k	62.68 (79.02)	65.95 (80.93)	**3.27 (1.91)**
Alibaba 34K SKU Product	51.60 (77.04)	53.47 (78.89)	**1.87 (1.85)**
Alibaba 1M SKU Product	88.25 (97.13)	90.25 (97.99)	**2.00 (0.86)**

Table 2: The $top1$ and $top3$ (in parentheses) classification accuracies and performance improvements compared to the baselines on the four datasets.

Figure 7: Visualization of learned attention word vectors on four examples. Each example consists of a image, an original image title in Chinese and a series of words with corresponding attention value in both Chinese and English. These words are displayed in the descending order of the corresponding value in the attention vector. In each example, three most important words are highlighted in green and three most unimportant words are highlighted in red.

Meanwhile, as shown in Tab. 3, we compare our proposed bilinear word attention based embedding with several other methods on the proposed Alibaba 1M SKU product dataset. The method of Alinlp embeddings mean pooling simply takes the mean of the Alinlp word embeddings as the final title embedding. The bi-LSTM method takes advantage of a bidirectional long short-term memory (LSTM) module to generate a final title embedding. The bi-LSTM attention method introduces a self attention mechanism into the bi-LSTM method. It is observed that our proposed bilinear word attention based embedding outperforms the other four methods. The two bi-LSTM based methods take the order of words into consideration by using a sequential model bi-LSTM to describe the context. However, the inferior performance of the two methods indicate that the order of words is less helpful to carrying out a classification task. Compared to the Alinlp embeddings mean pooling method, our proposed attention method has a positive effect on boosting the performance of classification by using the bilinear word attention model.

Methods	Top1 Accuracy	Gain
baseline	88.25	-
Alinlp embeddings mean pooling	89.14	**0.89**
bi-LSTM	89.13	**0.88**
bi-LSTM attention	88.62	**0.37**
bilinear word attention	89.41	**1.16**

Table 3: Comparison of our proposed bilinear word attention based embedding with other four methods on the proposed Alibaba 1M SKU Product dataset.

4.4.3 Effect on long-tailed distribution of training data. As shown in Table 4, we illustrate the effect of our proposed co-training scheme on the problem of long-tailed distributed training data. Compared with the baseline on the four datasets, we report the averaged top1 and top3 classification accuracies of the head and tail classes, respectively. It is observed that our approach achieves improvement

		#Training Samples	Baseline	Co-training	Gain
iMaterialist Product 2019	Head	200 - max	57.38 (83.49)	58.37 (85.08)	**0.99** (**1.59**)
	Tail	1 - 200	37.21 (81.40)	41.86 (83.72)	**4.56** (**2.32**)
Webvision2.0 Selected 1k	Head	5000 - max	55.18 (72.91)	55.86 (72.55)	**0.68** (**-0.36**)
	Tail	1 - 5000	63.04 (79.26)	66.23 (81.11)	**3.19** (**1.85**)
Alibaba 34K SKU Product	Head	100 - max	62.98 (84.94)	63.54 (85.51)	**0.56** (**0.57**)
	Tail	1 - 100	49.24 (75.40)	51.38 (77.51)	**2.14** (**2.11**)
Alibaba 1M SKU Product	Head	200 - max	95.09 (99.53)	95.61 (99.62)	**0.52** (**0.09**)
	Tail	1 - 200	86.87 (96.64)	89.16 (97.66)	**2.29** (**1.02**)

Table 4: The averaged $top1$ and $top3$ (in parentheses) classification accuracies and performance improvements on the head classes and the tail classes, repetitively.

of the top1 and top3 classification accuracies in both the head classes and the tail classes. Noted that the performance gains in the head classes are more significant than the gains in the tail classes. This phenomenon indicates that our approach improves performance of the tail classes while does not harm the performance of the head classes. In fact, our approach often can slightly improve the performance of the head classes at the same time. This mainly owes to that the procedure of knowledge transfer is bidirectional in the proposed co-training system, and the knowledge extracted from the tail classes is also beneficial to the training of the head classes.

4.4.4 Application in Pailitao. Pailitao [1] is a visual search application which aims to assisting customers to find the same or similar products by a mobile phone camera shot image. Pailitao is still experiencing swift growth of daily active users (DAU). In Pailitao, the unique visitor (UV) conversion rate is a common measurement which is calculated as Equ. 2.

$$\text{UV conversion rate} = \frac{\text{number of trading UV}}{\text{number of visiting UV}} \tag{2}$$

To further improve the UV conversion rate of Pailitao, a huge scale SKU level product recognition service is settled in Pailitao as an upgrade. As shown in Fig. 2, the recognition result of a query is displayed at the top panel of the page in conjunction with the results of visual search. The panel includes a clickable image, a short title and several tags of the recognized product. The product recognition service is constructed by using the proposed side information based co-training system. Compared to the original version of Pailitao, there is a relative 3.1 percent gain of daily UV conversion rate after this upgrade.

5 CONCLUSION

The phenomenon of long-tailed imbalanced training data naturally occurs under the E-commercial circumstance and challenges the performance of a large scale visual recognition task. In this paper we address the problem of long-tailed distributed training data by exploring a side information based visual recognition co-training (SICoT) system. We firstly introduce a bilinear word attention subnetwork to distinguish important words from the noisy side information, followed by generating a semantic embedding as the distilled knowledge of the side information. An end-to-end visual feature and semantic embedding co-training system is then proposed to help transfer knowledge from the head classes to the tail classes. Experimental results on four large scale datasets demonstrate the effectiveness of the proposed approach. Our approach improves the performance of the tail classes without any harm to the head classes. Moreover, a SICoT driven product visual recognition service is settled in Pailitao and achieves a significant gain of unique visitor conversion rate. Our approach has shown good scalability ranging from medium to large scale datasets, and it is of great value for establishing industrial visual recognition applications.

REFERENCES

[1] Alibaba. [n.d.]. PaiLiTao. http://www.pailitao.com.
[2] Cristiano L Castro and Antônio P Braga. 2013. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems* 24, 6 (2013), 888–899.
[3] Daniel Shimon Cohen, Yair Adato, and Dolev Pomeranz. 2019. Method and a system for object recognition. US Patent 10,402,777.
[4] Charles Corbiere, Hedi Ben-Younes, Alexandre Rame, and Charles Ollion. 2017. Leveraging weakly annotated data for fashion image retrieval and label prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2268–2274.
[5] Alibaba DAMO. [n.d.]. AliNLP. https://damo.alibaba.com/labs/language-technology?lang=en.
[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
[7] Qi Dong, Shaogang Gong, and Xiatian Zhu. 2017. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 1851–1860.
[8] Q. Dong, S. Gong, and X. Zhu. 2019. Imbalanced Deep Learning by Minority Class Incremental Rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 6 (June 2019), 1367–1381.
[9] E. A. Garcia and H. He. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21, 09 (sep 2009), 1263–1284. https://doi.org/10.1109/TKDE.2008.239
[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[12] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. 2016. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*. Springer, 67–84.
[13] kaggle. [n.d.]. iMaterialist Challenge on Product Recognition. https://www.kaggle.com/c/imaterialist-product-2019.
[14] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems* 29, 8 (2017), 3573–3587.
[15] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. 2018. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982* (2018).
[16] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, Jesse Berent, Abhinav Gupta, Rahul Sukthankar, and Luc Van Gool. 2017. WebVision Challenge: Visual Learning and Understanding With Web Data. *Arxiv Preprint* (2017).

[17] Joseph J Lim, Russ R Salakhutdinov, and Antonio Torralba. 2011. Transfer learning by borrowing examples for multiclass object detection. In *Advances in neural information processing systems*. 118–126.
[18] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. 2015. Unifying distillation and privileged information. arXiv:1511.03643 [stat.ML]
[19] Yunshan Ma, Xun Yang, Lizi Liao, Yixin Cao, and Tat-Seng Chua. 2019. Who, Where, and What to Wear? Extracting Fashion Knowledge from Social Media. In *Proceedings of the 27th ACM International Conference on Multimedia*. 257–265.
[20] T. Maciejewski and J. Stefanowski. 2011. Local neighbourhood extension of SMOTE for mining imbalanced data. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. 104–111. https://doi.org/10.1109/CIDM.2011.5949434
[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[22] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. 2011. Learning to share visual appearance for multiclass object detection. In *CVPR 2011*. IEEE, 1481–1488.
[23] Liuyihan Song, Pan Pan, Kang Zhao, Hao Yang, Yiming Chen, Yingya Zhang, Yinghui Xu, and Rong Jin. 2020. Large-Scale Training System for 100-Million Classification at Alibaba (under review). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (San Diego, California USA) *(KDD '20)*. ACM, New York, NY, USA, 993–1001.
[24] Y. Tang, Y. Zhang, N. V. Chawla, and S. Krasser. 2009. SVMs Modeling for Highly Imbalanced Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 1 (Feb 2009), 281–288. https://doi.org/10.1109/TSMCB.2008.2002909
[25] Malong Technologies. [n.d.]. ProductAI. https://www.productai.com/home.
[26] Trax. [n.d.]. Traxretail. https://traxretail.com/.
[27] Vladimir Vapnik and Rauf Izmailov. 2015. Learning Using Privileged Information: Similarity Control and Knowledge Transfer. *Journal of Machine Learning Research* 16, 61 (2015), 2023–2049. http://jmlr.org/papers/v16/vapnik15b.html
[28] Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural Networks* 22, 5 (2009), 544 – 557. https://doi.org/10.1016/j.neunet.2009.06.042 Advances in Neural Networks Research: IJCNN2009.
[29] Weiran Wang. 2019. Everything old is new again: A multi-view learning approach to learning using privileged information and distillation. *arXiv preprint arXiv:1903.03694* (2019).
[30] X. Yang, M. Wang, and D. Tao. 2018. Person Re-Identification With Metric Learning Using Privileged Information. *IEEE Transactions on Image Processing* 27, 2 (Feb 2018), 791–805.
[31] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. 2017. Range Loss for Deep Face Recognition with Long-Tailed Training Data. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 5419–5428. https://doi.org/10.1109/ICCV.2017.578
[32] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. 2018. Visual Search at Alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 993–1001. https://doi.org/10.1145/3219819.3219820
[33] Zhi-Hua Zhou and Xu-Ying Liu. 2006. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18, 1 (Jan 2006), 63–77. https://doi.org/10.1109/TKDE.2006.17
[34] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. 2014. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 915–922.