E^2BoWs: An End-to-End Bag-of-Words Model via Deep Convolutional Neural Network for Image Retrieval

Xiaobin Liu[a], Shiliang Zhang[a], Tiejun Huang[a], Qi Tian[b]

[a] *Peking University, Beijing, China*
[b] *Department of Computer Science, University of Texas at San Antonio, USA*

Abstract

Traditional Bag-of-Words (BoWs) model is commonly generated with many steps, including local feature extraction, codebook generation and feature quantization, *etc*. Those steps are relatively independent with each other and are hard to be jointly optimized. Moreover, the dependency on hand-crafted local feature makes BoWs model not effective in conveying high-level semantics. These issues largely hinder the performance of BoWs model in large-scale image applications. To conquer these issues, we propose an End-to-End BoWs (E^2BoWs) model based on Deep Convolutional Neural Network (DCNN). Our model takes an image as input, then identifies and separates semantic objects in it, and finally outputs visual words with high semantic discriminative power. Specifically, our model firstly generates Semantic Feature Maps (SFMs) corresponding to different object categories through convolutional layers, then introduces Bag-of-Words Layers (BoWL) to generate visual words from each individual feature map. We also introduce a novel learning algorithm to reinforce the sparsity of the generated E^2BoWs model, which further ensures the time and memory efficiency. We evaluate the proposed E^2BoWs model on several image search datasets including *MNIST*, *SVHN*, *CIFAR-10*, *CIFAR-100*, *MIRFLICKR-25K* and *NUS-WIDE*. Experimental results show that our method achieves promising accuracy and efficiency compared with recent deep

Email addresses: xbliu.vmc@pku.edu.cn (Xiaobin Liu), slzhang.jdl@pku.edu.cn (Shiliang Zhang), tjhuang@pku.edu.cn (Tiejun Huang), qitian@cs.utsa.edu (Qi Tian)

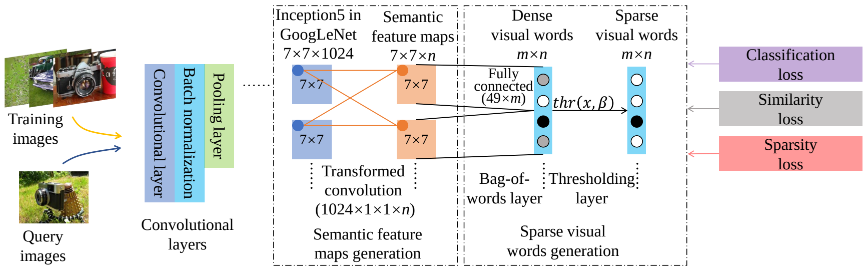


Figure 1: Framework of the proposed E^{2}BoWs model. The structure of our deep model is identical to the one of GoogLeNet [1] with BN [2] till the Inception5 layer. The output size of Inception5 is $7 \times 7 \times 1024$. Pool5 in GoogLeNet [1] is discarded. The n-way output layer is transformed into a convolutional layer to generate n semantic feature maps. m sparse visual words are then generated by bag-of-words layer from each individual semantic feature map, resulting in $m \times n$ visual words. Finally, a three-component loss function is applied for training the model.

learning based retrieval works.

Keywords: Large-scale Image Retrieval, Bag-of-words Model, Deep Convolutional Neural Network

1. Introduction

A huge number of images are being uploaded to the Internet every moment, and each image commonly conveys rich information. This makes Content-Based Image Retrieval (CBIR) a challenging and promising task. Bag-of-Words (BoWs) model, which considers an image as a collection of visual words, has been widely applied for large-scale image retrieval [3, 4]. Conventional BoWs model is computed with many stages, *e.g.*, feature extraction, codebook generation, and feature quantization [3, 5, 6, 7]. Then inverted file index and Term Frequency-Inverse Document Frequency (TF-IDF) strategy can be used for indexing and retrieval. Since the number of visual vocabulary is commonly quite large, *e.g.*, 1 million in [3], and an image only contains a small number of visual words, indexes generated by BoWs model are sparse and thus ensure the high retrieval efficiency.

Most of existing BoWs models are based on hand-crafted local features, *e.g.*, SIFT [8]. These models have shown promising performance in large-scale partial-duplicate image retrieval [3, 5, 6]. However, as the local descriptor cannot effectively describe high-level semantics, *i.e.*, commonly known as the "semantic gap" issue, BoWs models built on local descriptors always fail to address the semantic similar image retrieval task [9]. Although some works have been proposed to conquer this issue [10, 11, 12, 13, 4], most of these works introduce extra computations and memory overheads.

Recent years have witnessed a lot of breakthroughs in end-to-end deep learning model for vision tasks. After AlexNet [14] achieving the best performance in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), Deep Convolutional Neural Network (DCNN) has been applied to various vision tasks, including image classification [1, 15], object detection [16, 17], semantic segmentation [18] and many others [19, 20, 21, 22, 23, 24, 25]. Generally, DCNN based models yield much better performance on these tasks compared with conventional methods. Most of DCNNs consist of a set of convolutional layers and Fully Connected (FC) layers [14, 1, 15]. It is found that convolutional layers can extract high-level semantic cues from pixel-level input [26], and hence provide a possible solution to solve the "semantic gap" issue. Therefore, it is straightforward to leverage DCNN in CBIR [9]. Some works use DCNN to generate hash codes and yield promising performance [27, 28, 29, 30, 31]. However, there is still a lack of research efforts on DCNN based BoWs model, which could be integrated with inverted file indexing and TF-IDF weighting for large-scale image retrieval.

Targeting to leverage the efficiency of BoWs model and the semantic learning ability of DCNN models in large-scale semantic similarity based image retrieval, we propose to generate a novel DCNN based End-to-End BoWs (E^2BoWs) model as shown in Fig. 1. Structure of our E^2BoWs model coincides with GoogLeNet [1] with Batch Normalization (BN) [2] up to Inception5. We discard Pool5 layer and transform the last FC layer into a convolutional layer to generate Semantic Feature Maps (SFMs) corresponding to different object categories. Different with conventional feature maps in DCNN based models that contain latent semantics, SFMs have clear semantic cues corresponding to object categories. This makes it potential to generate semantic visual words from each SFM. Then a Bag-of-Words Layer (BoWL) and a Thresholding Layer are introduced to generate sparse visual words from each semantic feature map. Instead of using fully connected layers to generate features from feature maps, we use proposed BoWL to generate visual words from each SFM individually. This ensures the resulting visual words to preserve clear semantic cues. Instead of building up a large vocabulary tree to ensure the sparsity of BoWs models, we introduce the Thresholding layer to learn a threshold w.r.t different datasets to further ensure the feature sparsity and high retrieval efficiency. Finally, a novel three-component loss function is designed to ensure: 1) fast convergence of the training procedure, 2) similar images sharing more visual words, and 3) high sparsity of the generated E^{2}BoWs model, respectively.

In semantic similarity based image retrieval scenario, the proposed method has several advantages compared with traditional BoWs models: 1) Instead of using hand-crafted features in traditional models, incorporating DCNN into BoWs model is potential to bring higher discriminative power in aspect of semantics and provides a better solution for semantic similar image search task. 2) Instead of being generated with several independent steps like feature extraction, codebook generation and feature quantization, our E^{2}BoWs model is generated in an end-to-end manner based on DCNN. Thus, the model is more efficient and easier to be jointly optimized and tuned. Our E^{2}BoWs model also shows advantages over traditional hashing methods: 1) Generated visual words convey clear semantic cues such as object categories, which will be evaluated in Sec. 4.7. 2) Instead of generating short and dense hash codes, we generate longer visual words to obtain better discriminative power. The visual words are also sparse to ensure the efficiency of retrieval. We evaluate the proposed E^{2}BoWs model on several image search datasets including *MNIST* [32][1], *CIFAR-10* [33][2], *CIFAR-100* [33][3], *SVHN* [34][4], *MIRFLICKR-25K* [35][5] and *NUS-WIDE* [36][6]. Comparisons with recent deep learning based image retrieval works show that our method achieves promising accuracy and efficiency.

The rest of this paper is organized as follows: Section 2 discusses some works related to proposed model. Section 3 presents our model in detail. Section 4 evaluates the proposed model on different datasets and Section 5 gives our conclusions.

2. Related Work

As a fundamental task in multimedia content analysis and computer vision [37, 9, 38], CBIR aims to search for images similar with the query in an image gallery. Since directly computing similarity between two images with raw image pixels is infeasible, BoWs model is widely used as an image representation for large-scale image retrieval, such as [3, 11, 4, 39, 12]. In the following parts, we will introduce related works on BoWs model and DCNN based image retrieval, respectively.

2.1. BoWs Model

Over the past decade, various BoWs models [3, 5, 6, 7] have been proposed based on local descriptors, such as SIFT [8] and SURF [40]. Those BoWs models have shown promising performance in large-scale image retrieval. Conventional BoWs models consider an image as a collection of visual words and are generated by many stages, *e.g.*, feature extraction, codebook generation and feature quantization [3, 5, 6, 7]. For instance, Nister *et al.* [3] extract SIFT [8] descriptors from MSER regions [41] and then hierarchically quantize SIFT descriptors

[1] http://yann.lecun.com/exdb/mnist/
[2] http://www.cs.toronto.edu/ kriz/cifar.html
[3] http://www.cs.toronto.edu/ kriz/cifar.html
[4] http://ufldl.stanford.edu/housenumbers/
[5] http://press.liacs.nl/mirflickr/
[6] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

by vocabulary tree. As individual visual word cannot depict the spatial cues in images, images sharing visual words may have different spatial relationship among patches. To enhance the discriminative ability of visual words, some works combine visual words with spatial information [39, 42, 43] to make the resulting BoWs model contain more spatial cues. Wang and Jiang [44] present a new challenging dataset *INSTRE* to promote the development of CBIR. They evaluate several models on *INSTRE* and integrate several methods into a simple yet efficient model.

Though conventional visual words show promising performance in partial-duplicate image retrieval, the dependency on hand-crafted local feature hinders their ability to convey semantic cues. This is the consequence of the "semantic gap" between low-level pixel information and high-level semantics. For instance, two objects from different categories might share similar local features, which can be quantized to same visual words in the vocabulary tree and thus increases the similarity between them.

Some works have been proposed to enhance the discriminative power of BoWs model in aspect of semantic cues [10, 11, 12, 13, 4]. Perronnin [10] computes a histogram for each class using class-specific vocabulary and universal vocabulary. And a set of classifiers can be trained based on these histograms for each category. Then images can be classified by these classifiers. Lazebnik *et al.* [11] jointly quantize continuous features and class labels by an alternating minimization procedure, making the quantized representation more discriminative to class information. Wu *et al.* [12] propose an off-line distance metric learning scheme to make semantically related features mapped to the same visual words, resulting in an optimized codebook for semantic retrieval. Wu *et al.* [13] present an on-line metric learning algorithm to improve the BoWs model by optimizing the proposed semantic objective function. Zhang *et al.* [4] propose a method to co-index semantic attributes into inverted index generated by local features, which makes the index convey more semantic cues. However, most of these works need extra computations either in the off-line indexing or on-line retrieval stages. Moreover, since these models are generated by independent steps, they are hard to be jointly optimized to achieve better efficiency and accuracy.

Therefore, we propose the E^{2}BoWs model to generate visual words in an end-to-end model based on DCNN. We incorporate DCNN into proposed E^{2}BoWs model to bring high discriminative power in the aspect of semantics. And we train the model in an end-to-end manner to jointly optimize the feature extraction and visual words generation.

2.2. DCNN Based Model

Recently, many works try to leverage DCNN in image retrieval [9, 45, 28] to better conquer the "semantic gap" issue. Babenko *et al.* [45] adopt DCNN on instance retrieval and show that fine-tuning the model on the test domain can boost the retrieval performance. Wan *et al.* [9] propose three schemes to apply features generated by DCNN in CBIR: 1) Directly using the features from the DCNN model pre-trained on *ImageNet* [46]. 2) Refining the features by metric learning. 3) Retraining the entire model on the target dataset and then using the features for retrieval. They prove that features extracted from DCNN can significantly outperform hand-crafted features after being fine-tuned. However, they don't consider the retrieval efficiency when apply the real-value features in large-scale datasets. Xia *et al.* [28] introduce a DCNN based hashing method for fast and accurate image retrieval. The method consists of two steps: 1) Generating hash codes on training set by an iterative algorithm. 2) Learning a hash function based on DCNN to fit the hash codes generated in step 1. Though they achieve better performance than conventional hashing methods, the independence of two steps hinders the joint optimization of the entire DCNN model.

Some works then apply DCNN on CBIR in an end-to-end manner to jointly optimize feature extraction and hash codes generation [29, 31, 57]. Lin *et al.* [29] propose a framework to generate hash codes directly by end-to-end training with a classification object function, which achieves better performance than previous methods. Liu *et al.* [31] propose a unified framework to generate hash codes that preserve both category and attribute similarity relationship. And the framework is also trained with classification objective function. They show that features in deep model trained for classification task can be adopted for CBIR task directly, and the end-to-end training can boost the performance of hash codes.

However, features in model trained for classification task are not optimal for retrieval task. In classification task, features are trained to be correctly categorised by classifier. However, correctly categorized features of different categories could be close to each other in Euclidean or Hamming distance. In retrieval task, features should preserve similarity relationship among images, *i.e.*, features of images in same category should be close to each other, and vice versa. To directly optimize hash codes for retrieval, contrastive loss [27, 47] and triplet loss [48, 9] are used to optimize the hash model. These algorithms aim to minimize intra-class distance and maximize inter-class distance of features. So that hash codes will preserve semantic relations among images and be more suitable for retrieval.

In these aforementioned methods, real-value hash codes are learned during training. Then hash codes are quantized into binary codes for retrieval. Hence quantization error will be involved in quantization step. Moreover, using different distance metrics for training and retrieval, *e.g.*, Euclidean distance for training and Hamming distance for retrieval, will also bring approximation error. Different from previous works in quantizing hash codes after training, Liu *et al.* [27], Zhu *et al.* [47] and Jain *et al.* [55] propose new training algorithms respectively to enforce the networks to output binary-like hash codes during training. They reduce quantization error and approximation error, and thus outperform previous methods.

So far, most of deep learning based retrieval works focus on generating and optimizing hashing codes. There still lack research efforts in generating DCNN based BoWs model. It is promising to generate a discriminative BoWs model directly from an end-to-end DCNN based model and leverage the scalability of BoWs model for large-scale image retrieval.

3. Proposed Method

We propose E^2BoWs model to generate visual words in an end-to-end manner based on DCNN. Given an input image $\mathcal{I}_i$, a vector of visual words v_i is generated directly by proposed model: $v_i = \mathcal{F}(\mathcal{I}_i, \theta)$, where $\mathcal{F}$ is the mapping function of proposed model and θ is parameters in E^2BoWs model.

We design E^2BoWs model by modifying the GoogLeNet [1] with BN [2]. As shown in Fig. 1, the structure of our deep model is identical to the one of GoogLeNet [1] with BN [2] before the Inception5 layer. Most of previous works extract features for retrieval from FC layers. Differently, we propose to learn features from feature maps which preserve more visual cues than FC layers. We thus transform the last n-way FC layer into a convolutional layer to generate n SFMs corresponding to n training categories, which convey both semantics and visual details. Then, m sparse visual words are generated from each individual SFM by Bag-of-Words Layer (BoWL) and thresholding layer, resulting in $m \times n$ visual words. So that the mapping function is composed of a set of convolutional layers, SFMs generation and visual words generation. Finally, a three-component loss function is applied to train the model. In the following parts, we present the details of the network structure, model training and generalization ability improvement.

3.1. Semantic Feature Maps Generation

In GoogLeNet [1], the output layer conveys semantic cues because the label supervision is directly applied on it. However, the output layer losses certain visual details of the images, such as the location and size of objects, which could be beneficial in image retrieval. Meanwhile, Inception5 contains more visual cues than semantics. Learning visual words from the output layer or Inception5 may loss discriminative power to either visual details or semantic cues. To preserve both semantics and visual details, we propose to generate Semantic Feature Maps (SFMs) from Inception5 and generate visual words from SFMs.

SFMs is generated by transforming the parameters in FC layers into a convolutional layer. This transformation is illustrated in Fig. 2. The size of pa-

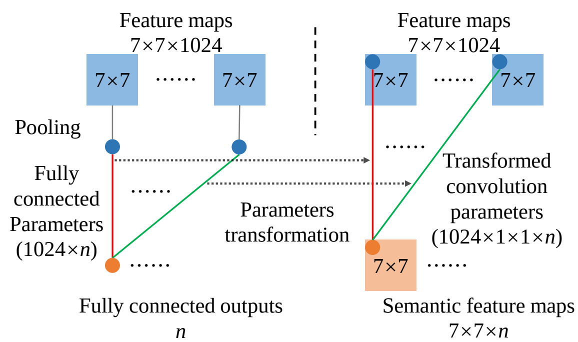


Figure 2: Illustration of transforming parameters of FC layer into a convolutional layer to generate SFMs. Lines in same color indicate the same parameters.

rameters in the FC layer is $1024 \times n$, where 2014 is the feature dimensionality after pooling and n is the number of training categories. Those parameters can be reshaped into n convolutional kernels of size $1024 \times 1 \times 1$. In other words, we transform parameters corresponding to each output in FC layer of size 1024×1 into a convolutional kernel of size $1 \times 1 \times 1024$. Therefore, n-channels of convolutional kernel can be generated. Accordingly, n SFMs can be generated after Inception5.

In FC layers, each output is a classification score for an object category. Compared with the output of FC layer, SFMs also contain such classification cues. For example, average pooling the activation on each SFM gets the classification score for the corresponding category. Moreover, SFMs preserve certain visual cues because they are produced from Inception5 without pooling.

We illustrate examples of SFMs in Fig. 3. Three images with the same label "elkhound" in *ImageNet* [46] and their SFMs with the top-4 largest response values are illustrated. It can be observed that, the illustrated SFMs show 75% overlap among the three images. SFM #175 constantly shows the strongest activation. This means the activation values of SFMs represent the semantic and category cues. Moreover, the location and size of object are also presented by SFMs.

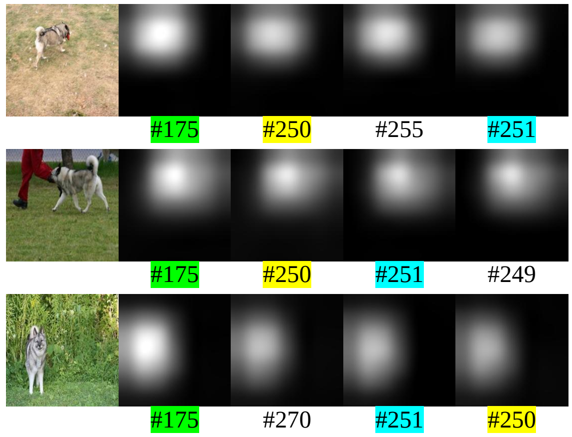


Figure 3: Visualization of some SFMs. Input images are in the first column. The rest are SFMs with top-4 largest response values. The number under each SFM denotes its unique ID in all SFMs. The same IDs are highlighted with the same color.

3.2. Sparse Visual Words Generation

Because different SFMs correspond to different object categories, they are potential to identify and separate the objects in images. Those characteristics make SFMs more suitable to generate visual words that convey both semantic and visual cues. To preserve the spacial and semantic cues in SFMs, we introduce a Bag-of-Words Layer (BoWL) to generate sparse visual words directly from each individual SFM. To reinforce the sparsity of generated visual words, we introduce a Threshold Layer to discard visual words whose value are smaller than a threshold.

3.2.1. Bag-of-Words Layer

Specifically, a local FC layer with ReLU [49] is used to generate m visual words from each individual SFM. This strategy finally generates $m \times n$ visual words. Each local FC layer is trained independently. Compared with traditional FC layer, local FC layer better preserves semantic and visual cues in each SFM. Specially, visual words generated from different SFMs will convey certain semantic cues corresponding to different object categories. And the local FC layer introduces less parameters to learn. For example, BoWL needs $49 \times m \times n$

parameters, while a FC layer following a pooling layer needs $(49 \times n) \times (m \times n)$ parameters. Less parameters will make the model easy to train and will also reduce the risk of overfitting.

It should be noted that we discard SFMs with negative average active values during visual words generation. Because the average active values of SFMs reflect the corresponding classification score, negative average active values indicate the absence of corresponding objects in the input image. Therefore, those SFMs shouldn't be involved in visual words generation. Moreover, discarding those SFMs reduces the number of nonzero visual words and improves the efficiency for indexing and retrieval.

Generated SFMs contain both semantic and spatial information of objects contained in input images. The generated visual words from each SFM would also convey those cues. Note that, our generated visual word is different from traditional visual word [3] generated from local features and is not designed to conduct partial duplicated image retrieval. However, it is an interesting problem how to encode more local details and part cues into the visual words generated by DCNN. Recently, some region proposal methods and objective detection methods [17, 16] have been proposed to locate objects in images. Further considering the object parts or foregrounds may improve the discriminative power of resulting visual words.

3.2.2. Thresholding Layer

The generated visual words are L2-normalized for inverted file indexing and retrieval. Our experiments show that, there commonly exist many visual words with small response values, *e.g.*, 1*e*-3. During online retrieval, those visual words won't contribute much to the similarity computation. Moreover, they are harmful to the sparsity of the BoWs model and would make more images embedded in inverted lists, resulting in more memory and time overhead. We find that discarding such visual words, dramatically improves the retrieval efficiency without degrading the accuracy to much. This procedure is formulated

Table 1: Retrieval efficiency and accuracy on *CIFAR-100* [33] testing set with different thresholds.

Threshold	0	0.05	0.06	0.07	0.08	0.09
mAP	0.697	0.686	0.689	0.693	0.697	0.700
ANV	409.0	50.4	36.7	28.4	23.0	19.0
ANI	4090	500	370	280	230	190
ANO	$1,672,810$	$25,200$	$13,579$	$7,952$	$5,290$	$3,610$
Threshold	0.10	0.11	0.13	0.13	0.14	0.15
mAP	0.703	**0.704**	0.703	0.700	0.693	0.684
ANV	16.8	15.0	13.5	12.3	11.4	10.6
ANI	170	150	140	120	110	110
ANO	$2,856$	$2,250$	$1,890$	$1,476$	$1,254$	$1,166$

as follows:

$$thr(x,\beta) = \begin{cases} x, & x > \beta, \\ 0, & otherwise, \end{cases} \tag{1}$$

where β denotes the threshold and x denotes the response value of a visual word.

We evaluate this procedure on the testing set of *CIFAR-100* [33] with different thresholds. We measure the retrieval performance by mean Average Precision (mAP). The efficiency is measured by Average Number of Operation (ANO) per query image. Using inverted file index, ANO can be approximately computed as the product of Average Number of nonzero Visual words generated for each image (ANV) and Average Number of Images in each inverted list (ANI), *i.e.*, ANO=ANV×ANI. Therefore, a large mAP implies high discriminative power, and a small ANO implies high efficiency for indexing and retrieval.

The results are shown in Tab. 1. It is clear that, retrieval efficiency is significantly improved by discarding visual words with small response values without discarding the accuracy. Specially, retrieval accuracy is improved to 0.704 from 0.697 and ANO is reduced to 2,250 from around 1.7 million when β

is set to 0.11.

In the aforementioned procedure, the threshold is hard to decide for different testing sets. To determine the threshold β automatically, we design a sparsity loss function based on KLD as following:

$$\ell_{spa}(\beta) = \hat{\rho} \log \frac{\hat{\rho}}{\rho} + (1 - \hat{\rho}) \log \frac{(1 - \hat{\rho})}{1 - \rho}, \tag{2}$$

where $\hat{\rho}$ denotes the desired ratio between the number of nonzero visual words and the total number of visual words. ρ is the ratio computed on training set of N images, *i.e.*,

$$\rho = \frac{1}{N \times m \times n} \sum_{i=1}^{N} \sum_{j=i}^{m \times n} sign(v_i(j) - \beta). \tag{3}$$

$sign(\cdot)$ is sign function defined as follows:

$$sign(x) = \begin{cases} 1, & x > 0, \\ 0, & otherwise. \end{cases} \tag{4}$$

With this object function, the model is trained to learn the threshold β to ensure a ratio of $\hat{\rho}$ visual words are nonzero. We thus use this sparsity loss to control the sparsity of the generated visual words.

3.3. Model Training

Aiming to apply the proposed E^{2}BoWs model on large-scale image retrieval, the generated visual words should 1) preserve the similarity relationship among images for accuracy and 2) be sparse for efficiency. We also expect fast convergence when training the model. Thus we design the overall objective function as follows:

$$L(\theta) = \ell_{cls} + \lambda_1 \ell_{tri} + \lambda_2 \ell_{spa}. \tag{5}$$

θ denotes all parameters in proposed E^{2}BoWs model, including weights and bias in convolutional layers and threshold parameter β in sparsity loss. β denotes the threshold in BoWL. ℓ_{cla}, ℓ_{tri} and ℓ_{spa} denote the loss of classification, triplet similarity and sparsity, respectively. λ_1 and λ_2 denote loss weights for triplet loss and sparsity loss separately. Classification loss is introduced to ensure fast convergence, since only using the triplet loss takes a long time to converge. The triplet similarity loss ensures the discriminative ability of the learned features in similarity computation. And the sparsity loss ensures retrieval efficiency.

We design the triplet similarity loss as:

$$\ell_{tri}(v_a, v_p, v_n) = max\{0, sim_{v_a}^{v_n} - sim_{v_a}^{v_p} + \alpha\}, \tag{6}$$

where α is the margin parameter, v_a, v_p and v_n are the vectors of L2-normalized visual words of anchor image, similar image and dissimilar image, respectively. $sim_{v_1}^{v_2}$ is the cosine distance between two vectors computed as follows:

$$sim_{v_1}^{v_2} = v_1^T * v_2. \tag{7}$$

When $\ell_{tri}(v_a, v_p, v_n) \neq 0$, the gradient with respect to each vector can be computed as:

$$\frac{\partial \ell_{tri}(v_a, v_p, v_n)}{\partial v_a} = v_n - v_p, \tag{8}$$

$$\frac{\partial \ell_{tri}(v_a, v_p, v_n)}{\partial v_p} = -v_a, \tag{9}$$

$$\frac{\partial \ell_{tri}(v_a, v_p, v_n)}{\partial v_n} = v_a. \tag{10}$$

Different from other works that use Euclidean distance to compute the triplet similarity, we choose Cosine distance to make similar images share more visual words and vice versa. This is mainly because we also use Cosine distance to compute image similarity during retrieval based on inverted indexes.

The sparsity loss ℓ_{spa} is formulated in Eq. 2. Since the $sign(\cdot)$ function is non-differential, we define the gradient of it as follows:

$$\begin{aligned} \frac{\partial sign(v_i(j) - \beta)}{\partial \beta} &= -sign(v_i(j) - \beta) \\ &= \begin{cases} -1, & v_i(j) - \beta > 0, \\ 0, & otherwise. \end{cases} \end{aligned} \tag{11}$$

The gradient of $\ell_{spa}(\beta)$ can be computed as:

$$\begin{aligned} \frac{\partial \ell_{spa}(\beta)}{\partial \beta} &= \frac{\partial \ell_{spa}(\beta)}{\partial \rho} \cdot \frac{\partial \rho}{\partial \beta} \\ &= \frac{\hat{\rho} - \rho}{1 - \rho}. \end{aligned} \tag{12}$$

Therefore, β can be leaned by gradient descent method.

3.4. Generalization Ability Improvement

Most of conventional retrieval models based on DCNN need to be fine-tuned on the target dataset [9]. However, fine-tuning is commonly unavailable in real image retrieval applications. Then *ImageNet* [46] could be a reasonable option for training as it contains large-scale labeled images. However, *ImageNet* contains some fine-grained categories and some categories are both visually and semantically similar as shown in Fig. 4.

In our method, different categories correspond to different SFMs, which hence generate different visual words. It's not reasonable to regard similar categories to generate unrelated visual words, when using *ImageNet* as the training set. For example, images of "red fox" should be allowed to share more visual words with images of "kit fox" than with images of "jeep". Therefore, original labels in *ImageNet* [46] are not optimal for training E^{2}BoWs and may mislead the model for retrieval tasks.

To tackle the above issue, we change the parameter α in triplet loss function according to the similarity of two categories, *i.e.*, set a small value of α for images of similar categories and set a large value for images of dissimilar categories. Specifically, we first compute the similarity between two categories based on the tree struct[7] of *ImageNet* [46]. Given H denotes the height of the tree and $h_{c_1}^{c_2}$ denotes the height of the common parent nodes of two different categories c_1 and c_2, the similarity $S(c_1, c_2)$ between c_1 and c_2 is defined as:

$$S(c_1, c_2) = \frac{h}{H}. \tag{13}$$

[7]ImageNet Tree View. http://image-net.org/explore/

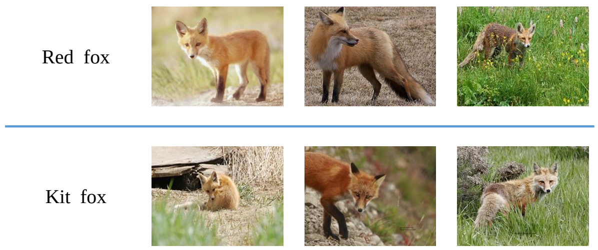


Figure 4: Illustration of two categories in *ImageNet* [46], that are visually and semantically similar.

Then we modify parameter α as:

$$\alpha' = \frac{\alpha}{(1 + S(c_1, c_2))^2}. \tag{14}$$

The above strategy allows images from similar categories to share more common visual words, thus makes $ImageNet$ a more reasonable training set. It is thus potential to improve the generalization ability of the learned E^{2}BoWs on other unseen datasets.

4. Experiments

4.1. Datasets

We first evaluate our model in tiny image retrieval task on *MNIST* [32], *SVHN* [34], *CIFAR-10* [33] and *CIFAR-100* [33]. Then, our model is evaluated in image retrieval task on *MIRFLICKR-25K* [35]. On these datasets, proposed method is compared with several state-of-the-arts including ITQ [50], ITQ-CCA [50], KSH [51], SH [52], MLH [53], BRE [54], CNNH [28], CNNH+ [28], DNNH [48], DSH [27], SUBIC [55], BDNN [56], DSRH [30], DRSCH [57] and BHC [29]. For each state-of-the-art method, we adopt its best reported performance for comparison. We compare mean Average Precision (mAP) on datasets such as *MNIST*, *SVHN*, *CIFAR-10* and *CIFAR-100*. We use Normalized Discounted Cumulative Gain @1000 (NDCG@100) [58] as the evaluation metric to consider different levels of relevance on *MIRFLICKR-25K*. Finally, we compare the generalization ability between the proposed E^{2}BoWs and deep features extracted from GoogLeNet [1] without/with BN [2] by first training the model on *ImageNet* [46][8] and then testing the model on *NUS-WIDE* [36] and mAP is used to evaluate the performance. Details of those test sets are given as follows:

- *MNIST* contains 70,000 28×28 images of handwritten digits from 0 to 9. It has 60,000 images for training and 10,000 images for testing.
- *SVHN* contains 99289 32×32 images of real-world digits from 0 to 9. It has 73,257 images for training and 26,032 images for testing. The digits is cropped from real-world images of street view house numbers. Many of the images contain distractors at the sides. Retrieval task on it is more challenging than the one on *MNIST*.
- *CIFAR-10* contains 60,000 32×32 images belonging to 10 classes, such as cat, ship and dog. Each class contains 5,000 training images and 1,000 testing images.
- *CIFAR-100* contains 60,000 32×32 images belonging to 100 classes, such as bridge, boy and maple. Each class contains 500 training images and 100 testing images. Retrieval task on it is more challenging than the one on *CIFAR-10*.
- *MIRFLICKR-25K* is a multi-label dataset and consists of 25,000 images labelled with 24 semantic concepts, such as dog, boy and bird. 14 of these concepts are used as stricter labels, resulting in 38 concepts in total.
- *NUS-WIDE* is also a multi-label dataset and consists of around 270K images labelled with 81 concepts.
- *ImageNet* contains roughly 1.2 million training images and 50,000 validation images. Images are labelled with 1,000 categories, such as kit fox, weasel and zebra.

[8]http://www.image-net.org/

Table 2: Model settings on each dataset.n denotes the number of SFM. m denotes the number of visual words generated from each SFM. $n \times m$ is the number of visual words generated per image. $\hat{\rho}$ denotes the parameter in sparsity loss function defined in Eqn. 2. And α denotes the margin parameter in similarity loss function defined in Eqn. 6.

	n	m	$n \times m$	$\hat{\rho}$	α
MNIST	10	3	30	0.08	0.2
SVHN	10	3	30	0.08	
CIFAR-10	10	10	100	0.08	
CIFAR-100	100	10	1,000	0.01	
MIRFLICKR-25K	38	10	380	0.11	
ImageNet	1000	25	25,000	0.14	

4.2. Implementation Details

In the propose E^{2}BoWs model, each SFM corresponds to a category on the training set. Therefore, the number of SFMs equals to the number of training categories.

We decide the number of visual words generated from each SFM on different datasets based on their intra-class semantic diversity. On *MNIST* and *SVHN*, 3 visual words are generated from each SFM as the two datasets are composed of digits images and thus contain less intra-class semantic diversity. So that 30 visual words are generated for each image on the two datasets. On *CIFAR-10*, *CIFAR-100* and *MIRFLICKR-25K*, 10 visual words are generated from each SFM, because these datasets exhibit more substantial intra-class semantic diversity. This results in 100, 1,000 and 380 visual words, respectively. For *ImageNet*, we generate 25 visual words on each SFM to enhance the generalization ability and get totally 25,000 visual words.

The sparsity loss parameter $\hat{\rho}$ is determined based on the number of categories of each dataset. Note that, each visual word is expected to be activated only for images containing its descriptive objects. For example, visual words descriptive to “cat” category are expected to be activated for images containing cat. Given n categories in dataset and $n \times m$ visual words are generated. Then only m visual words are expected to be activated for images in this dataset to ensure high feature sparsity. So that $\hat{\rho}$ is basically set as the reciprocal for the number of categories. For example, $\hat{\rho}$ is set to 0.01 on *CIFAR-100* to encourage 1% visual words to be active for images of each category. On *MNIST*, *SVHN* and *CIFAR-10*, $\hat{\rho}$ is set to 0.08 instead of 0.1 to enhance the sparsity of visual words. And $\hat{\rho}$ is set to 0.11 on *MIRFLICKR-25K* for the same reason. On the other hand, $\hat{\rho}$ is set to 0.14 instead of 0.001 on *ImageNet* to preserve more visual words to enhance the generalization ability. Margin parameter in similarity loss function is set to 0.2 on all datasets experientially. We summarize the model settings on each dataset in Tab. 2.

Our E^{2}BoWs model is implemented with Caffe [59]. The weights in convolutional layers are initialized from model pre-trained on ImageNet[9]. We fine-tune the model by SGD algorithm on a TITAN X 12GB GPU. The base learning rate is set to 0.001 and decreases by 70% after every 6 epochs (around 20 epochs in total on each dataset). The momentum is set to 0.9 and the weight decay is set to 0.0005.

In Tab. 3, 4, and 5, the tag "-B" denotes that visual words or feature is binarized by using $sign(\cdot)$ function defined in Eqn. 4 to accelerate the retrieval.

4.3. Performance on MNIST and SVHN

For *MNIST*, we following the settings in [29]: The training set is used to fine-tune the model. When retrieval, training set and testing set are used as gallery set and query set respectively. For *SVHN*, we following the settings in [48]: 1,000 images (100 images per category) are randomly selected as queries from testing set. And 5,000 images (500 images per category) are randomly selected to fine-tune the model, and also selected as gallery set for retrieval.

We compare the retrieval performance between proposed E^{2}BoWs and existing methods. The performance comparison is summarized in Tab. 3. As

[9]https://github.com/lim0606/caffe-googlenet-bn

BHC proposed by [29] reports the best performance among other methods and it doesn't report its performance on *SVHN*, we reimplement the method on SVHN based on GoogLeNet [1] with BN [2] in the same settings. CNNH [28] doesn't report its performance on *SVHN* either, so we show the performance reimplemented by [48]. In Tab. 3, "*" denotes our implementation and "#" denotes the implementation provided by [48]. It can be observed that DCNN based methods outperform all conventional methods on both datasets, and proposed E^{2}BoWs model shows the best performance compared with others. Conventional methods can achieve good performance on *MNIST* and there is a small margin between them and DCNN based methods. This might be because *MNIST* defines a relatively easy retrieval task. On the more challenging real-world *SVHN* dataset that contains many distractors, the performance gains of DCNN based methods become more substantial. It's obvious that DCNN based methods are more robust against distractors. It also can be observed that, though previous DCNN based methods yield quite good performance on *MNIST*, our proposed E^{2}BoWs model further improves the mAP to 0.996 from 0.985 achieved by BHC [29]. Morever, E^{2}BoWs model achieves the mAP of 0.987 after binarizing the visual words, which is still higher than 0.985 achieved by BHC [29]. This proves the better discriminative ability of proposed E^{2}BoWs model.

4.4. Performance on CIFAR

On *CIFAR-10* and *CIFAR-100*, we use the training sets for model fine-tuning. When retrieval on *CIFAR-10*, we follow the setting in [29]: All testing images are used as queries and the training set is used as gallery set. For *CIFAR-100*, we use the testing set as both gallery and query set for a fast evaluation. We also reimplement BHC [29] on *CIFAR-100*. The comparison of proposed E^{2}BoWs model and previous models is summarized in Tab. 3. Tab. 3 shows that, E^{2}BoWs model also shows the best performance on the two datasets. It can be observed from Tab. 3 that, methods based on DCNN perform better than conventional retrieval methods using hand-crafted features. Among DCNN

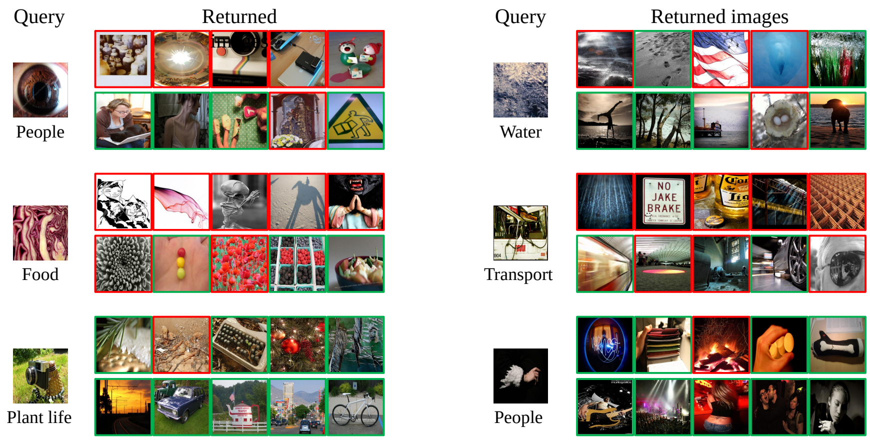


Figure 5: Examples of retrieval results of BHC [29] and proposed E^{2}BoWs-B on *MIRFLICKR-25K* [35]. In each example, the query image is placed on the left with the ground truth label under it. The first row shows the top 5 images returned by BHC [29], the second row shows the result of proposed E^{2}BoWs-B. Relevant/irrelevant images are annotated by green/red boxes, respectively.

based methods, our model yields the highest mAP on the two datasets. It is also clear that, our work also show substantial advantage on the more challenging *CIFAR-100* [33] dataset.

4.5. Performance on MIRFLICKR-25K

On *MIRFLICKR-25K* [35], we follow the experimental setting of [30]: 2,000 images are randomly selected as query images and the rest are used for training and gallery. We also implement BHC [29] for comparison because it shows the best performance among the compared works on *MNIST*, *SVHN*, *CIFAR-10* and *CIFAR-100*.

Performance comparison is shown in Tab. 4. It can be observed that, D-CNN based methods still perform better than the conventional methods. This implies the powerful feature learning ability of deep models. It is also clear that, binarized E^{2}BoWs achieves the best performance. The reason why E^{2}BoWs-B outperforms E^{2}BoWs might be because that, the relevance among images is directly measured by the number of their shared labels, rather than their similarity. Therefore, the binarized E^{2}BoWs-B might be more suited to such evaluation setting than E^{2}BoWs. Examples of image retrieval results of BHC [29] and E^{2}BoWs-B are shown in Fig. 5. As shown in Fig. 5, E^{2}BoWs-B is more discriminative to semantic cues. For example, E^{2}BoWs-B effectively identifies the semantic of "people" from an human eye image, and gets better retrieval results than BHC [29].

4.6. Evaluation on Generalization Ability

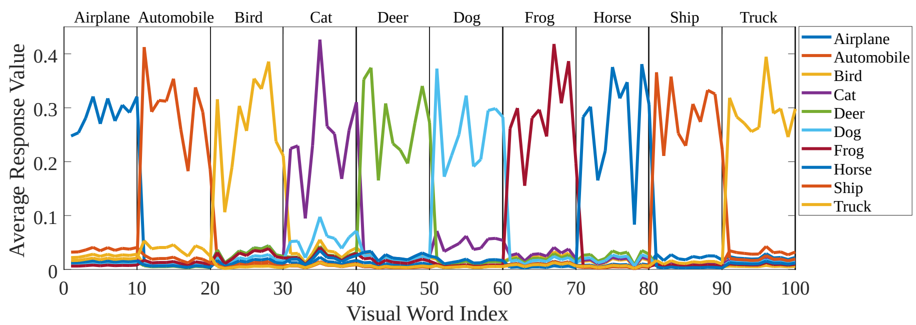
To validate the generalization ability of the proposed E^{2}BoWs feature, we first train E^{2}BoWs on *ImageNet* [46], then test it on *NUS-WIDE* [36]. The retrieval on *NUS-WIDE* [36] uses the same experimental setting in [27, 28], *i.e.*, use the images associated with the 21 most frequent concepts and the testing set in [27], which consists of 10,000 images. As one image may be associated with many concepts, we follow [27] and consider two images are similar if they share at least one concept. We compare our model with features generated directly from GoogLeNet [1] with and without BN [2], *i.e.*,

- $\mathrm{GN}_{1024}/\mathrm{GN}^{BN}_{1024}$: 1024-d feature extracted from the pool5 layer in GoogLeNet [1] without/with BN [2].
- $\mathrm{GN}_{1000}/\mathrm{GN}^{BN}_{1000}$: 1000-d feature extracted from the output layer in GoogLeNet [1] without/with BN [2].

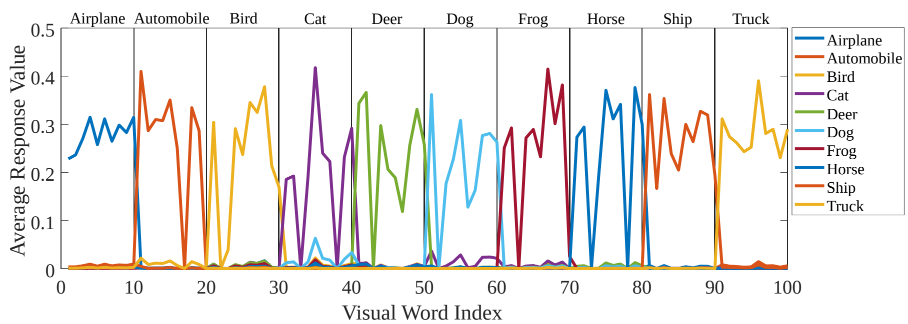
The comparison between E^{2}BoWs and GoogLeNet features is summarized in Tab. 5. It could be observed that our model constantly shows better retrieval accuracy. Note that, the above experiments use independent training and testing sets. Therefore, we can conclude that E^{2}BoWs shows better generalization ability than GoogLeNet features.

4.7. Test of the Semantic Cues Conveyed in Visual Words

We study the information conveyed in visual words on the test set of *CIFAR-10* [33] to prove that 1) proposed visual words convey clear semantic cues and 2) more detail in images can be preserved.



(a) Before Thresholding Layer



(b) After Thresholding Layer

Figure 6: Average response value of each visual words with respect to each category before thresholding layer in (a), and after thresholding layer in (b). All of the 100 visual words can be uniformly separated into 10 groups corresponding to 10 categories, as annotated on top of figure.

Our visual words are designed to show clear semantic cues, *e.g.*, certain visual words are activated only if an image contains corresponding objects. In Fig. 6, we show the average response value of each visual word to different categories before and after thresholding layer. It is clear that each visual word corresponds to a category and all of the 100 visual words can be separated into 10 groups by the semantics they convey. For example, the 1st to 10th visual words are activated when the image contains airplane. They thus present strong discriminative to the semantic of airplane. This shows that our generated visual words convey clear semantic cues. It is also clear from Fig. 6(b) that, the thresholding layer enhances the sparsity of visual words without changing such

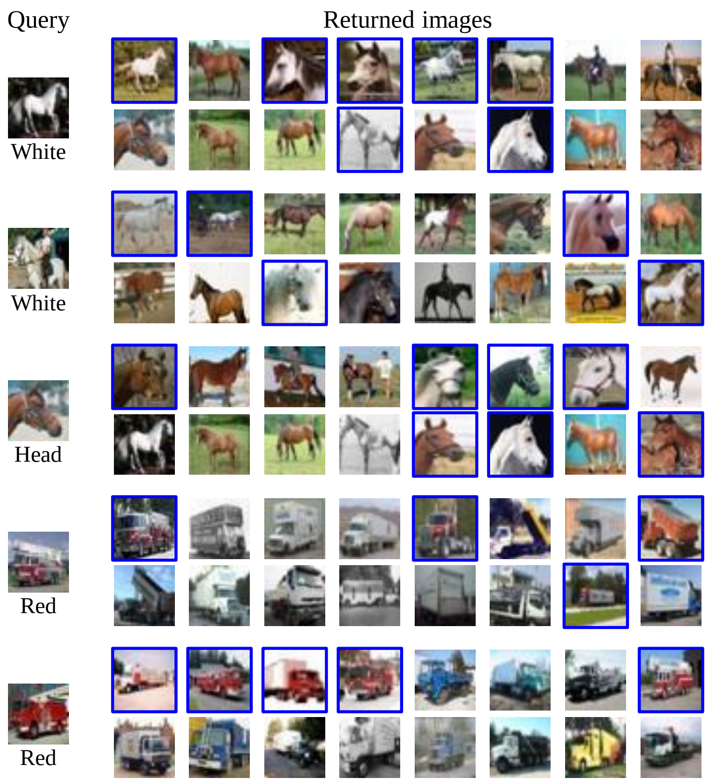


Figure 7: Examples of retrieval results of 100 visual words and 48-bit hash codes on *CIFAR-10*. In each example, the query image is placed on the left with its salient attribute under it. The first row shows top 8 images returned by 100 visual words and the second row shows the results of 48-bit hash codes. Returned images sharing same salient attribute are annotated by blue box.

property, therefore, makes the retrieval system more memory and time efficient.

We compare the returned images by 100 visual words and 48-bit hash codes proposed by [29] in Fig. 7. The 48-bit hash codes are reimplemented by us and achieve mAP of 0.907 on *CIFAR-10*, higher than 0.897 reported by [29]. It can be observed that our visual words show stronger discriminative power to semantics and generates more accurate image retrieval results than [29].

4.8. Discussions

During training, we encourage E^{2}BoWs to be sparse to ensure its high efficiency in inverted file indexing and retrieval. On *MNIST*, *SVHN*, *CIFAR-10*, *CIFAR-100*, and *MIRFLICKR-25K*, we analyze the retrieval complexity of our E^{2}BoWs model and compare it with the one of 48-bit binary code generated by BHC [29].

As shown in Tab. 6, E^{2}BoWs is sparse. For instance, the average number of visual words in each image on *MIRFLICKR-25K* is about 44, which is significantly smaller than the total visual word size 380. It is also clear that, with inverted file index, retrieval based on E^{2}BoWs can be efficiently finished with less operations than the linear search with binary code. From the above experiments, we can conclude that 1) E^{2}BoWs shows advantages in the aspects of both accuracy and efficiency compared with BHC [29], 2) the proposed visual words present stronger discriminative power to visual and semantic cues than the hash codes generated by [29].

5. Conclusions

This paper presents E^{2}BoWs for large-scale CBIR based on DCNN. E^{2}BoWs first transforms FC layer in GoogLeNet [1] into convolutional layer to generate semantic feature maps. Visual words are then generated from these feature maps by the proposed bag-of-words layer and to preserve both the semantic and visual cues. A threshold layer is hence introduced to ensure the sparsity of generated visual words, which further ensures the time and memory efficiency. We also introduce a novel learning algorithm to reinforce the fast convergence, semantically discriminative ability and sparsity of the generated E^{2}BoWs model. Experiments on six benchmark datasets demonstrate that our model shows substantial advantages in the aspects of discriminative power, efficiency, and generalization ability.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015.

[2] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[3] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: CVPR, 2006.

[4] S. Zhang, M. Yang, X. Wang, Y. Lin, Q. Tian, Semantic-aware co-indexing for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (12) (2015) 2573–2587.

[5] Z. Wu, Q. Ke, M. Isard, J. Sun, Bundling features for large scale partial-duplicate web image search, in: CVPR, 2009.

[6] J. Sivic, A. Zisserman, et al., Video google: A text retrieval approach to object matching in videos., in: ICCV, 2003.

[7] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, W. T. Freeman, Discovering objects and their location in images, in: ICCV, 2005.

[8] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[9] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep learning for content-based image retrieval: A comprehensive study, in: ACM MM, 2014.

[10] F. Perronnin, Universal and adapted vocabularies for generic visual categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1243–1256.

[11] S. Lazebnik, M. Raginsky, Supervised learning of quantizer codebooks by information loss minimization, IEEE transactions on Pattern Analysis and Machine Intelligence 31 (7) (2009) 1294–1309.

[12] L. Wu, S. C. Hoi, N. Yu, Semantics-preserving bag-of-words models and applications, IEEE Transactions on Image Processing 19 (7) (2010) 1908–1920.

[13] L. Wu, S. C. Hoi, Enhancing bag-of-words models with semantics-preserving metric learning, IEEE MultiMedia 18 (1) (2011) 24–37.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[16] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: CVPR, 2014.

[17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: NIPS, 2015.

[18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015.

[19] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: CVPR, 2015.

[20] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, in: CVPR, 2014.

[21] S. Bell, K. Bala, Learning visual similarity for product design with convolutional neural networks, ACM Transactions on Graphics 34 (4) (2015) 98.

[22] L. Shen, Z. Lin, Q. Huang, Learning deep convolutional neural networks for places2 scene recognition, CoRR, vol. abs/1512.05830.

[23] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: NIPS, 2014.

[24] W. Longhui, Z. Shiliang, Y. Hantao, G. Wen, T. Qi, Glad: Global-local-alignment descriptor for pedestrian retrieval, in: ACM MM, 2017.

[25] S. Chi, L. Jianing, Z. Shiliang, X. Junliang, G. Wen, T. Qi, Pose-driven deep convolutional model for person re-identification, in: ICCV, 2017.

[26] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014.

[27] H. Liu, R. Wang, S. Shan, C. X., Deep supervised hashing for fast image retrieval, in: CVPR, 2016.

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

[29] K. Lin, H. Yang, J. Hsiao, C. Chen, Deep learning of binary hash codes for fast image retrieval, in: CVPRW, 2015.

[30] F. Zhao, Y. Huang, L. Wang, T. Tan, Deep semantic ranking based hashing for multi-label image retrieval, in: CVPR, 2015.

[31] H. Liu, R. Wang, S. Shan, C. X., Learning multifunctional binary codes for both category and attribute oriented retrieval tasks, in: CVPR, 2017.

[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[33] K. Alex, Learning multiple layers of features from tiny images, Tech. rep., Department of Computer Science, University of Toronto (2009).

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS workshop on deep learning and unsupervised feature learning, 2011.

[35] M. J. Huiskes, M. S. Lew, The mir flickr retrieval evaluation, in: MIR, 2008.

[36] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: A real-world web image database from national university of singapore, in: CIVR, 2009.

[37] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[38] Y. Jing, S. Baluja, Visualrank: Applying pagerank to large-scale image search, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (11) (2008) 1877–1890.

[39] S. Battiato, G. Farinella, G. Gallo, D. Ravì, Spatial hierarchy of textons distributions for scene classification, Advances in Multimedia Modeling (2009) 333–343.

[40] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: ECCV, 2006.

[41] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image and Vision Computing 22 (10) (2004) 761–767.

[42] D. Liu, G. Hua, P. Viola, T. Chen, Integrated feature selection and higher-order spatial feature extraction for object categorization, in: CVPR, 2008.

[43] L. Chu, S. Jiang, S. Wang, Y. Zhang, Q. Huang, Robust spatial consistency graph model for partial duplicate image retrieval, IEEE Transactions on Multimedia 15 (8) (2013) 1982–1996.

[44] S. Wang, S. Jiang, Instre: a new benchmark for instance-level object retrieval and recognition, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 11 (3) (2015) 37.

[45] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: ECCV, 2014.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009.

[47] H. Zhu, M. Long, J. Wang, Y. Cao, Deep hashing network for efficient similarity retrieval., in: AAAI, 2016.

[48] H. Lai, Y. Pan, Y. Liu, S. Yan, Simultaneous feature learning and hash coding with deep neural networks, in: CVPR, 2015.

[49] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: ICML, 2010.

[50] Y. Gong, S. Lazebnik, Iterative quantization: A procrustean approach to learning binary codes, in: CVPR, 2011.

[51] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: CVPR, 2012.

[52] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: NIPS, 2009.

[53] M. Norouzi, D. M. Blei, Minimal loss hashing for compact binary codes, in: ICML, 2011.

[54] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: NIPS, 2009.

[55] H. Jain, J. Zepeda, P. Pérez, R. Gribonval, Subic: A supervised, structured binary code for image search, in: ICCV, 2017.

[56] T.-T. Do, A.-D. Doan, N.-M. Cheung, Learning to hash with binary deep neural network, in: ECCV, 2016.

[57] R. Zhang, L. Lin, R. Zhang, W. Zuo, L. Zhang, Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification, IEEE Transactions on Image Processing 24 (12) (2015) 4766–4779.

[58] K. Järvelin, J. Kekäläinen, Ir evaluation methods for retrieving highly relevant documents, in: ACM SIGIR, 2000.

[59] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093.

Table 3: Comparison of mAP (%) among different methods on *MNIST*, *CIFAR-10*, *CIFAR-100* and *SVHN*. "*" denotes our implementation. "#" denotes the results is reported by [48].

Method	MNIST	SVHN	CIFAR-10	CIFAR-100
ITQ [50]	0.429	0.139	0.175	—
ITQ-CCA [50]	0.726	0.509	0.295	—
KSH [51]	0.900	0.581	0.315	—
SH [52]	0.250	0.140	0.132	—
MLH [53]	0.654	0.273	0.211	—
BRE [54]	0.634	0.237	0.196	—
CNNH [28]	0.960	$0.896^{\#}$	0.522	—
CNNH+ [28]	0.975	—	0.532	—
DNNH [48]	—	0.923	0.581	—
DHN [47]	—	—	0.621	—
DSH [27]	—	—	0.676	—
SUBIC [55]	—	—	0.686	—
BDNN [56]	0.955	—	0.696	—
DSRH [30]	—	—	0.618	—
DRSCH [57]	0.981	—	0.633	—
BHC [29]	0.985	0.941^{*}	0.897	0.650^{*}
E^2BoWs	**0.996**	**0.942**	**0.926**	**0.689**
E^2BoWs-B	0.987	0.925	0.923	0.624

Table 4: Comparison of NDCG@100 among different methods on *MIRFLICKR-25K* [35].

ITQ-CCA [50]	KSH [51]	BHC [29]	E^2BoWs	E^2BoWs-b
0.402	0.350	0.510^{*}	0.492	**0.526**

Table 5: Comparison of mAP (%) between GoogLeNet feature and E^{2}BoWs on *NUS-WIDE* [36]. The compared features are trained on an independent training set.

Feature	GN_{1024}	GN_{1000}	GN_{1024}^{BN}	GN_{1000}^{BN}	E^{2}BoWs
mAP	0.552	0.594	0.551	0.591	**0.599**

Feature	GN_{1024}-B	GN_{1000}-B	GN_{1024}^{BN}-B	GN_{1000}^{BN}-B	E^{2}BoWs-B
mAP	0.388	0.549	0.326	0.543	**0.563**

Table 6: Retrieval efficiency of different methods on *MNIST*, *SVHN*, *CIFAR-10*, *CIFAR-100* and *MIRFLICKR-25K*.

	BHC [29]	E^{2}BoWs		
	ANO	ANO	ANV	ANI
MNIST	2,880,000	4,560	1.51	3020
SVHN	240,000	1,095	2.54	431
CIFAR-10	2,400,000	37,688	8.65	4357
CIFAR-100	480,000	1,124	10.6	106
MIRFLICKR-25K	1,104,000	42,510	43.6	975