E^{2}BoWs: An End-to-End Bag-of-Words Model via Deep Convolutional Neural Network

Xiaobin Liu*, Shiliang Zhang*, Tiejun Huang*, Qi Tian†
*School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
Email: {xbliu.vmc, slzhang.jdl, tjhuang}@pku.edu.cn
†Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-1604, USA
Email: qitian@cs.utsa.edu

Abstract—**Traditional Bag-of-visual 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^{2}BoWs) model based on Deep Convolutional Neural Network (DCNN). Our model takes an image as input, then identifies and separates the semantic objects in it, and finally outputs the 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 for each individual feature map. We also introduce a novel learning algorithm to reinforce the sparsity of the generated E^{2}BoWs model, which further ensures the time and memory efficiency. We evaluate the proposed E^{2}BoWs model on several image search datasets including *CIFAR-10*, *CIFAR-100*, *MIRFLICKR-25K* and *NUS-WIDE*. Experimental results show that our method achieves promising accuracy and efficiency compared with recent deep learning based retrieval works.**

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-visual Words (BoWs) model, which considers an image as a collection of visual words, has been widely applied for large-scale image retrieval. Conventional BoWs model is computed with many stages, *e.g.*, feature extraction, codebook generation, and feature quantization [3–6]. 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 very large, *e.g.*, 1 million in [3], and a certain 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 [7]. These models have shown promising performance in large-scale partial-duplicate image retrieval [3–5]. However, as the local descriptor cannot effectively describe high-level semantics, *i.e.*, commonly known as the "semantic gap" issue, BoWs models build on local descriptors always fail to address the semantic similar image retrieval task [8]. Although some works have been proposed to conquer this issue [9–11], 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 [12] 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, 13], object detection [14, 15], semantic segmentation [16] and many other tasks [17–20]. Most of DCNNs consist of a set of convolutional layers and Fully Connected (FC) layers. It is found that convolutional layers can extract high-level semantic cues from pixel-level input and hence provide a possible solution to solve the "semantic ga" issue. Therefore, it is straightforward to leverage DCNN in image retrieval [8]. Some works use DCNN to generate hash codes and yield promising performance [21–24]. However, there still lacks 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 image retrieval, we propose to generate a DCNN based End-to-End BoWs (E^{2}BoWs) model as shown in Fig. 1. Structure of our E^{2}BoWs 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) specifically corresponding to different object categories. A Bag-of-Words Layer (BoWL) is then introduced to generate sparse visual words from each semantic feature map. This ensures the resulting visual words to preserve clear semantic

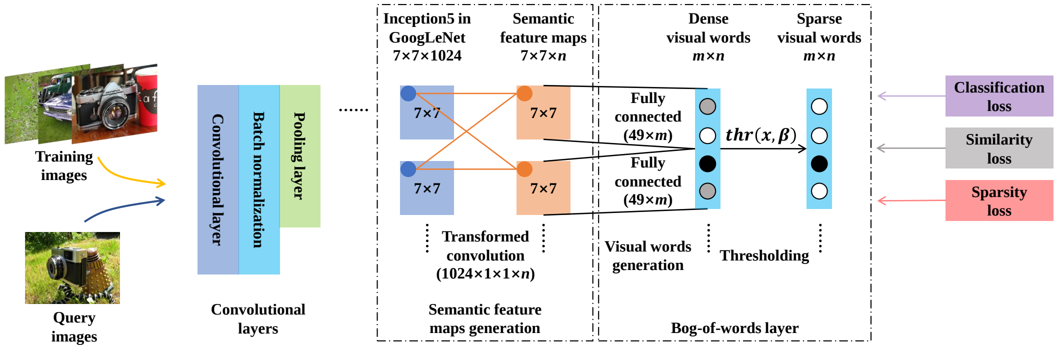

Figure 1: Framework of the proposed E^{2}BoWs model. The structure of our deep model is identical to the one of GoogLeNet [1] with BN [2] till the Inception5 layer. The output size of Inception5 is $7 \times 7 \times 1024$. Pool5 in GoogLeNet [1] is discarded. The n-way output layer is transformed into a convolutional layer to generate n semantic feature maps. m sparse visual words are then generated by bog-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.

cues. Finally, a 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.

The proposed method has several advantages compared with traditional BoWs models: 1) Instead of using hand-crafted features and being generated with several steps, our E^{2}BoWs model is generated in an end-to-end manner, thus is more efficient and easier to be jointly optimized and tuned. 2) Incorporating DCNN into BoWs model is potential to bring higher discriminative power to semantics and provide a better solution for semantic similar image search task. Our E^{2}BoWs model also shows advantages over traditional hashing methods in that it conveys clear semantic cues. We evaluate the proposed E^{2}BoWs model on several image search datasets including *CIFAR-10*, *CIFAR-100*, *MIRFLICKR-25K*, and *NUS-WIDE*. 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 our 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 [8, 25, 26], 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. Over the past decade, various BoWs models [3–6] have been proposed based on local descriptors, such as SIFT [7] and SURF [27]. 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 is generated by many stages, *e.g.*, feature extraction, codebook generation and feature quantization [3–6]. For instnace, Nister *et al.* [3] extract SIFT [7] descriptors from MSER regions [28] and then hierarchically quantize SIFT descriptors by the vocabulary tree. As individual visual word cannot depict the spatial cues in images, some works combine visual words with spatial information [29, 30] to make the resulting BoWs model more discriminative to the spatial cues. Some other works aim to generate more effective and discriminative vocabularies [31, 32].

However, the dependency on hand-crafted local feature hinders the ability of conventional visual words to convey semantic cues due to the “semantic gap” between low-level local features 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.

Some works have been proposed to enhance the discriminative power of BoWs model to semantic cues [9–11]. Wu *et al.* [9] propose an off-line distance metric learning scheme to map related features to the same visual words to generate an optimized codebook. Wu *et al.* [10] present an on-line metric learning algorithm to improve the BoWs model by optimizing the proposed semantic loss. Zhang *et al.* [11] propose a method to co-index semantic attributes into inverted index generated by local features to make it 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 many independent steps, they are hard to be jointly optimized to achieve better efficiency and accuracy.

Recently, many works leverage DCNN in CBIR [8, 21–24, 33, 34]. Wan *et al.* [8] propose three schemes to apply DCNN in CBIR, *i.e.*, 1) directly use the features from the model pre-trained on *ImageNet* [35], 2) refine the features by metric learning, and 3) retrain the model on the domain dataset. They prove that DCNN based features can significantly outperform hand-crafted features after being fine-tuned. However, they don’t consider the retrieval efficiency when apply the features in large-scale datasets. Xia *et al.* [22] introduce a DCNN based hashing method. The method consists of two steps: first generate hash codes on training set by an iterative algorithm, and then learn a hash function based on DCNN to fit the hash codes generated in step 1. The independence of two steps hinders the joint learning of the whole model. Lin *et al.* [23] propose a framework to generate hash codes directly by a classification object function. They show that deep model trained by classification task can be adopted for CBIR task. Zhao *et al.* [24] and Lai *et al.* [33] use triplet loss to train the network to preserve semantic relations of images.

In these aforementioned methods, real-value hash codes are learned during training. The real-value hash codes are then quantized to binary codes for testing. Different distance metrics used in training and testing, *e.g.*, Euclidean distance and Hamming distance, may bring approximation error and hinder the training efficiency. Quantization error could also be produced by the quantization stage. Different from those works, Liu *et al.* [21] and Zhu *et al.* [34] reinforce the networks to output binary-like hash codes to reduce quantization error and approximation error. So far, most of deep learning based retrieval works focus on generating hashing codes. There still lacks research efforts in DCNN based BoWs model. It is promising to generate a discriminative BoWs model directly from an end-to-end DCNN and leverage the scalability of BoWs model for large-scale image retrieval.

3. Proposed Method

E^{2}BoWs model is generated by modifying the GoogLeNet [1] with BN [2]. As shown in Fig. 1, before the Inception5 layer, the structure of our deep model is identical to the one of GoogLeNet [1] with BN [2]. 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. Then, m sparse visual words are generated from each individual SFM by the Bag-of-Words Layer, resulting in $m \times n$ visual words. 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. Meantime, 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.

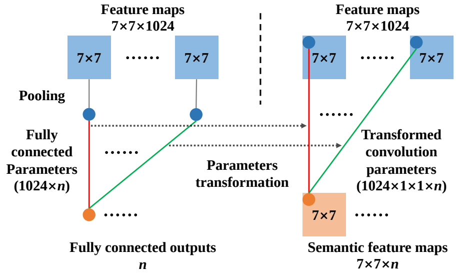

Figure 2: Illustration of transforming parameters of FC layer into a convolutional layer to generate SFMs. Lines in same color indicate the same parameters.

SFMs is generated by transforming the parameters in FC layers into a convolutional layer. This transformation is illustrated in Fig. 2. The size of parameters 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* [35] 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 presented by SFMs.

3.2. Bag-of-Words Layer

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 conveys both semantic and visual cues. To preserve the spacial and semantic cues in SFMs, we introduce Bag-of-Words Layer (BoWL) to generate sparse visual words directly from each individual SFM.

Specifically, a local FC layer with ReLU 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

TABLE 1: Retrieval efficiency and accuracy on *CIFAR-100* [36] testing set with different thresholds.

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

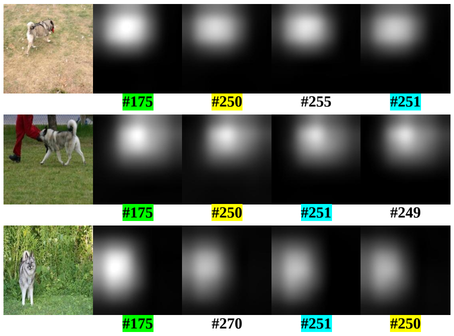

Figure 3: Visualization of some SFMs. Input images are in the first column. The rest are SFMs with top-4 largest response values. The number under each SFM denotes its unique ID in all SFMs. The same IDs are highlighted with the same color.

cues in each SFM, and 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. It should be noted that, we discard SFMs with negative average active values during visual words generation. This reduces the number of nonzero visual words and improves the efficiency for indexing and retrieval.

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.*, $1e$-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 overhead. We find that discarding visual words, whose response values are smaller than a threshold, dramatically improves the retrieval efficiency without degrading the accuracy. This procedure is formulated as follows:

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

where β denotes the threshold.

We evaluate this procedure on the testing set of *CIFAR-100* [36] 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 per image (ANV) and Average Number of Images in each inverted list (ANI), *i.e.*, ANO=ANV×ANI. Therefore, large mAP implies high discriminative power, and 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 filtering visual words with small response values. Meanwhile, retrieval accuracy is improved by removing noisy visual words.

In the aforementioned procedure, the threshold is hard to decide for different testing sets. To determine the threshold 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

The overall network is trained by SGD with object function as following,

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

where θ denotes parameters in convolutional layers, β denotes the threshold in BoWL, ℓ_{cla}, ℓ_{tri} and ℓ_{spa} denote the loss of classification, triplet similarity and sparsity, respectively. Since only using the triplet loss takes a long time to converge, we further introduce the classification loss to ensure fast convergence. The triplet similarity loss ensures the discriminative ability of the learned features in similarity computation. 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, *i.e.*, $sim_{v_1}^{v_2} = v_1^T * v_2$.

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

$$\begin{aligned} \frac{\partial \ell_{tri}(v_a, v_p, v_n)}{v_a} &= v_n - v_p && (7) \\ \frac{\partial \ell_{tri}(v_a, v_p, v_n)}{v_p} &= -v_a && (8) \\ \frac{\partial \ell_{tri}(v_a, v_p, v_n)}{v_n} &= v_a && (9) \end{aligned}$$

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

$$\begin{aligned} \frac{\partial sign(v_i(j) - \beta)}{\partial \beta} &= -sign(v_i(j) - \beta) \\ &= \begin{cases} -1, & v_i(j) - \beta > 0 \\ 0, & otherwise \end{cases} && (10) \end{aligned}$$

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

$$\begin{aligned} \frac{\partial \ell_{spa}(\beta)}{\partial \beta} &= \frac{\partial \ell_{spa}(\beta)}{\partial \rho} \cdot \frac{\partial \rho}{\partial \beta} \\ &= \frac{\hat{\rho} - \rho}{1 - \rho} && (11) \end{aligned}$$

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 domain dataset [8]. However, fine-tuning is commonly unavailable in real image retrieval applications. Then *ImageNet* [35] 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* [35] 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 use a large value for images of dissimilar categories. Specifically, we first compute the similarity between two categories based on the tree struct[1] of *ImageNet* [35]. 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: $S(c_1, c_2) = \frac{h}{H}$. Then we modify parameter α as:

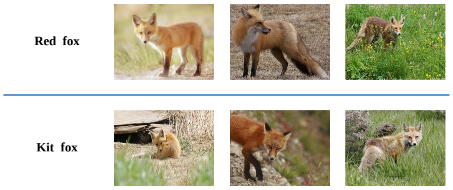

Figure 4: Illustration of two categories in *ImageNet* [35], that are visually and semantically similar.

$$\alpha' = \frac{\alpha}{(1 + S(c_1, c_2))^2} \tag{12}$$

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 and Implementation Details

We first evaluate our model in tiny image retrieval task on *CIFAR-10* [36] and *CIFAR-100* [36]. Then, our model is evaluated in image retrieval task on *MIRFLICKR-25K* [37]. Finally, we compare the generalization ability between the proposed E^{2}BoWs and deep features extracted from GoogLeNet [1] without/with BN [2] by firt training the model on *ImageNet* [35] and then testing the model on *NUS-WIDE* [38]. Details of those test sets are given as follows:

- *CIFAR-10* [36] contains tiny images belonging to 10 classes. Each class contains 5,000 training images and 1,000 testing images.
- *CIFAR-100* [36] contains 100 classes of tiny images. Each class contains 500 training images and 100 testing images. Retrieval task on it is more challenging than the one on *CIFAR-10* [36].
- *MIRFLICKR-25K*[37] consists of 25,000 images with 38 concepts.
- *ImageNet* [35] contains 1,000 categories and around 1,200 images per category.
- *NUS-WIDE* [38] consists of around 270K images and 81 concepts.

Each SFM corresponds to a category on the training set. Therefore, the number of SFMs equals to the number of training categories. On *CIFAR-10*, *CIFAR-100*, and *MIRFLICKR-25K*, 10 visual words are generated from each

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

TABLE 2: Comparison of mAP (%) among different methods on *CIFAR-10* [36] and *CIFAR-100* [36].

Method	CIFAR-10	CIFAR-100
ITQ [39]	0.175	—
ITQ-CCA [39]	0.295	—
KSH [40]	0.315	—
SH [41]	0.132	—
MLH [42]	0.211	—
BRE [43]	0.196	—
CNNH [22]	0.522	—
CNNH+ [22]	0.532	—
DNNH [44]	0.581	—
DSH [21]	0.676	—
BHC [23]	0.897	0.650*
E^{2}BoWs	**0.909**	**0.689**
E^{2}BoWs-B	0.908	0.624

SFM. This results in 100, 1,000 and 380 visual words, respectively. For *ImageNet* [35], we generate 25 visual words on each SFM and get totally 25,000 visual words. Margin parameter in similarity loss is set to 0.2 on all datasets.

mAP (mean Average Precision) is used to evaluate the retrieval performance on *CIFAR-10*, *CIFAR-100*, and *NUS-WIDE*. For *MIRFLICKR-25K* [37], we use NDCG@100 as the evaluation metric to consider different levels of relevance. In Tab. 2, 3, and 4, the tag “-B” denotes that feature is binarized by using $sign(\cdot)$ function to accelerate the retrieval.

4.2. Performance on CIFAR

On *CIFAR-10* and *CIFAR-100*, we use the training sets for model fine-tuning and use the test sets for retrieval, respectively. The sparsity loss parameter $\hat{\rho}$ is set as 0.08 and 0.01 on *CIFAR-10* and *CIFAR-100*, respectively depending on the number of categories. We compare the retrieval performance between E^{2}BoWs and existing methods including ITQ [39], ITQ-CCA [39], KSH [40], SH [41], MLH [42], BRE [43], CNNH [22], CNNH+ [22], DNNH [44], DSH [21], and BHC [23].

The performance comparison is summarized in Tab. 2, which shows the best performance of each method with 48-bit codes. The compared methods do not report their performance on the *CIFAR-100*. Among those methods, BHC [23] shows the best performance on *CIFAR-10*. Therefore, we implement BHC [23] and report its performance on *CIFAR-100* for comparison. In Tab. 2, “*” denotes our implementation. It can be observed from Tab. 2 that, methods based on DCNN perform better than conventional retrieval methods using hand-crafted features. Among DCNN 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* [36] dataset.

4.3. Performance on MIRFLICKR-25K

On *MIRFLICKR-25K* [37], we follow the experimental setting of [24], where 2,000 images are randomly selected as query images and the rest are used for training. We set sparsity loss parameter $\hat{\rho}$ to 0.11. We also implement BHC [23] for comparison because it shows the best performance among the compared works on *CIFAR-10* [36].

TABLE 3: Comparison of NDCG@100 among different methods on *MIRFLICKR-25K* [37].

ITQ-CCA [39]	KSH [40]	BHC [23]	E^{2}BoWs	E^{2}BoWs-b
0.402	0.350	0.510*	0.492	**0.526**

Performance comparison is shown in Tab. 3. It can be observed that, DCNN based methods also 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. Examples of image retrieval results of BHC [23] 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 effectively identifies the semantic of “people” from an human eye image, and gets better retrieval results than BHC [23].

4.4. Evaluation on Generalization Ability

To validate the generalization ability of the proposed E^{2}BoWs feature, we first train E^{2}BoWs on *ImageNet* [35], then test it on *NUS-WIDE* [38]. When training on *ImageNet* [35], the sparsity loss parameter is relaxed to 0.14 and 25 visual words are generated from each SFM. The retrieval on *NUS-WIDE* [38] uses the same experimental setting in [21, 22], *i.e.*, use the images associated with the 21 most frequent concepts and the testing set in [21], which consists of 10,000 images. As one image may be associated with many concepts, we follow [21] 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.*,

- GN_{1024}/GN^{BN}_{1024}: 1024-d feature extracted from the pool5 layer in GoogLeNet [1] without/with BN [2].
- GN_{1000}/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. 4. 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.

4.5. Discussions

During training, we encourage E^{2}BoWs to be sparse to ensure its high efficiency in inverted file indexing and retrieval. On *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 [23].

As shown in Tab. 5, 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,

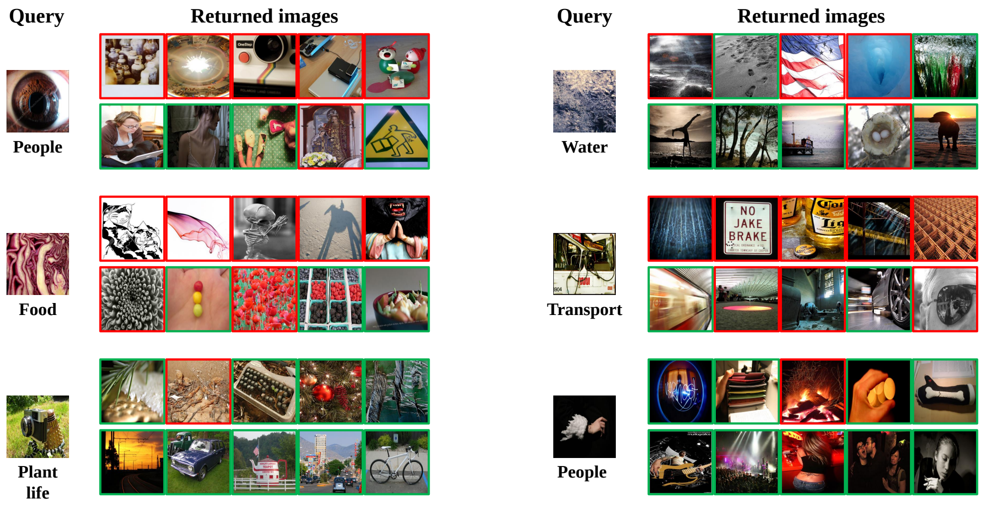

Figure 5: Examples of retrieval results of BHC [23] and proposed E^{2}BoWs-B on *MIRFLICKR-25K* [37]. In each example, the query image is placed on the left with the ground truth label under it. The first row shows the top 5 images returned by BHC [23], the second row shows the result of E^{2}BoWs-B. Relevant/irrelevant images are annotated by green/red boxes, respectively.

TABLE 4: Comparison of mAP (%) between GoogLeNet feature and E^{2}BoWs on *NUS-WIDE* [38]. The compared features are trained on an independent training set.

Feature	GN_{1024}	GN_{1000}	GN^{BN}_{1024}	GN^{BN}_{1000}	E^{2}BoWs	GN_{1024}-B	GN_{1000}-B	GN^{BN}_{1024}-B	GN^{BN}_{1000}-B	E^{2}BoWs-B
mAP	0.552	0.594	0.551	0.591	**0.599**	0.388	0.549	0.326	0.543	**0.563**

TABLE 5: Retrieval efficiency of different methods on *CIFAR-10* [36], *CIFAR-100* [36] and *MIRFLICKR-25K* [37].

Method		CIFAR-10	CIFAR-100	*MIRFLICKR-25K*
BHC [23]	ANO	480, 000	480, 000	406, 944
E^{2}BoWs	ANV	8.64	10.6	43.6
	ANI	960	110	975
	ANO	8, 294	1, 166	42, 510

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 E^{2}BoWs shows advantages in the aspects of both accuracy and efficiency, compared with BHC [23].

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 to preserve both the semantic and visual cues. A threshold layer is hence introduced to ensure the sparsity of generated visual words. We also introduce a novel learning algorithm to reinforce the sparsity of the generated E^{2}BoWs model, which further ensures the time and memory efficiency. Experiments on four benchmark datasets demonstrate that our model shows substantial advantages in the aspects of discriminative power, efficiency, and generalization ability.

Acknowledgments

This work is supported by National Science Foundation of China under Grant No. 61572050, 91538111, 61620106009, 61429201, and the National 1000 Youth Talents Plan, in part to Dr. Qi Tian by ARO grant W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

References

[1] 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.

[2] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[3] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *CVPR*, 2006.

[4] Z. Wu, Q. Ke, M. Isard, and J. Sun, “Bundling features for large scale partial-duplicate web image search,” in *CVPR*, 2009.

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

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

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] 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.

[9] L. Wu, S. C. Hoi, and N. Yu, “Semantics-preserving bag-of-words models and applications,” *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1908–1920, 2010.

[10] L. Wu and S. C. Hoi, “Enhancing bag-of-words models with semantics-preserving metric learning,” *IEEE MultiMedia*, vol. 18, no. 1, pp. 24–37, 2011.

[11] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, “Semantic-aware co-indexing for image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2573–2587, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

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

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

[16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.

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

[18] 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.

[19] L. Shen, Z. Lin, and Q. Huang, “Learning deep convolutional neural networks for places2 scene recognition,” *CoRR, vol. abs/1512.05830*, 2015.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *NIPS*, 2014.

[21] H. Liu, R. Wang, S. Shan, and C. X., “Deep supervised hashing for fast image retrieval,” in *CVPR*, 2016.

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

[23] K. Lin, H. Yang, J. Hsiao, and C. Chen, “Deep learning of binary hash codes for fast image retrieval,” in *CVPRW*, 2015.

[24] F. Zhao, Y. Huang, L. Wang, and T. Tan, “Deep semantic ranking based hashing for multi-label image retrieval,” in *CVPR*, 2015.

[25] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[26] Y. Jing and S. Baluja, “Visualrank: Applying pagerank to large-scale image search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1877–1890, 2008.

[27] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *ECCV*, 2006.

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[29] S. Battiato, G. Farinella, G. Gallo, and D. Ravì, “Spatial hierarchy of textons distributions for scene classification,” *Advances in Multimedia Modeling*, pp. 333–343, 2009.

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

[31] S. Lazebnik and M. Raginsky, “Supervised learning of quantizer codebooks by information loss minimization,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, 2009.

[32] F. Perronnin, “Universal and adapted vocabularies for generic visual categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.

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

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

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

[36] K. Alex, “Learning multiple layers of features from tiny images,” Department of Computer Science, University of Toronto, Tech. Rep., 2009.

[37] M. J. Huiskes and M. S. Lew, “The mir flickr retrieval evaluation,” in *MIR*, 2008.

[38] T. 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 *CIVR*, 2009.

[39] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *CVPR*, 2011.

[40] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, “Supervised hashing with kernels,” in *CVPR*, 2012.

[41] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *NIPS*, 2009.

[42] M. Norouzi and D. M. Blei, “Minimal loss hashing for compact binary codes,” in *ICML*, 2011.

[43] B. Kulis and T. Darrell, “Learning to hash with binary reconstructive embeddings,” in *NIPS*, 2009.

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