Local Deep Descriptors in Bag-of-Words for Image Retrieval

Jiewei Cao
The University of Queensland
Australia
j.cao3@uq.edu.au

Zi Huang
The University of Queensland
Australia
huang@itee.uq.edu.au

Heng Tao Shen
University of Electronic Science and Technology of China
China
shenhengtao@hotmail.com

ABSTRACT

The Bag-of-Words (BoW) models using the SIFT descriptors have achieved great success in content-based image retrieval over the past decade. Recent studies show that the neuron activations of the convolutional neural networks (CNN) can be viewed as local descriptors, which can be aggregated into effective global descriptors for image retrieval. However, little work has been done on using these local deep descriptors in BoW models, especially in the case of large visual vocabularies.

In this paper, we provide the key ingredients to build an effective BoW model using deep descriptors. Specifically, we show how to use the CNN as a combination of local feature detector and extractor, without the need of feeding multiple image patches to the network. Moreover, we revisit the classic issues of BoW – including the burstiness and quantization error – in our scenario and improve the retrieval accuracy by addressing these problems. Lastly, we demonstrate that our model can scale up to large visual vocabularies, enjoying the advantages of both the sparseness of visual word histogram and the discriminative power of deep descriptor. Experiments show that our model achieves state-of-the-art performance on different datasets without re-ranking.

CCS CONCEPTS

• **Information systems** → **Image search**; *Multimedia and multimodal retrieval*;

ACM Reference format:
Jiewei Cao, Zi Huang, and Heng Tao Shen. 2017. Local Deep Descriptors in Bag-of-Words for Image Retrieval. In *Proceedings of ThematicWorkshops'17, October 16–20, 2017, Snowbird, UT, USA., ,* 7 pages.
DOI: https://doi.org/10.1145/3126686.3127018

1 INTRODUCTION

Content-based image retrieval (CBIR) has been an active research topic since the seminal work of Sivic and Zisserman [38], which is based on the Bag-of-Words (BoW) model using local descriptor SIFT [22]. The general workflow of BoW is that first a visual vocabulary (or codebook) is learned on a set of local descriptors, and each image is quantized to a visual word histogram. The classic weighting schemes such as TF-IDF and the inverted index are incorporated for retrieval. The naive model is further enriched by using better feature detectors and descriptors [23, 27, 36], large visual vocabularies [24, 26, 29], spatial verification [29, 35] and query expansion [10].

Recently, the successes of the convolutional neural network (CNN) [19] in large scale image classification [34] and many other computer vision problems attract a lot of attention. Several works [1, 5, 11, 18, 31, 33, 40] show that the convolutional feature maps (CFMs) extracted from the CNN can be viewed as a set of local descriptors, which are able to be aggregated into powerful global features for image retrieval. Compared to the conventional descriptors, these local deep descriptors are learned by training a CNN on large labeled image datasets [34] and/or fine-tuning a pre-trained CNN on task-specific datasets. The state-of-the-art performances achieved by these works suggest that the local deep descriptors are much more discriminative.

Different from compact image descriptors, the BoW model decomposes an image into a bag of visual elements, providing "word"-level granularity representations. This is a desired property when searching small instances or multiple objects in a large image corpus. Equipped with large visual vocabularies, the BoW features become very sparse and therefore the inverted index can be incorporated for efficient storage and retrieval. Previous success of the SIFT base BoW models and the emerging trends of local deep descriptors motivate us to design methods to combine the best of both worlds. Little works [20, 25] have been done on this direction, and further investigations are required for the retrieval model and codebook construction. In this paper, we answer the following question: *How to effectively apply the local deep descriptors in the BoW models with large codebooks?*

We make the following contributions:

(1) For feature extraction, we replace the conventional affine region detector and descriptor with a pre-trained CNN. An effective method, namely high-norm selection, is proposed (section 4.1) to select the most discriminative local deep descriptors according to their neuron activations. Different from previous works, dividing the image into multiple patches before feeding to the network is not required in our model. The full-size image goes through the network only once, which speeds up the feature extraction. Moreover, different deep descriptor pre-processing methods are evaluated (section 4.2).
(2) Two classic issues of the BoW models: burstiness [14] and quantization error [30] are revisited in section 4.3 and section 4.4 respectively. We show that despite of the different behaviors of the CNN features and handcraft descriptors, these issues still persist in our scenario. Addressing these problems further improves the retrieval accuracy by 10% relatively on average.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ThematicWorkshops'17, October 16–20, 2017, Snowbird, UT, USA.
© 2017 ACM. ISBN 978-1-4503-4916-1/17/10…$15.00
DOI: https://doi.org/10.1145/3126686.3127018

(3) Our model uses a large visual vocabulary (up to 1 million), which results in sparse BoW histograms as image features. Inverted index is adopted for efficient storage and fast retrieval. The discriminative deep descriptors allow us to obtain better initial retrieval results, providing a good starting point for re-ranking methods. The effects of different settings during codebook construction are also evaluated.

Extensive experiments show that our model achieves state-of-the-art performance (section 4.6) without using re-ranking methods, e.g., spatial verification and query expansion. Related work will be discussed in section 2 and the paper is concluded in section 5.

2 RELATED WORK

In the past decade, image retrieval is mainly handled by the methods using local invariant descriptors, such as SIFT [22]. Previous work can be roughly divided into two categories: 1) methods that encode local descriptors into large visual codebooks and sparse representations, namely Bag-of-Words (BoW) [13, 24, 29, 38, 39]; and 2) methods that aggregate local descriptors into dense and compact features [4, 15, 16, 28]. Due to the loss of spatial information and the degradation of discriminative power of the descriptor after visual word quantization, BoW models are usually followed by some post-processing steps, e.g., spatial verification [29] or query expansion [9], in order to eliminate false positive results. A different strategy is to aggregate the local descriptors into compact representations, e.g., compressed Fisher Vector [28], VLAD [4, 15] and T-embedding [16]. However, recent studies [5, 6, 18, 32, 33, 40] show that the neuron activations extracted from CNN serve as good image representations, which surpass conventional features in low dimensionality settings.

CNNs are widely used in computer vision since the success of "AlexNet" [19] in large-scale image classification [34]. Recent studies [6, 32, 42] show that the neuron activations of CNNs can be used as generic features for image retrieval, where the features are from the *fully-connected layers*. However, these layers are trained on labeled objects to facilitate image classification and hence might not generalize to some instance types. These methods usually require fine-tuning the CNN on the target (or visually similar) datasets [6, 7, 42] to obtain satisfactory retrieval performance.

Besides of the fully-connected layers, there is an emerging trend [1, 5, 11, 18, 31, 33, 40] toward using the activations of the convolutional layers, named as *convolutional feature maps* (CFMs), as image features which shows superior performance. Specifically, Razavian et al. [33] propose to segment the image into multiple square patches and extract patch descriptors using CNN. During searching, they cross-match all the patches to obtain the best match results. Obviously, this method cannot handle large-scale datasets due to the high computational cost. Babenko et al. [5] propose a simple but effective CFMs aggregation method based on sum-pooling, which generates compact global representations (256 dimensions) for retrieval. But their performance still lags behind the traditional methods. Tolias et al. [40] propose an aggregation method which first decomposes the CFMs into multiple regions at different scales and then aggregates them via sum-pooling. This method outperforms [5] in most cases. Both Radenovic et al. [31] and Gordo et al. [11] show that fine-tuning the CNNs can further improve the performance of CFMs-based features.

Table 1: Characteristics of the benchmark datasets.

	# images	# queries	# descriptors
Ox5k	5,063	55	15.1M
Pa6k	6,392	55	18.8M
Scu.	3,170	70	8.6M
Hol.	1,491	500	4.3M

3 FRAMEWORK AND EVALUATION

We first introduce the basic settings of our baseline framework, including the feature extraction process, the BoW model, and the benchmark datasets for testing. Improvements of different parts in our framework will be provided in the following sections.

For image features, we use VGG19 [37] provided by Caffe [17] and extract the CFMs from the last convolutional layer as in [5, 18, 40]. The original image sizes are kept during feature extraction. All the input images are zero-centered by *RGB mean pixel subtraction* [12]. The output CFMs $\mathbf{X} \in \mathbb{R}^{H\times W\times D}$ of an image can be equivalently represented as a set of local descriptors $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_i \in \mathbb{R}^D | i \in \{1, \ldots, H \times W\}\}$, where H and W are height and width of each feature map, and D denotes the number of feature maps (or channels) in that layer ($D = 512$ in our case). Finally, we perform L_2-normalization for all the local descriptors.

For the BoW framework, we follow the standard model described in [29]. Approximate K-means[1] is applied to generate the visual codebook with the size of 500,000 words[2]. The local descriptors of each image are then hard assigned to their closest words using approximate nearest neighbor method. We use the standard tf-idf weighting scheme and compute the image similarity by L_2 distance. The image similarities are computed efficiently using an inverted index. Note that in this baseline, the visual codebooks are learned on the target datasets respectively.

Four datasets are chosen for evaluation in the experiments: Oxford5k [29], Paris6k [30], Sculptures [2] and INRIA Holidays [13]. These datasets have diverse types of images: Oxford5k and Paris6k are landmarks related, while Sculptures focuses on textureless sculptures, and INRIA Holidays contains a variety of scenes/objects. The diversity of image types help us to better evaluate the performance of different methods. The query bounding boxes are used for cropping out the target objects during retrieval when provided. The evaluation metric is the mean average precision (mAP) for all the datasets. The characteristics of these datasets are summarized in table 1. The baseline results are shown in table 2.

4 IMPROVING THE NAIVE BOW MODEL

4.1 Descriptor selection

In our baseline approach, all the local descriptors of given a database image are used when generating its BoW histogram representation. However, in previous literatures, the *keypoints* (or interest points) of an image are first detected by feature detectors (e.g.,

[1] The VLFeat [41] library is used for implementations.
[2] Various codebook sizes will be evaluated in section 4.5.

Table 2: The mAPs of baseline on four datasets with different feature selection methods. "+ keypoints": baseline with keypoints selection and "+ high-norm": baseline with high-norm selection (60% selection rate).

	Ox5k	Pa6k	Scu.	Hol.
baseline	0.656	0.739	0.402	0.774
+ keypoints	0.718	0.785	0.219	0.770
+ high-norm	**0.762**	**0.816**	**0.472**	**0.814**

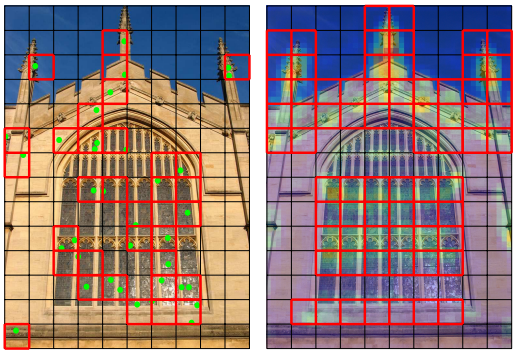
Difference-of-Gaussians (DoG) [22], Hessian-Affine [23]), which are then described by SIFT descriptors. The keypoint detection reduces the number of unnecessary local descriptors in order to provide better image representations. Here we propose two novel descriptor selection strategies for the CFMs local descriptors, as illustrated in figure 1.

- Keypoints selection: we apply the covariant region detectors (e.g., DoG [22]) to obtain a set of keypoints for an image. Given the spatial shape of CFMs is $H \times W$, we use a uniform square mesh to evenly divide the image into $H \times W$ patches, each of which corresponds to one local descriptor. We select the descriptors that are with at least one keypoint inside their patches. Note that the local descriptor' receptive field on the image might be different from the assigned patch. We empirically find that this simple strategy produce reasonable results;
- High-norm selection: the CFMs local descriptors with large norms ($||\mathbf{x}_i||_1$) are more discrimiantive [5]. Inspired by this observation, for the second method, we extract a certain percentage of descriptors that are with the largest L_1-norms.

During the BoW histogram generation, instead of using all the local descriptors, we apply the proposed selection methods to filter out unwanted descriptors. The results of these two methods are shown in table 2 and figure 2. From table 2 we see that keypoint selection increases the mAPs of two landmark datasets. However, there is a large accuracy drop on the Sculptures dataset. The reason is that the DoG detector is inappropriate for detecting keypoints from smooth objects.

For the high-norm selection, the results of various selection rate on Oxford5k in figure 2 show that filtering out low-norm descriptors significantly improves the accuracy. This observation is consistent with Babenko and Lempitsky's findings [5]. It is interesting to see that the accuracy is still higher that the baseline even 80% of the descriptors are discarded. In general, the average selected percentage of keypoints selection is around 42 - 60%. Table 2 shows that the high-norm selection consistently improves the performance on all datasets. Besides of the robust improvements, another merit is that we *only need one CNN* for both feature extraction and selection without using any external detectors.

Therefore, we use the high-norm selection with 60% selection rate for all datasets without further fine-tuning this parameter in our experiments.



(a) (b)

Figure 1: Different feature selection strategies: (a) keypoints selection; and (b) high-norm selection. The grids over the images denote $H \times W$ patches. The green dots in (a) are the detected keypoints. The warmer areas in (b) represent local descriptors with larger norms. The red boxes denote the selected local descriptors. (Best viewed in color.)

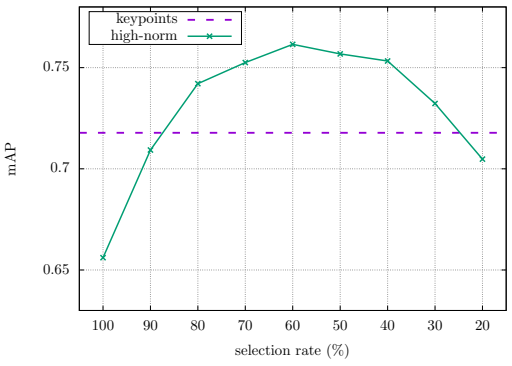


Figure 2: The mAPs of different high-norm selection rates on Oxford5k.

4.2 Descriptor pre-processing

Previously, the local descriptors are L_2-normalized after extraction. In this section, we evaluate three other pre-processing methods:

- L_1-normalization;
- RootCFM: inspired by RootSIFT [3], we pre-process the CFMs local descriptor by taking the square root of each element and L_2-normalization. When comparing the processed descriptors, namely RootCFM, using Euclidean distance, it is equivalent to using the Hellinger kernel to compare the original descriptors;
- L_2-PCAw-L_2: similar to [5, 25, 40], the descriptors are processed with L_2-normalization, PCA-whitening, and L_2-normalization at last.

The results are shown in table 3. The RootCFM generally achieves the best performance among the four methods. The effectiveness

Table 3: The mAPs of different pre-processing methods. The retrieval model is the baseline with high-norm descriptor selection (c.f. section 4.1).

	Ox5k	Pa6k	Scu.	Hol.
L_2-norm	0.762	0.816	0.472	0.814
L_1-norm	0.741	**0.823**	0.424	0.808
RootCFM	**0.788**	0.820	**0.498**	0.843
L_2-PCAw-L_2	0.748	0.770	0.417	**0.845**

of RootCFM on other tasks (e.g. image classification) will be investigated in future work. In the following sections, RootCFM preprocessing is applied.

4.3 Burstiness

Visual word burstiness [14] is a phenomenon that certain words appear many times in an image due to the repetitive of visual elements. The bursty features will dominate the similarity measure and therefore compromise the image retrieval accuracy. In this section, we show that burstiness still persists in our case and a word weighting normalization is performed to tackle the problem.

Figure 3 shows the top five busty words (the ones with highest term frequencies) of different images. Obviously, burstiness occurs when there are repetitive structures or similar textures. Due to the overlaps of receptive fields in convolutional filters, nearby local descriptors are usually assigned to the same words and therefore the bursty words appear in groups.

To quantitatively measure burstiness, we plot the visual word distribution in figure 4. For each visual word k, its term frequency (TF) in image i is denoted as $v_{i,k}$. We count its maximum TF M_k across the image collection, i.e., $M_k = \max_i v_{i,k}$. The number of words with certain value of M_k is shown in figure 4a. Similarly, we denote the maximum TF in image i as N_i (i.e., $N_i = \max_k v_{i,k}$), and the number of images with certain value of N_i is shown in figure 4b. Figure 4a shows that a majority of words maximally appear three times at most in an image. Meanwhile, in figure 4b, most images have a maximum TF larger than 10. The above observations suggest the prevalent existence of burstiness in general images, which violates the assumption in the BoW models that visual words are emitted independently in the image.

Our solution to tackle the burstiness problem is given below. The similarity of two images' BoW represenations can be interpreted as a voting score of their matched local descriptors [13]. Specifically, given the query image $\mathbf{Q}$ and its local descriptors $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_m\}$, the database image $\mathbf{D}$ and its descriptors $\{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n\}$. The tf-idf weighting similarity between $\mathbf{Q}$ and $\mathbf{D}$ is measured by:

$$S_{\text{tf-idf}}(\mathbf{Q}, \mathbf{D}) = \sum_{k=1}^{K} \sum_{\substack{(\mathbf{q}_i, \mathbf{d}_j) \\ \mathbf{q}_i \in \mathbf{Q}, \mathbf{d}_j \in \mathbf{D} \\ w(\mathbf{q}_i) = w(\mathbf{d}_j) = k}} \text{idf}^2(k), \tag{1}$$

where k denotes the k-th visual word, K is the codebook size, $w(\mathbf{q}_i) = w(\mathbf{d}_j) = k$ means $\mathbf{q}_i$ and $\mathbf{d}_j$ are both assigned to word k, and $\text{idf}(k)$ is the inverse document frequency (IDF) of word k.

Table 4: The mAPs of different scoring strategies. The retrieval model is the baseline with high-norm descriptor selection and RootCFM.

	Ox5k	Pa6k	Scu.	Hol.
$S_{\text{tf-idf}}$	0.788	0.820	0.498	0.843
S_{burst}	**0.818**	**0.838**	**0.535**	**0.847**

Note that this score will be normalized by the L_2-norms of $\mathbf{Q}$ and $\mathbf{D}$'s BoW histograms at last [13].

To handle the burstiness, the new scoring function is:

$$S_{\text{burst}}(\mathbf{Q}, \mathbf{D}) = \sum_{k=1}^{K} \sum_{\substack{(\mathbf{q}_i, \mathbf{d}_j) \\ \mathbf{q}_i \in \mathbf{Q}, \mathbf{d}_j \in \mathbf{D} \\ w(\mathbf{q}_i) = w(\mathbf{d}_j) = k}} \frac{\text{idf}^2(k)}{\text{tf}_{\mathbf{D}}(k)}, \tag{2}$$

where $\text{tf}_{\mathbf{D}(k)} = v_{\mathbf{D},k}$ denotes the TF of word k in image $\mathbf{D}$. This is a normalization term to reduce the impact of the bursty word in the same image. There is no further normalization for the final score. The results of these two scoring strategies are shown in table 4, and S_{burst} consistently outperforms $S_{\text{tf-idf}}$. Unlike [14] where normalizing the score directly by the number of occurrences of the visual word is found too hard, our experiments demonstrate that this simple normalization method is better. It might due to the differences between burstiness patterns of the CFMs local descriptors and the ones of SIFT. Note that there is no modification required for the other parts of the retrieval model, both scoring functions can be computed efficiently using the inverted index.

4.4 Soft-assignment

With a learned codebook, the local descriptor is quantized to its nearest visual word. This "hard"-assignment might lead to large quantization error because of the information loss and the visual word ambiguity. In [21, 30], "soft"-assignment (SA) is proposed to alleviate this drawback by assigning each local descriptor to n nearest visual words instead of one, and the weight to each word is proportional to $\exp(-\beta d^2)$, where d is the distance of the descriptor to the assigned word.

We evaluate the effectiveness of SA in our case and the results are shown in figure 5. Here we assign each descriptor to its three nearest neighbors (i.e. $n = 3, \beta = 0$) for all datasets without fine-tuning this parameters. As expected, SA further improves the retrieval accuracies on most of the datasets. Since the average list length of the inverted file becomes n times longer and there are up to n times more visual words need to be evaluated for the query, SA takes n times higher memory consumption and requires n^2 times longer query time.

Finally, we summarize the improvements obtained when all the proposed methods are employed in figure 5. Compared with the baseline, there are 27%, 18%, 42% and 9% relative increases in terms of mAP for the Oxford5k, Paris6k, Sculptures, and INRIA Holidays respectively. These results are promising especially when considering no post-processing steps (e.g., spatial verfication [29] and query expansion [3, 10]) are required.

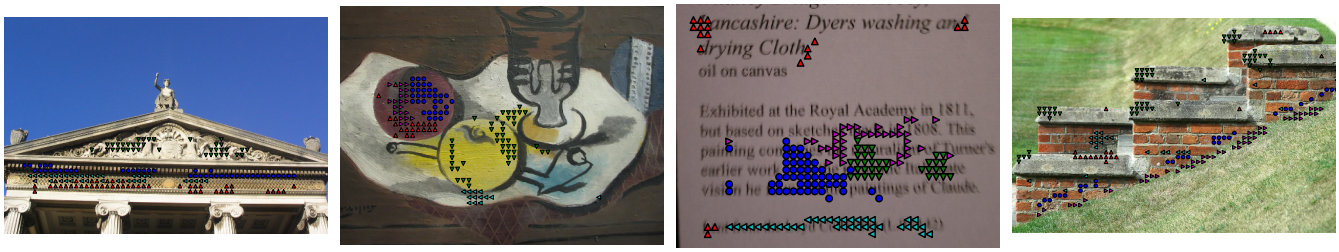


Figure 3: Illustration of burstiness. Different colors and markers denote different visual words.

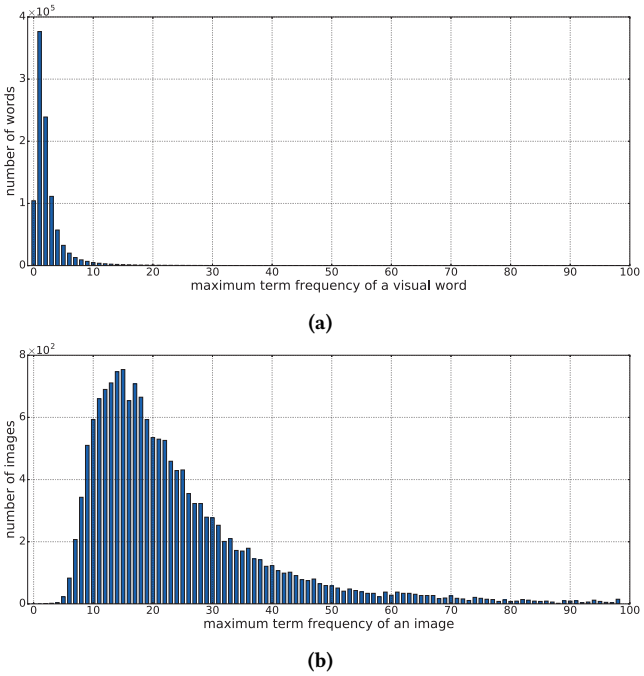


Figure 4: (a) Histogram of visual words with different maximum term frequencies; (b) Histogram of images with different maximum term frequencies. The data is evaluated on all 16,116 images of the four datasets, and the codebook is learned on collected Flickr images and its size is 1 million (See section 4.5 for details).

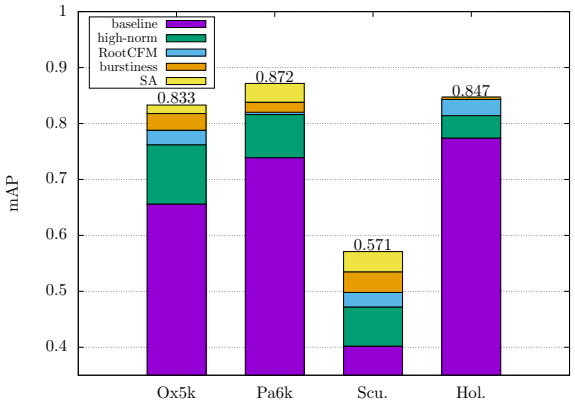


Figure 5: The improvements over the baseline.

4.5 Codebook construction

There are two major concerns when constructing the visual codebook: where is the codebook learned from and what is the proper codebook size. So far, the visual codebook is learned on the target dataset. Previous studies [3, 30] show that learning the visual codebook on an independent dataset will diminish the performance. On the other hand, the optimal size of codebook varies in different scenarios [24, 29]. In this section, we study the effects of these two factors.

Following [13], we retrieve around 109,000 images from Flickr, namely Flickr109k, for codebook learning. We use the same feature extraction and codebook construction pipeline on this dataset. The codebook sizes ranging from 25K to 1M are evaluated. The results are shown in figure 6. When the codebooks are learned on the target datasets, the performance curves are quite flat for all datasets except the Sculptures as shown in figure 6a. As the codebook size marginally changes the retrieval performance, we stop testing larger codebook size for this case. Similar behaviors can be observed when the codebooks are learned on an independent dataset (see figure 6b). Comparing figure 6a and 6b, the accuries degrade on all datasets, which is consistent with previous findings [3, 30].

4.6 Comparison with existing work

We compare our model with the state of the art in table 5. For all the BoW, only methods without post re-ranking are considered. According to whether the codebooks are learned on independent datasets, we divide them into two different groups. Our model outperforms all the SIFT-based BoW methods [24, 27, 35, 43] on all datasets except [27] on Oxford5k. These results are more notable when considering no re-ranking methods are adopted. Both [25] and [20] use CNN features as local descriptors in BoW models. However, their codebook sizes are small compared to ours. In [20], the features are extracted from multiple layers of the CNN in a sliding windows fashion on a single image. In contrast, our model only needs to feed the whole image into the network once, which requires less computation efforts.

For completeness, we summarize recent studies [1, 5, 11, 18, 31, 40] on aggregating the CFMs into a global descriptor for image retrieval. The dimensionalities of the resulting global descriptors

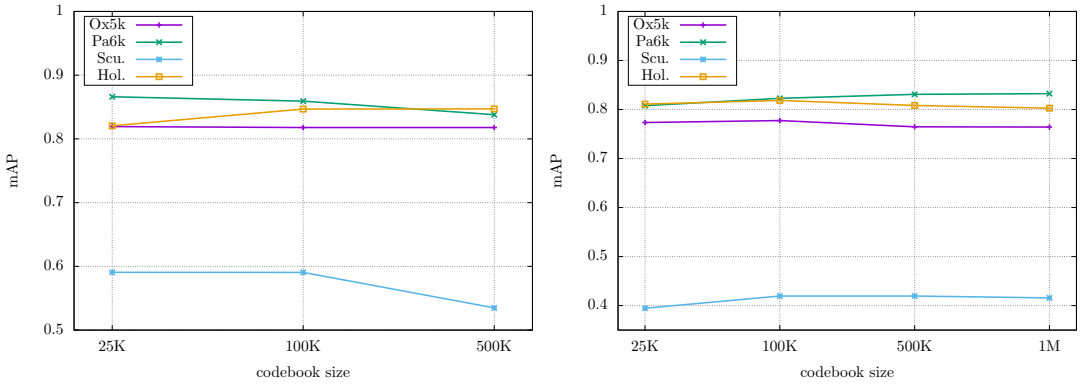


Figure 6: The mAPs of different codebook size. (a) The codebooks are learned seperately on the target datasets; (b) The codebooks are learned on the Flickr109k dataset.

Table 5: Accuracy comparison with the state of the art. "Feature": the feature detectors/extractors; "Words": codebook size; "Target": codebook learned on the target datasets; "D": Dimensionality; "FT": fine-tuning. Scores marked with a * manually rotate Holidays images to up-right orientation.

	Method	Feature	Words	Target	Ox5k	Pa6k	Scu.	Hol.
Bag-of-Words	Shen et al. [35]	Hesaff [27]	1M	Y	0.752	0.741	-	0.762
	Zheng et al. [43]	DoGaff [36]	1M	Y	0.744	0.759	-	0.654
	Perdoch et al. [27]	Hesaff [27]	1M	Y	**0.846**	-	-	-
	Mohedano et al. [25]	VGG [37]	25K	Y	0.738	0.820	-	-
	Li et al. [20]	CNN-M [8]	20K	Y	-	-	-	**0.858**
	Ours	VGG [37]	500K	Y	0.833	**0.872**	**0.571**	0.846
	Perdoch et al. [27]	Hesaff [27]	1M	N	0.725	-	-	0.769
	Mikulik et al. [24]	Hesaff [27]	16M	N	0.742	0.749	-	0.749*
	Ours (on Flickr109k)	VGG [37]	1M	N	**0.792**	**0.843**	**0.441**	**0.818**
Global descriptors	Method	Feature	D	FT	Ox5k	Pa6k	Scu.	Hol.
	Babenko & Lempitsky [5]	VGG [37]	256	N	0.589	-	-	0.802*
	Tolias et al. [40]	VGG [37]	512	N	0.669	**0.830**	-	-
	Kalantidis et al. [18]	VGG [37]	512	N	**0.708**	0.797	-	**0.851***
	Arandjelovic et al. [1]	VGG [37]	256	Y	0.634	0.715	-	0.768
	Radenovic et al. [31]	VGG [37]	512	Y	0.801	0.850	-	0.825*
	Gordo et al. [11]	VGG [37]	512	Y	**0.831**	**0.871**	-	**0.867**

are usually ranging from 256 to 512. For [5, 18, 40], features are extracted using the pre-trained CNNs. Nevertheless, more recnet studies [1, 11, 31] show that fine-tuning the CNNs on task-specific dataset (e.g., landmark retrieval) can provide further improvements. Our method can also benefit from the success of these methods. For example, we can use the fine-tuned CNN for feature extraction on task-specific retrieval.

5 CONCLUSION

We propose an efficient bag-of-words model using local deep descriptors from the convolutional neural network. The high-norm descriptor selection provides a simple and effective way to choose the most discriminative local descriptors, which improves the retrieval accuracy significantly. Different descriptor pre-processing methods are evaluated and the RootCFM is found to be the best. We demonstrate that the problems of burstiness and quantization error still persist in our scenario and addressing these issues provides further improvements on accuracy. Our model uses a large visual codebook combined with inverted index for efficient storage and fast retrieval.

The efforts we have made suggest that the lessons learned from the past SIFT-based methods can help us to better customize the BoW model for deep features. There are several directions for the future work, such as designing spatial verification methods for deep descriptors, utilizing different layers' features from a CNN, encoding the spatial information of convolutional feature maps into inverted index, combining SIFT and CNN features to enhance the BoW model, and so on.

REFERENCES

[1] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*.

[2] Relja Arandjelovic and Andrew Zisserman. 2011. Smooth object retrieval using a bag of boundaries. In *ICCV*.

[3] Relja Arandjelovic and Andrew Zisserman. 2012. Three things everyone should know to improve object retrieval. In *CVPR*.

[4] Relja Arandjelovic and Andrew Zisserman. 2013. All About VLAD. In *CVPR*.

[5] Artem Babenko and Victor S. Lempitsky. 2015. Aggregating Deep Convolutional Features for Image Retrieval. In *ICCV*.

[6] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. 2014. Neural Codes for Image Retrieval. In *ECCV*.

[7] Jiewei Cao, Zi Huang, Peng Wang, Chao Li, Xiaoshuai Sun, and Heng Tao Shen. 2016. Quartet-net Learning for Visual Instance Retrieval. In *MM*.

[8] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *BMVC*.

[9] Ondrej Chum, Andrej Mikulík, Michal Perdoch, and Jiri Matas. 2011. Total recall II: Query expansion revisited. In *CVPR*.

[10] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. 2007. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *ICCV*.

[11] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. 2016. Deep Image Retrieval: Learning Global Representations for Image Search. In *ECCV*.

[12] Forrest N. Iandola, Matthew W. Moskewicz, Sergey Karayev, Ross B. Girshick, Trevor Darrell, and Kurt Keutzer. 2014. DenseNet: Implementing Efficient ConvNet Descriptor Pyramids. *CoRR* abs/1404.1869 (2014).

[13] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *ECCV*.

[14] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2009. On the burstiness of visual elements. In *CVPR*.

[15] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. 2012. Aggregating Local Image Descriptors into Compact Codes. *TPAMI* (2012).

[16] Hervé Jégou and Andrew Zisserman. 2014. Triangulation Embedding and Democratic Aggregation for Image Search. In *CVPR*.

[17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *MM*.

[18] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *ECCV Workshops*.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

[20] Ying Li, Xiangwei Kong, Liang Zheng, and Qi Tian. 2016. Exploiting Hierarchical Activations of Neural Network for Image Retrieval. In *MM*.

[21] Lingqiao Liu, Lei Wang, and Xinwang Liu. 2011. In defense of soft-assignment coding. In *ICCV*.

[22] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* (2004).

[23] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. 2005. A Comparison of Affine Region Detectors. *IJCV* (2005).

[24] Andrej Mikulík, Michal Perdoch, Ondrej Chum, and Jiri Matas. 2013. Learning Vocabularies over a Fine Quantization. *IJCV* (2013).

[25] Eva Mohedano, Kevin McGuinness, Noel E. O'Connor, Amaia Salvador, Ferran Marqués, and Xavier Giró i Nieto. 2016. Bags of Local Convolutional Features for Scalable Instance Search. In *ICMR*.

[26] David Nistér and Henrik Stewénius. 2006. Scalable Recognition with a Vocabulary Tree. In *CVPR*.

[27] Michal Perdoch, Ondrej Chum, and Jiri Matas. 2009. Efficient representation of local geometry for large scale object retrieval. In *CVPR*.

[28] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. 2010. Large-scale image retrieval with compressed Fisher vectors. In *CVPR*.

[29] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

[30] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*.

[31] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. 2016. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *ECCV*.

[32] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *CVPR Workshops*.

[33] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. 2015. A Baseline for Visual Instance Retrieval with Deep Convolutional Networks. In *ICLR Workshops*.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* (2015).

[35] Xiaohui Shen, Zhe Lin, Jonathan Brandt, Shai Avidan, and Ying Wu. 2012. Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking. In *CVPR*.

[36] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Learning Local Feature Descriptors Using Convex Optimisation. *PAMI* (2014).

[37] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[38] Josef Sivic and Andrew Zisserman. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*.

[39] Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W. M. Smeulders. 2014. Locality in Generic Instance Search from One Example. In *CVPR*.

[40] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*.

[41] A. Vedaldi and B. Fulkerson. 2008. VLFeat: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/. (2008).

[42] Ji Wan, Dayong Wang, Steven Chu-Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. 2014. Deep Learning for Content-Based Image Retrieval: A Comprehensive Study. In *MM*.

[43] Liang Zheng, Shengjin Wang, and Qi Tian. 2014. Lp-Norm IDF for Scalable Image Retrieval. *TIP* (2014).