Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1603 – 1618

doi:10.3906/elk-1803-3

Combined feature compression encoding in image retrieval

Lu HUO*, **Leijie ZHANG**
College of Computer Science and Technology, Hangzhou Dianzi University,Hangzhou, P.R. China

Received: 01.03.2018 • **Accepted/Published Online:** 24.02.2019 • **Final Version:** 15.05.2019

Abstract: Recently, features extracted by convolutional neural networks (CNNs) are popularly used for image retrieval. In CNN representation, high-level features are usually chosen to represent the images in coarse-grained datasets, while mid-level features are successfully applied to describe the images for fine-grained datasets. In this paper, we combine these different levels of features as a joint feature to propose a robust representation that is suitable for both coarse-grained and fine-grained image retrieval datasets. In addition, in order to solve the problem that the efficiency of image retrieval is influenced by the dimensionality of indexing, a unified subspace learning model named spectral regression (SR) is applied in this paper. We combine SR and the robust representation of the CNN to form a combined feature compression encoding (CFCE) method. CFCE preserve the information without noticeably impacting image retrieval accuracy. We find the tendency of the image retrieval performance to change the compressed dimensionality of features. We further discover a reasonable dimensionality of indexing in image retrieval. Experiments demonstrate that our model provides state-of-the-art performances across datasets.

Key words: Convolutional neural networks, feature selection, image retrieval, spectral regression

1. Introduction

After AlexNet [1] broke many records, convolutional neural networks (CNNs) have achieved great successes in a number of computer vision tasks, including object detection [2], human action recognition [3], visual recognition [4], and semantic segmentation [5]. As deep features create higher discriminative ability and semantic representation power, and different distribution properties, their representation provides powerful descriptors for image retrieval. Many researchers employ this type of deep learning model to solve problems in specific domains. In terms of representation (features), the performance of image retrieval is deeply dependent on the choice of feature selection [6]. Many studies reveal that deep features are viable alternatives to traditional hand-engineered features [7–9].

In image retrieval, the encoder process tries to preserve as much information about the image as possible. Even though describing images with a high-dimensional vector maintains higher discriminative power than a low-dimensional one, a high-dimensional indexing vector falls prey to "the curse of dimensionality" [10]. This problem may decrease the indexing efficiency of image retrieval and search results fall behind the brute-force linear search. Generally speaking, generating a compact indexed representation is essential in image retrieval.

This paper focuses on generating compact representation that is well suited for image retrieval. Unlike the singular deep features applied to most state-of-the-art large-scale retrieval systems, our work is mainly

*Correspondence: Lu_Huo_77@163.com

focused on feature combination, and we use these combined features in subspace learning to generate compact representation. In addition, we discuss the suitable encoding bits in compression.

The contributions of our work are threefold:

- We join different types of features extracted by a CNN to form a representation that can convey image information such as local position, object part, and mixture of patterns.
- We apply subspace learning as a dimensionality reduction method and find that subspace learning not only reduces the dimensionality without noticeably impacting image retrieval accuracy but also excludes the redundant information and preserves the valid information.
- After many experiments, we find the regular pattern of image retrieval performance with the growth of compressed dimensionality of features. In addition, we discover a reasonable length of encoding when deep features are applied in image retrieval.

The rest of the paper is organized as follows. Section 2 reviews the choice of features and feature encoding in image retrieval. Section 3 introduces the detailed theory of the combined feature compression encoding (CFCE) method. Section 4 compares our method with different methods popularly used in this field, and Section 5 concludes the paper and discusses the results.

2. Related work

The process of image retrieval is divided into two parts: the choice of features (local or global) and feature encoding, aggregating them into a reasonable vector. This section briefly reviews the development of these two processes and introduces the spectral regression.

2.1. Choice of features

Different types of features may cause different effects with respect to different datasets. In some coarse-grained or genetic classification datasets, the global (high-level) features outperform local (middle-levels) features [8, 11]. Besides, local features are superior to global features [12–14] in some fine-grained or identical classes.

Hand-crafted features are applied to traditional state-of-the-art image retrieval systems. Local representation, such as scale-invariant feature transform (SIFT) [15] and local binary patterns(LBP) [16], can be aggregated into fixed-length vectors to describe a whole image used in image retrieval. Moreover, global representations such as GIST [17] and histograms [18] are also used in image retrieval. These two types of hand-crafted features are, however, used separately with respect to specific domains. Hand-crafted features need to be redesigned if the domain changes. Several works have shown that deep descriptors significantly outperform the state-of-the-art approach on common retrieval benchmarks [8, 19, 20]. Therefore, the focus in computer vision has shifted from traditional hand-crafted features towards the deep features produced by CNN.

The CNN algorithm is a particular kind of representation learning procedure that discovers multiple levels of representation, with higher-level features representing more abstract aspects of the data [21]. As shown in Figure 1, with an increasing number of layers, the corresponding size of receptive fields becomes larger, and the types of features represented shift from local to global features. CNN architectures can potentially generate progressively abstract features in global representations of higher layers that are only sensitive to some very specific types of changes in the input [21]. Even though the representations in higher layers are more likely to represent object parts, they may represent a mixture of patterns. Such complex knowledge representations in

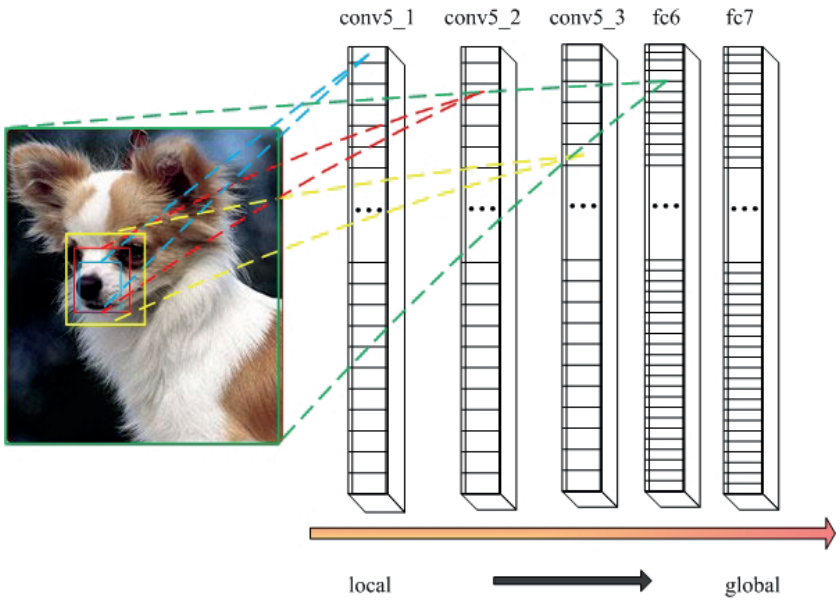


Figure 1. Features in convolutional layers and fully connected layers.

higher layers diminishes the interpretability of the network. In contrast, an interpretable CNN is activated by a certain portion of the image in local representation [22].

Razavian et al. [8] revealed that features obtained from CNNs should be the primary candidates in most visual recognition tasks. Babenko et al. [7] used local features extracted from the convolutional layers to describe particular regions of whole images. Then, using an aggregation strategy, they embedded these local features into a vector representation for the whole image. Elisha et al. [23] revealed that in networks performing well the representations in general "improve" from layer to layer and smoothness improves from layer to layer. Razavian et al. used a feature representation of size 4096, extracted from the first fully connected layer, to achieve better performance than state-of-the-art retrieval pipelines. However, using a fully connected layer will cause some loss of information or break the aspect ratio, which is harmful to the task of visual retrieval while convolutional layers preserve more spatial information [24]. Zheng et al. [25] observed that average/max pooling of features from intermediate layers is effective in improving invariance to image translations. Specifically, the pooled *conv*5 feature, with much lower dimensionality, was shown to yield competitive accuracy with the FC features. In addition, a high-layer filter (FC features) may represent a mixture of patterns, which will greatly decrease the performance of the fine-grained image retrieval [26].

Therefore, these two types of features may be not applicable to different types of datasets. From the point of view of CNN representation, a single type of deep features may not be suitable for different kinds of datasets. We combine the different types of features to capture more diverse information biased toward visual appearance for both of these types of datasets.

2.2. Feature encoding

Fisher vector(FV) [27], vector of locally aggregated descriptors (VLAD) [28], and bag of features (BOF) [29] are leading aggregation local descriptor approaches. These approaches project each local descriptor into different components or visual words of a codebook. Then all encoded vectors are aggregated into a single vector

using sum or average operations. Even though these approaches achieve better performance than those used before, their shortcoming is that the dimensionality of the final encoding bits is quite high. Projecting the original feature into latent space can solve the problem of dimensionality. Each feature is mapped into a fixed-length vector. Many deep learning models use latent nodes corresponding to encoding bits, such as restricted Boltzmann machines (RBMs), autoencoder (AE), and CNNs. The values of the latent variables in the deepest layer of RBMs are easy to infer and give a much better representation, as proposed by Salakhutdinov et al. [30]. In latent space, semantically similar images are mapped to nearby addresses. Carreira-Perpinan et al. [31] introduced the binary AE model, which consists of an encoder to generate codes and a decoder to reconstruct each images. Each high-dimensional image is mapped onto a binary, low-dimensional vector. Yang et al. [32] increased the hidden layer in CNNs. In this model, the number of hidden nodes corresponds to the encoding bits. The value of hidden nodes is adjusted with classification loss. Furthermore, if the data labels are available, this deep CNN model may generate encoding and image descriptors simultaneously. Although these methods may generate dense encodings of high quality, once the encoding bits change, these deep models also need to be retrained.

In contrast, an alternative approach is using subspace learning algorithms. This method acts as a dimensional reduction method to discover the discriminant structure in feature space and preserve the information of features without noticeably impacting the image retrieval accuracy.

2.3. Spectral regression

For graph-based subspace learning, the purpose of graph embedding is to represent each vertex of a graph as a low-dimensional vector that preserves similarities between the vertex pairs, where similarity is measured by the edge weight. In a graph G with k vertices, each vertex corresponds to a feature point. Let W be a symmetric $m \times m$ matrix with W_{ij} denoting the weight of the edge joining vertices i and j. Suppose $y = [y_1, y_2, ..., y_k]^T$ is the projection of the graph onto the real line. The y is given by minimizing

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} = 2y^T Ly, \tag{1}$$

where $L = D - W$ according to the graph Laplacian and $D_{ii} = \sum_j W_{ij}$. We can finally get and the optimal y by maximizing eigenvectors of the eigenproblem:

$$Wy = \lambda Dy. \tag{2}$$

If we choose a linear function, i.e. $y_i = f(\mathbf{x}_i) = \mathbf{a}^T\mathbf{x}_i$, to map $\mathbf{y}$ and all samples $\mathbf{X}$, we have $\mathbf{y} = \mathbf{X}^T\mathbf{a}$. Therefore, the optimal a values are the eigenvectors corresponding to the maximum eigenvalue of the eigenproblem:

$$\mathbf{XWX}^T\mathbf{a} = \lambda\mathbf{XDX}^T\mathbf{a}, \tag{3}$$

and the following theorem is given to solve Eq. (3) more efficiently.

Theorem 1 *Let* $\mathbf{y}$ *be the eigenvector of the eigenproblem in Eq. (2) with eigenvalue* λ*. If* $\mathbf{X}^T\mathbf{a} = \mathbf{y}$*, then* $\mathbf{a}$ *is the eigenvector of the eigenproblem in Eq. (3) with the same eigenvalue* λ*.*

Theorem 1 shows that instead of solving the eigenproblem of Eq. (3), the linear embedding functions can be acquired through the following two steps: 1) solving the eigenproblem in Eq. (2) to get $\mathbf{y}$, and 2) finding $\mathbf{a}$ to satisfy $\mathbf{X}^T\mathbf{a} = \mathbf{y}$. In reality, such an $\mathbf{a}$ might not exist. A possible way is to find $\mathbf{a}$ that can best fit the equation in the least squares sense $\mathbf{a} = \arg\min_{\mathbf{a}} \sum_{i=1}^{m} (\mathbf{a}^T\mathbf{x}_i - y_i)^2$, where y_i is the ith element of $\mathbf{y}$. Such a two-step approach essentially performs regression after the spectral analysis on the graph. Therefore, Cai et al. named it spectral regression [33].

3. Combined feature compression encoding

In this section, we give details of the proposed CFCE algorithm. One of the most important purposes of our method is to find a better explanation of the input, so we compress and encode the combined representation through SR, which is a reconstruction subspace method. Meanwhile, this coding method is suitable for both fine-grained and coarse-grained datasets. We start by using the CNN to describe the image data by extracting features and combining the different types of features. To generate an efficient indexing and enhance encoding quality for image retrieval, we adopt subspace learning as a dimensionality reduction method to make a compact representation. The standard pipeline used to build combined feature compression encoding is shown in Figure 2. The first and second stage belong to combinations with different types of features. Between the third and sixth stages they belong to the dimensionality reduction method.

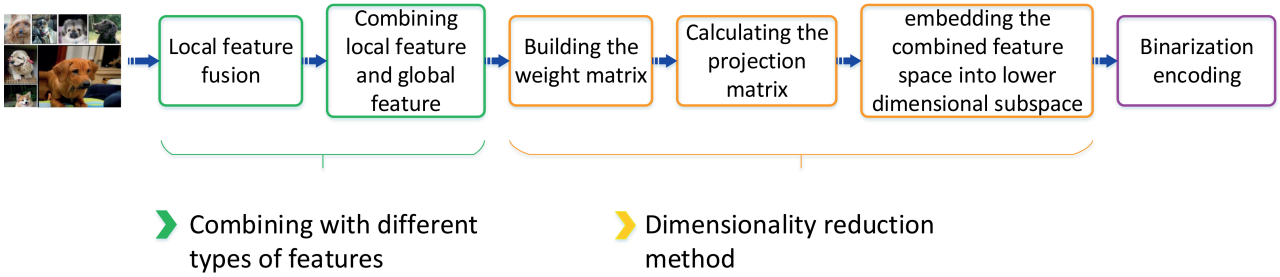


Figure 2. The standard pipeline used to build combined feature compression encoding. The first and second stages belong to combination with different types of features. Between the third and sixth stages they belong to the dimensionality reduction method.

3.1. Combining with different types of features

We describe our approach using a unified representation as a way of building compact and efficient codes for image retrieval in this section. We can reuse different levels of features in CNN architectures, since a CNN is like a hierarchical organization. In fine-grained image retrieval, local patterns in images are more important than global patterns, while in coarse image retrieval, it may be better using high-level semantic information as a representation. Unlike using high-level or middle-level features, here we combine multiple types of features. In our approach, we combine the feature maps extracted separately from a certain portion of the image in middle levels and an object part in high levels. In this method, we use an f_l layer as an aggregation of local fusion features and f_g as global features. Then we combine these two features as a joint feature.

In addition, the dimensionality of aggregated local representation is much larger than global features in CNN representation. Furthermore, the information of local features is relatively sparse. Therefore, we adopt a different pooling method to maintain the local image information.

We use the f_l layer as an aggregation vector of local fusion feature and f_g as a global feature. Let I denote a set of training images, where $I_i \in I$ represents an image in the dataset. We denote the feature maps of layer k as f^k. Then we define the average pooling feature $f^k_{avg}(x)$ as:

$$f^k_{avg}(x) = \frac{1}{w \times h} \sum_{i,j=1}^{w,h} f(x), \tag{4}$$

where k correspondings to a $conv5\_3$ layer, w and h correspond respectively to the width and height of each channel, and i represents the channel.

The max pooling feature $f^k_{max}(x)$ is:

$$f^k_{max}(x) = max^{w,h}_{i,j=1} f(x), \tag{5}$$

where k corresponds to a $conv5\_3$ layer.

We denote aggregation $f_l(I_i)$ of the local fusion feature through average and max pooling as:

$$f_l(I_i) = \sigma\left(\omega^l\left(f^k_{avg}(x) + f^k_{max}(x)\right)\right). \tag{6}$$

The value of global feature representation in the layer is:

$$f_g(I_i) = \sigma\left(\omega^g \cdot f_l(I_i)\right). \tag{7}$$

The final feature representation $f_c(I_i)$ can be interpreted as the combination of local and global features:

$$f_c(I_i) = \begin{cases} f_l(I_i) & 0 \le j < B_l \\ f_g(I_i) & B_l + 1 \le j < B_l + B_g, \end{cases} \tag{8}$$

where B_l is the node of the local fusion layer, B_g is the encoding bits of the global layer, and $B_l + B_g$ is the encoding bits of the final combined feature vector.

We use exponential linear units (ELUs) as the activation functions of f_l and f_g to enlarge the margin of the encoding boundary since ELUs saturate to a negative values with smaller inputs and decreasing propagated variation, and lead to significantly better generalization performance than ReLUs and LReLUs on networks [34]. The ELU with $\alpha > 0$ is:

$$\sigma(x) = \begin{cases} x & if\ x > 0 \\ \alpha\ exp((x) - 1) & otherwise. \end{cases} \tag{9}$$

3.2. Dimensionality reduction and encoding method

Given a high-dimensional feature $f^t(I_i)$, only a small portion of the possible factors are relevant. This sparsity of the representation allows us to find compact encoding bits from a feature in a fixed-size vector with tolerable loss of image information. Moreover, to preserve multisemantic feature information and enhance image retrieval discriminant ability, the output encoding maps through regression for projective function learning and spectral graph analysis are used to model the complicated subspace of the features. Therefore, we adopt a unified approach for subspace learning, the SR [35] method, to build a compact indexing with proper encoding bits. In the SR method, we use a different weight matrix W to simulate different graph embedding methods.

Given k features $\{f^t(I_i)\}_{i=1}^k \subset \mathbb{R}^n$, the dimensionality reduction algorithm tries to find $\{z_i\}_{i=1}^k \subset \mathbb{R}^d$, $d \ll n$, where z_i is the embedding result of $f^t(I_i)$. In a graph G with k vertices, each vertex corresponds to a feature point. Let W be a symmetric $m \times m$ matrix with W_{ij} denoting the weight of the edge joining vertices i and j. Suppose $y = [y_1, y_2, \cdots, y_k]^T$ is the projection of the graph onto the real line. Then y is given by minimizing

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} = 2y^T L y, \tag{10}$$

where $L = D - W$ according to the graph Laplacian and $D_{ii} = \sum_j W_{ij}$.

In the LDA algorithm, the weight matrix is:

$$W_{ij} = \begin{cases} 1/k_t & if\ f^c(I_i)\ and\ f^c(I_j)\ both\ belong \\ & to\ the\ tth\ class; \\ 0 & otherwise \end{cases} \tag{11}$$

where we suppose there are c classes, and the tth class has k_t samples, $k_1 + ... + k_c = k$.

In the LPP algorithm, the weight matrix is:

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} & if\ x_i \in N_k(x_j)\ and\ x_j \in N_k(x_i) \\ 0 & otherwise. \end{cases} \tag{12}$$

W_{ij} has a value when the k nearest neighbors of the point have the same label with the point.

The projection matrix a satisfies:

$$a_i = \arg\min_a \left(\sum_{i=1}^m \left(a^T f^c(I_i) - y_i \right)^2 + \alpha \parallel a \parallel^2 \right). \tag{13}$$

Let $A = [a_1, a_2, ..., a_{c-1}]$. A is an $n \times (c-1)$ transformation matrix. The samples can be embedded into $c-1$ dimensional subspace by

$$x \to z = A^T x. \tag{14}$$

Since short binary codes provide very fast searching in image retrieval, we use the binary codes to provide efficient retrieval. The binary encoding b is:

$$b = sgn(z_i - mean(i)), \tag{15}$$

where $mean(i)$ is with respect to the mean value of corresponding encoding bits of the training data.

4. Experiments

This section reports experiments using cross datasets including fine-grained and coarse-grained data. Our network is trained in a NVIDIA TESLA P40. Its GPU global memory is 24 GB and it has 3840 CUDA cores.

4.1. Datasets and settings

For coarse-grained datasets, we use Caltech 101. Caltech 101 is composed of 101 widely varied categories. Each category has 40 to 800 images. Most categories have about 50 images. Categories such as motorbike, airplane,

cannon, etc. where two mirror image views were present were manually flipped, so all instances faced the same direction. The size of each image is roughly 300×200 pixels [36].

For fine-grained datasets we used Stanford dogs and indoor scene recognition. The Stanford dogs set contain 20,580 images with 120 dog breeds and approximately 150 images per class. This dataset is extremely challenging for a variety of reasons. First, being a fine-grained categorization problem, there is little interclass variation. Second, there is very large intraclass variation [37]. The images in the indoor scene recognition dataset were collected from different sources: online image search tools (Google and Altavista), online photo sharing sites (Flickr), and the LabelMe dataset. This dataset contains a total of 15,620 images, 67 indoor categories, and at least 100 images per category [38]. All images have a minimum resolution of 200 pixels in the smallest axis.

Our method is based on VGG-16 as configuration D [39], which is one of the most popular models in CNNs and can be replaced by other CNN models. For the experimental setting, we choose 60% of the data as a training set and the remaining 40% data as test data. However, FV and VLAD are time-consuming. We randomly chose 50 instances from each class from the remaining 40% of data to form a test dataset and then we repeated this process 10 times. The result of image retrieval performance was worked out by averaging the performances produced 10 times. In this paper, the retrieval time corresponds to the mean time of each image search of the top 1000 images.

To generate a good CNN model, we are supposed to collect a large amount of labeled data. However, this process can be very expensive and unrealistic. In the real world, the existing data are usually unlabeled and unbalanced. We can use inductive transfer learning to generate suitable models in CNNs. In coarse-grained image retrieval, the source domain and target domain are very similar, i.e. $D_S = D_T$, while their learning tasks are different, i.e. $T_S \neq T_T$. We can fine-tune the original model to generate good representation. In fine-grained image retrieval, the source domain and target domain are different, i.e. $D_S \neq D_T$, and their learning tasks are different, i.e. $T_S \neq T_T$. The learning rates in different parts are different since the different parts of CNNs have different similarity. Then we freeze the weights in low conv-layers, because the low levels have already been trained very well. The mid-level and high-level layers use relatively large learning rates since these features are dissimilar compared with the original feature part.

4.2. Comparison of layer performance

Inspired by Babenko [40], we compared the neural codes of different layers to find the relations between the performance of image retrieval and deep representation. Therefore, we compare the performance of our method versus the final local and global layers through the neural codes.

Tables 1 shows the results of this experiment using different layer representations to retrieve images. It shows that our powerful representation provides a viable alternative for neural codes of different layers. With the deepening of the layers, the image retrieval performance is getting better, because the deeper layer can provide more abstract representation. However, using local final layers creates a performance bottleneck due to positional information in local features being overlooked in representation and the local information of neural codes in the convolutional layer without further processing. When we combine different types of features, the search time and dimensionality slightly increase, so we use subspace learning to relieve the gap.

4.3. Feature compression using subspace learning

We compress the dimensionality to [16,32,64,128,256,512] separately when using LPP PCA, and to [16,32,64,c-1] separately when using LDA.

Table 1. Comparison of layer performance.

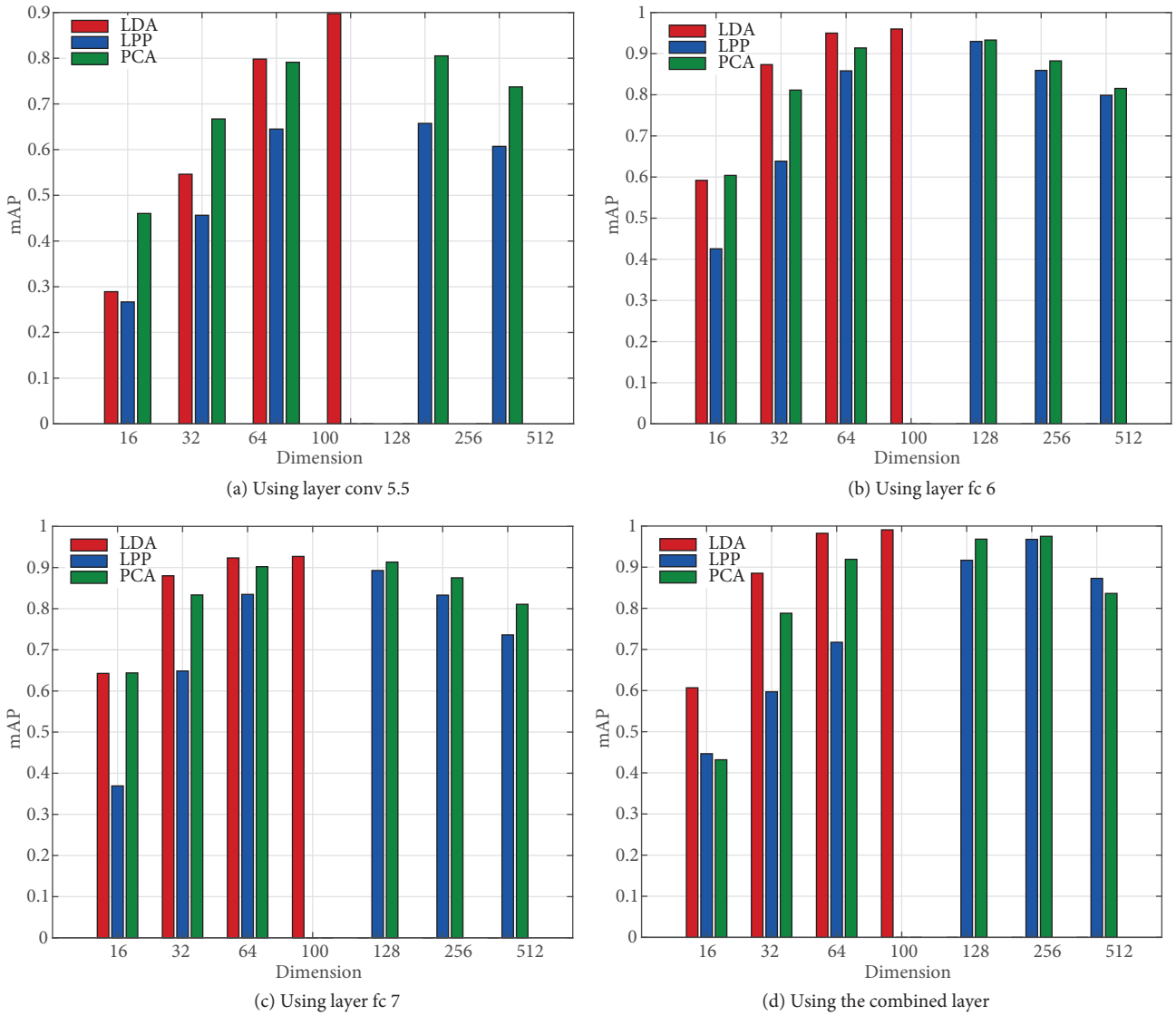
Method	Caltech 101	Stanford dogs	Indoor scene	Dim	Time (s)
$conv5\_1$	0.659	0.351	0.373	100352	10.76
$conv5\_2$	0.724	0.397	0.411	100352	10.76
$conv5\_3$	0.759	0.410	0.396	100352	10.76
$fc6$	0.938	0.765	0.762	4096	0.30
$fc7$	0.954	0.746	0.737	4096	0.30
combined representation	0.970	0.898	0.901	8192	0.59

Figures 3, 4, and 5 show the results of using different dimensionality reduction methods with different types of datasets. We can see that the mAP of LPP and PCA does not always rise with the increase of dimensionality. On the contrary, mAP performs best if the compressed dimensionality is close to the threshold, which is usually around the number categories. Besides, the mAP of LDA keeps increasing until the dimensionality is equal to $c-1$, so it can prove our conclusion as well. Before the encoding length comes close to the number of categories, the performance of image retrieval presents an upward trend. In contrast, away from this threshold, the tendency presents a downward trend.

The parameter α is chosen from the values $\{10^r : r \in \{-5,-4,-3,...3,4,5\}\}$. To discuss the influence of α when set as different values, we show the results of retrieval performance in Figure 6. Then we can get value of α for our experiment.

Table 2. Comparison among different encoding approaches.

Method	Caltech 101	Stanford dogs	Indoor scene	k	Dim	Time (s)
$fc7$	0.954	0.746	0.737	–	4096	0.30
FV	0.980	0.801	0.852	16	4096	0.30
	0.971	0.819	0.857	32	8192	0.59
	0.973	0.846	0.881	64	16384	1.2
	0.965	0.897	0.866	128	32768	2.48
VLAD	0.979	0.880	0.877	16	4096	0.30
	0.980	0.884	0.888	32	8192	0.59
	0.985	0.891	0.897	64	16384	1.2
	0.988	0.902	0.907	128	32768	2.48
SSDH	0.814	0.770	0.898	–	16	0.002
	0.953	0.878	0.905	–	32	0.0025
	0.977	0.894	0.904	–	64	0.0046
	0.975	0.890	0.906	–	128	0.0087
	0.974	0.895	0.912	–	512	0.0352
CFCE	0.610	0.351	0.551	–	16	0.002
	0.884	0.622	0.799	–	32	0.0025
	0.982	0.836	0.916	–	64	0.0046
	0.990	0.907	0.918	–	c−1	–



(a) Using layer conv 5.5

(b) Using layer fc 6

(c) Using layer fc 7

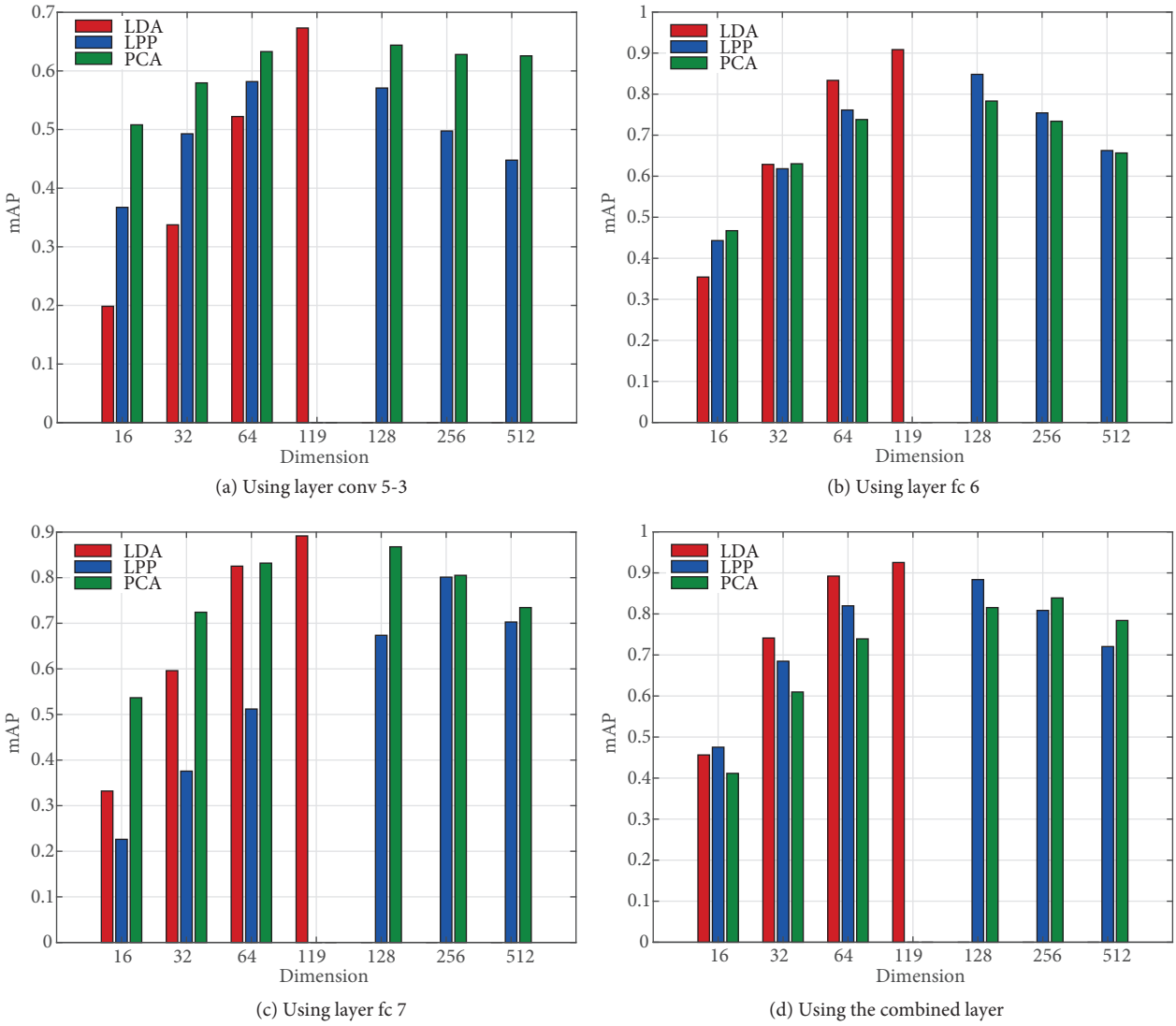
(d) Using the combined layer

Figure 3. In Caltech 101, retrieval performance comparison of three different dimensionality reduction approaches by representation as layer $conv5\_3$, representation as layer $fc6$, representation as layer $fc7$, or representation as combined layer.

4.4. Comparison among different encoding approaches

In this experiment, unified representation is used as a default feature. In order to verify the validity of our encoding method, our model is further compared with different encoding methods including aggregation approaches, neural codes [40], and SSDH [32]. For aggregation approaches, we choose FV [41] and VLAD [28] as typical examples. k is the number of Gaussian mixture models in FV. In VLAD, k is the number of cluster center points. For k, we chose values of [16,32,64,128] separately. Before using an aggregation approach to encode the indexing, we use a dimensionality reduction method to accelerate aggregation speed.

We evaluate the performances of different encoding approaches. The results are shown in Tables 2. Our approach provides arguably the best performance. The CFCE is comparable with the best. Even though all of these approaches perform well, the dimensions of FV and VLAD are relatively high. The result of SSDH is



(a) Using layer conv 5-3

(b) Using layer fc 6

(c) Using layer fc 7

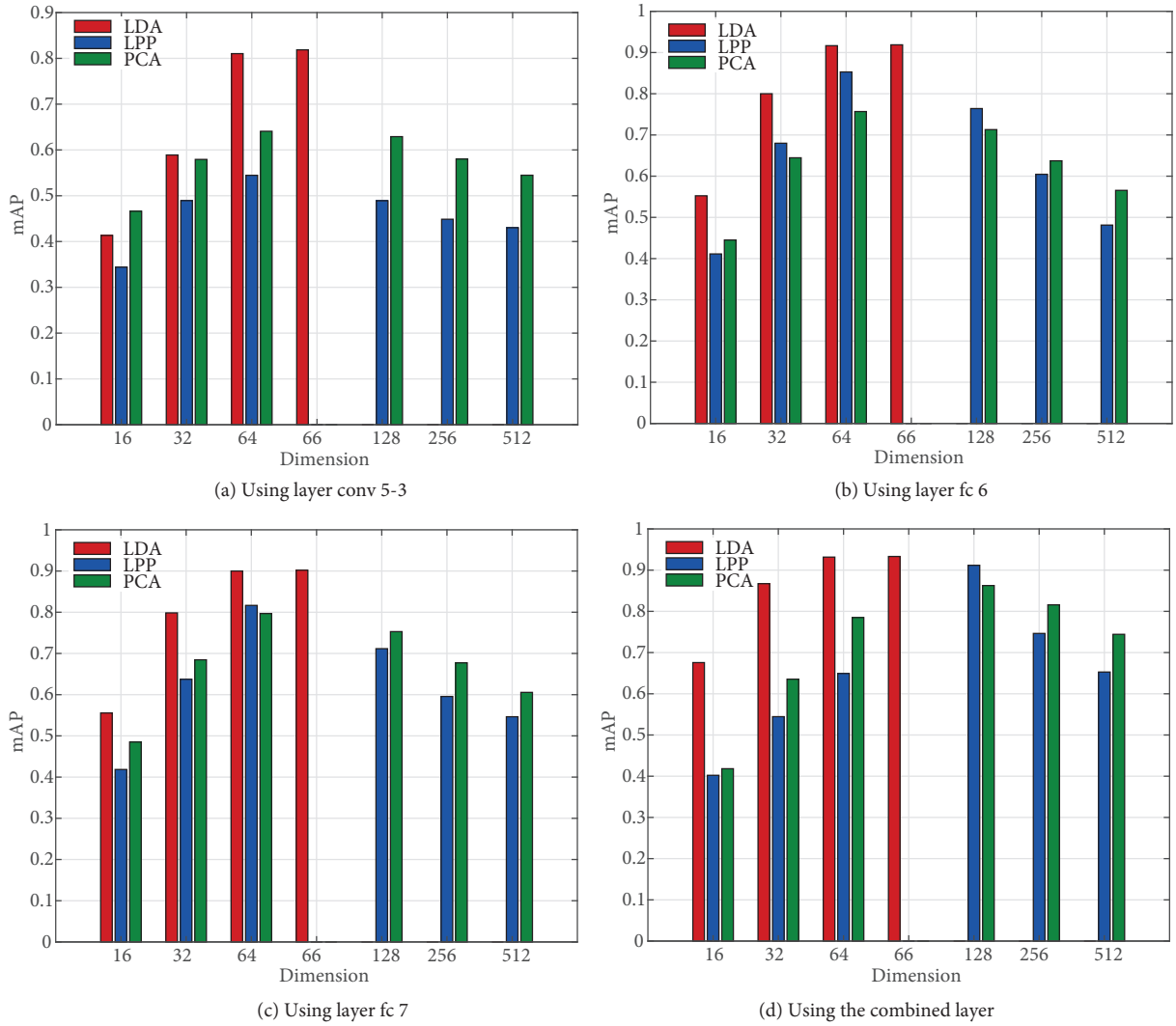
(d) Using the combined layer

Figure 4. In Stanford dogs, retrieval performance comparison of three different dimensionality reduction approaches by representation as layer $conv5\_3$, representation as layer $fc6$, representation as layer $fc7$, or representation as combined layer.

better when encoding bits are relatively shorter. However, the model requires retraining, which is quite time-consuming, once the lengths of encoding have changed. In our approach, however, we only need training one time to obtain a unified representation. For different encoding bits, we only modify the subspace learning as a dimensionality reduction method. Therefore, our approach provides a considerably better trade-off between the model training time, performance, and efficiency of image retrieval. At the same time, the retrieval performance of our approach is comparable to the SSDH method.

4.5. Results

- Our method is suitable for coarse-grained and fine-grained datasets in image retrieval.
- Our experiments prove that the suitable length of encoding bits is close to the number of classes in different dataset types.



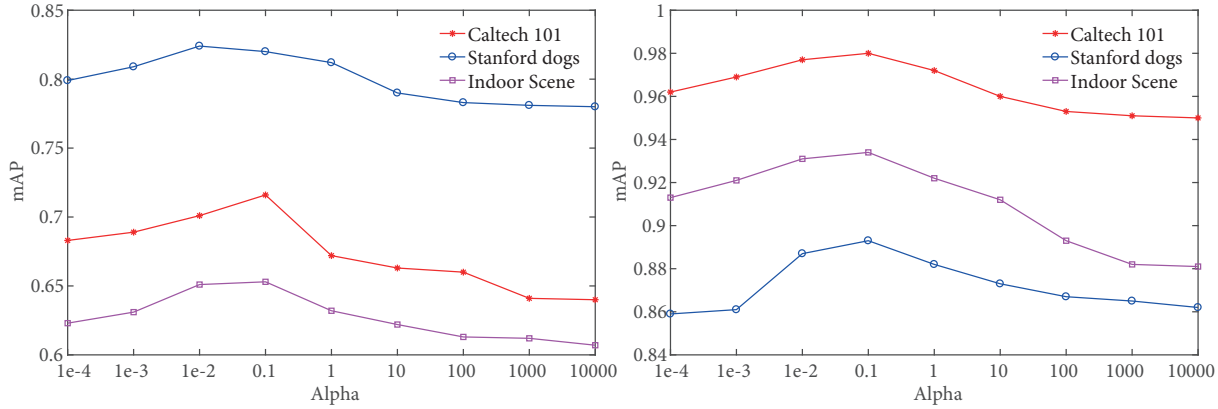
(a) Using layer conv 5-3

(b) Using layer fc 6

(c) Using layer fc 7

(d) Using the combined layer

Figure 5. In Indoor Scene, retrieval performance comparison of three different dimensionality reduction approaches by representation as layer $conv5\_3$, representation as layer $fc6$, representation as layer $fc7$, or by representation as combined layer.



(a) The setting of parameter a using lpp method.

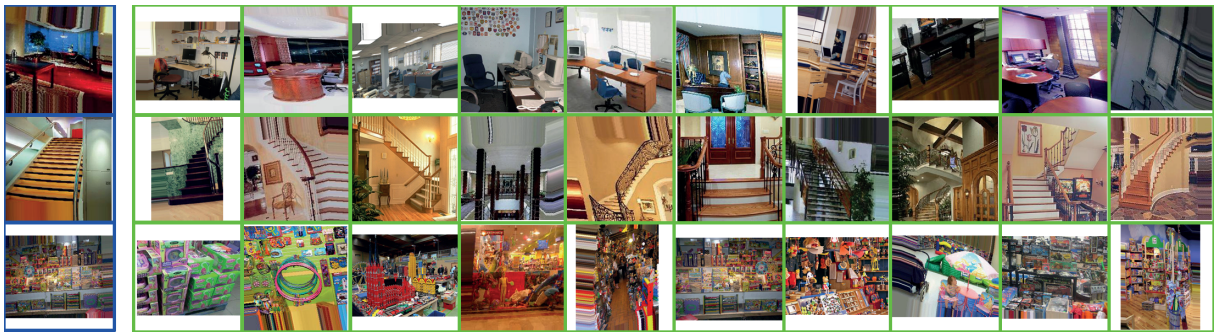
(b) The setting of parameter a using lda method.

Figure 6. The retrieval mAP when parameter α was set as different values in Caltech 101, Stanford dogs, and Indoor Scene datasets using LPP method (a) and LDA method (b).

- Subspace learning can significantly improve the performance of image retrieval and exclude the redundant information.

(a) Caltech 101

(b) Stanford dogs



(c) Indoor Scene recognition

Figure 7. Retrieval examples using CFCE method on cross datasets. Blue color corresponds to inquiry image, red to false results, and green to true results.

Figure 7 shows the top ten retrieval results for three typical examples of classes in the collected dataset. Most of the results are relevant to the inquiry. Therefore, CFCE can provide great performance for cross datasets.

5. Conclusion

We have proposed a new feature compression method suitable to both coarse-grained and fine-grained image retrieval, which provides state-of-the-art performances across datasets. In the process, spectral regression is the

primary candidate to explore the subspace of combined features without noticeably impacting accuracy. One potential drawback is that the retrieval performance will decrease once the length of the code is less than the number of categories. In the future, we will try to solve this problem.

References

[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM 2017; 60 (6): 84-90.

[2] Ren S, He K, Girshick RB. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017; 39 (6): 1137-1149.

[3] Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2013; 35 (1): 221-231.

[4] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2015; 37 (9): 1904-1916.

[5] Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017; 39 (4): 640-651.

[6] Bengio Y, Courville AC, Vincent P. Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 2013; 35 (8): 1798-1828.

[7] Babenko A, Lempitsky VS. Aggregating deep convolutional features for image retrieval. arXiv: Computer Vision and Pattern Recognition, 2015.

[8] Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S. CNN features off-the-shelf: an astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition; 2014; Columbus, OH, USA. New York, NY, USA: IEEE. pp. 806-813.

[9] Zhao F, Huang Y, Wang L. Deep semantic ranking based hashing for multi-label image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition; 2015; Boston, MA, USA. New York, NY, USA: IEEE. pp. 1556-1564.

[10] Bellman R. Dynamic programming. Science 1966; 153 (3731): 34-37.

[11] Azizpour H, Razavian AS, Sullivan J, Maki A, Carlsson S. Factors of transferability for a generic ConvNet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2016; 38 (9): 1790-1802.

[12] Babenko A, Lempitsky V. Aggregating local deep features for image retrieval. In: IEEE International Conference on Computer Vision; 2015; Boston, MA, USA. New York, NY, USA: IEEE. pp. 1269-1277.

[13] Tao R, Gavves E, Snoek CGM. Locality in generic instance search from one example. In: IEEE Conference on Computer Vision and Pattern Recognition; 2014; Columbus, OH, USA. New York, NY, USA: IEEE. pp. 2091-2098.

[14] Yue-Hei Ng J, Yang F, Davis LS. Exploiting local features from deep networks for image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition; 2015; Boston, MA, USA. New York, NY, USA: IEEE. pp. 53-61.

[15] Ke Y, Sukthankar R, Huston L, et al. Efficient near-duplicate detection and sub-image retrieval. In: ACM International Conference on Multimedia; 2004; New York, NY, USA. New York, NY, USA: ACM. p. 5.

[16] Liu L, Fieguth P, Guo Y, Wang X, Pietikäinene M. Local binary features for texture classification: taxonomy and experimental study. Pattern Recognition 2017; 62: 135-160.

[17] Oliva A, Torralba A. Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision 2001; 42 (3): 145-175

[18] Deng Y, Manjunath BS, Kenney CS. An efficient color representation for image retrieval. IEEE Transactions on Image Processing 2001; 10 (1): 140-147.

[19] Xia R, Pan Y, Lai H. Supervised hashing for image retrieval via image representation learning. In: Twenty-Eighth AAAI Conference on Artificial Intelligence; 2014; Quebec City, Canada. pp. 2156-2162.

[20] Yan K, Wang Y, Liang D. CNN vs. sift for image retrieval: alternative or complementary? In: ACM International Conference on Multimedia; 2016; Amsterdam, the Netherlands. New York, NY, USA: ACM. pp. 407-411.

[21] Bengio Y, Courville AC, Vincent P. Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 2013; 35 (8): 1798-1828.

[22] Zhang Q, Nian Wu Y, Zhu SC. Interpretable convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. New York, NY, USA: IEEE. pp. 8827-8836.

[23] Elisha O, Dekel S. Function space analysis of deep learning representation layers. arXiv: Artificial Intelligence, 2017.

[24] Razavian AS, Sullivan J, Carlsson S. Visual instance retrieval with deep convolutional networks. ITE Transactions on Media Technology and Applications 2016; 4 (3): 251-258.

[25] Zheng L, Zhao Y, Wang S. Good practice in CNN feature transfer. arXiv: Computer Vision and Pattern Recognition, 2016.

[26] Zhang Q, Nian WY, Zhu SC. Interpretable convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. New York, NY, USA: IEEE. pp. 8827-8836.

[27] Perronnin F, Liu Y, Sánchez J. Large-scale image retrieval with compressed fisher vectors. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010; San Francesco, CA, USA. New York, NY, USA: IEEE. pp. 3384-3391.

[28] Jégou H, Douze M, Schmid C. Aggregating local descriptors into a compact image representation. In: IEEE Conference on Computer Vision and Pattern Recognition; 2010; San Francesco, CA, USA. New York, NY, USA: IEEE. pp. 3304-3311.

[29] Sivic J, Zisserman A. Video Google: a text retrieval approach to object matching in videos. In: International Conference on Computer Vision; 2003; Nice, France. New York, NY, USA: IEEE. pp. 1470-1477.

[30] Salakhutdinov R, Hinton G. Semantic hashing. International Journal of Approximate Reasoning 2009; 50 (7): 969-978.

[31] Carreira-Perpinán MA, Raziperchikolaei R. Hashing with binary autoencoders. In: IEEE Conference on Computer Vision and Pattern Recognition; 2015; Boston, MA, USA. New York, NY, USA: IEEE. pp. 557-566.

[32] Yang HF, Lin K, Chen CS. Supervised learning of semantics-preserving hash via deep convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018; 40 (2): 437-451.

[33] Cai D, He X, Han J. Spectral regression: a unified approach for sparse subspace learning. In: IEEE International Conference on Data Mining; 2007; Omaha, NE, USA. New York, NY, USA: IEEE. pp. 73-82.

[34] Clevert D, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv: International Conference on Learning Representations, 2016.

[35] Cai D, He X, Han J. Spectral regression: a unified approach for sparse subspace learning. In: IEEE International Conference on Data Mining; 2007; Omaha, NE, USA. New York, NY, USA: IEEE. pp. 73-82.

[36] Fei-Fei L, Fergus R, Perona P. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 2007; 106 (1): 59-70.

[37] Khosla A, Jayadevaprakash N, Yao B. Novel dataset for fine-grained image categorization: Stanford dogs. In: CVPR Workshop on Fine-Grained Visual Categorization; 2011; Colorado Springs, CO, USA: . New York, NY, USA: IEEE.

[38] Quattoni A, Torralba A. Recognizing indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition; 2009; Miami Beach, FL, USA. New York, NY, USA: IEEE. pp. 413-420.

[39] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: International Conference on Learning Representations, 2015.

[40] Babenko A, Slesarev A, Chigorin A. Neural codes for image retrieval. In: European Conference on Computer Vision; 2014; Zurich, Switzerland. Berlin, Germany: Springer. pp. 584-599.

[41] Perronnin F, Dance C. Fisher kernels on visual vocabularies for image categorization. In: IEEE Conference on Computer Vision and Pattern Recognition; 2007; Minneapolis, MN, USA. New York, NY, USA: IEEE. pp. 1-8.