A novel feature representation: Aggregating convolution kernels for image retrieval

WANG, Qi; Lai, Jinxing; CLAESEN, Luc; Yang, Zhenguo; Lei, Liang & Liu, Wenyin (2020) A novel feature representation: Aggregating convolution kernels for image retrieval. In: Neural Networks, 130 (2020) , p. 1 -10.

DOI: 10.1016/j.neunet.2020.06.010
Handle: http://hdl.handle.net/1942/31322

A Novel Feature Representation: Aggregating Convolution Kernels for Image Retrieval

Qi Wang[a,b], Jinxing Lai[a], Luc Claesen[b], Zhenguo Yang[a,*], Liang Lei[a], Wenyin Liu[a,*]

[a] *Guangdong University of Technology, Guangzhou 510006, China*
[b] *Hasselt University, Martelarenlaan 42, Hasselt 3500, Belgium*

Abstract

Activated hidden unites in convolutional neural networks (CNNs), known as feature maps, dominate image representation, which is compact and discriminative. For ultra-large data sets, high dimensional feature maps in float format not only result in high computational complexity, but also occupy massive memory space. To this end, a new image representation by aggregating convolution kernels (ACK) is proposed, where some convolution kernels capturing certain patterns are activated. The top-n index numbers of the convolution kernels are extracted directly as image representation in discrete integer values, which rebuild relationship between convolution kernels and image. Furthermore, a distance measurement is defined from the perspective of ordered sets to calculate position-sensitive similarities between image representations. Extensive experiments conducted on Oxford Buildings, Paris, and Holidays, etc., manifest that the proposed ACK achieves competitive performance on image retrieval with much lower computational cost, outperforming the ones using feature maps for image representation.

Keywords: Image Representation; Feature Aggregating; Distance Measurement; Image Retrieval

1. Introduction

Automatic extraction, analysis and understanding of images are the aims of computer vision, which usually can be conducted in two stages, i.e., image representation, and pattern analysis (Radenovic et al., 2019; Yang et al., 2019). The first stage aims to obtain image representations in vector forms conveying discriminative information of images. Furthermore, pattern analysis methods for certain tasks, such as classification, detection, segmentation, etc., can be developed respectively. Deep neural networks unify the two stages jointly, which adapts image representation for specific tasks (Guo et al., 2019; Wang et al., 2019b; Zhan and Lu, 2019; Tian et al., 2020).

In terms of image representation, early methods index images by visual cues to extract global descriptors, such as texture, and color. However, these global descriptors cannot deal with the variations of images, such as illumination, translation, occlusion, and truncation. These variations compromise the retrieval accuracy and limit the applications of global descriptors. Bag-of-Words (BoW) model is proposed for image representation (Sivic and Zisserman, 2003) and image classification (Csurka et al., 2004; Jégou et al., 2010), relying on the scale-invariant feature transform (SIFT) descriptor (Lowe, 2004). The seminal work using deep learning is proposed by Krizhevsky et al. (Krizhevsky et al., 2012), where AlexNet achieves the state-of-the-art recognition accuracy on ILSRVC 2012. Furthermore, deep learning based methods (Ng et al., 2015; Wu et al., 2019; Roy and Boddeti, 2019; Liu et al., 2018), especially the convolutional neural network (CNN) dominates the area of image representation learning. Though convolution and fully connection layers show strong power for image representation (Bhat, 2017; Wang et al., 2018; Cheng et al., 2018; Vo et al., 2019), still suffering from some disadvantages. For instance, CNN-based image representation is in high dimension and float form, resulting in large memory consumption and computational cost for large-scale data.

Convolution preserves the spatial relationship between pixels by learning image features through using traversing the input data, while convolution kernels in different CNNs can extract different image features (Zheng et al., 2018; Koh and Liang, 2017; ElAdel et al., 2017). Convolution kernel is a filter in the process of data feed forward, continuously filtering out the information that does not match the current convolution kernel and purifying the data, and finally obtaining the feature descriptors of the image through layer convolution. Intuitively, we visualize the three convolution kernels of VGG-16 in Fig. 1 according to maximum activation of feature maps (Erhan et al., 2009; Szegedy et al., 2014; Zeiler and Fergus, 2014). We can observe that the kernel in subfigure (b) can extract the columnar structure features of the image in the yel-

*Corresponding authors
Email addresses: wangqi_6414@sina.com (Qi Wang), 1048703768@qq.com (Jinxing Lai), luc.claesen@uhasselt.be (Luc Claesen), zhengyang5-c@my.cityu.edu.hk (Zhenguo Yang), leiliang@gdut.edu.com (Liang Lei), liuwy@gdut.edu.cn (Wenyin Liu)

low rectangles; the kernel in subfigure (c) can extract the triangular structural features in the black rectangles; the kernel in subfigure (d) can extract the arched door features of the image in red rectangles. Intuitively, each individual convolution kernel responses to some certain features in an image.

In this paper, we propose aggregating convolution kernels (ACK), brightening convolution kernels of a layer and ranks the kernels based on their response intensity, which achieve a new image representation. The brightening operation in ACK maximizes the convolution kernel response to image intrinsic features. Furthermore, the top-n index numbers of the convolution kernels constitute an index sequence as the image representation, which indicates their convolution response intensity. Compared with the vector representations in float forms, ACK extracts semantic features with low computational complexity. To measure the distance between image representations achieved by ACK, we propose a similarity measurement by taking into account the overlap between the index numbers and their positions of the discrete numbers. The contributions of this paper are summarized as follows.

1. We propose to aggregate convolution kernels (ACK) by brightening on each kernel and selecting the ones with maximum response to image intrinsic features, achieving a new image representation consisting of kernel index numbers.
2. We design a position-sensitive similarity measurement for image representations in integer values achieved by ACK, taking into account the aspects of overlap and positions of the convolution kernels.
3. We conduct extensive experiments on public datasets, demonstrating the effectiveness and efficiency of ACK for image retrieval.

The paper is structured as follows. In Section 2, we discuss the related work on image representation and distance measurement. Section 3 presents the proposed ACK method. Section 4 introduces the metric method for the image representation. Section 5 shows the experimental results. Section 6 concludes the paper.

2. Related work

In this section, we investigate some representative works on image representation and distance measurement. For image representation, we divide the works into two categories (Zheng et al., 2018), including SIFT-based representation, and CNN-based representation.

2.1. SIFT-based image representation

Image representation is a fundamental issue in computer vision. Since Lowe et al. (Lowe, 2004) propose SIFT algorithm considering scale invariance of image features, many image representation methods (Jégou et al., 2010; Philbin et al., 2007; Nistér and Stewénius, 2006) have been proposed, which usually focus on small-scale data sets. For instance, Rosten et al. (Rosten and Drummond, 2006) apply FAST algorithm to extract the main direction of feature points. Calonder et al. (Calonder et al., 2010) introduce BRIEF algorithm, which output binary vectors to simplify the distance measurement. However, feature points for image representation are computational-extensive, which is considered to be undesirable in real life applications (Zheng et al., 2018).

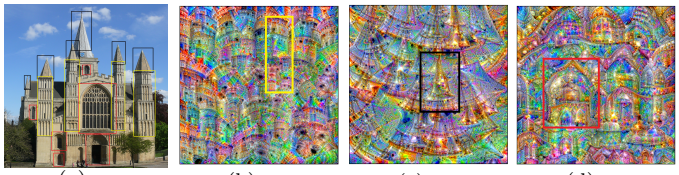


Fig. 1. (a) is the original image. (b), (c), (d) are the visualization of convolution kernels, whose index number is 199, 284, 368 respectively on conv5-3 layer of pretrained VGG-16 in ImageNet. The rectangles of different colors in (a) correspond to the visual features in b, c and d respectively. Note that the kernels are learned on all the images from ImageNet, which is independent with the input image from Oxford dataset.

2.2. CNN-based image representation

Recently, CNN-based image representation shows impressive performance on computer vision tasks, which can be roughly divided into five categories.

- **Fully-connected representation**, is generated after layers of convolutions with the input image, which has a global receptive field. This representation has been applied in many areas of computer vision, such as probability statistics with the last fully connection layer for classification problems (Geirhos et al., 2018; Liu et al., 2019; Sarigul et al., 2019), linear regression with the last fully connection layer for image detection (Bhat, 2017), spatial continuation with the fully connection layer for cross-modal problems (Pang et al., 2019), etc. It has strong global expression ability and contains high-level semantic information.

- **Convolutional representation**, applies the activations of convolutional layers followed by a global-pooling operation. A compact image representation is constructed in this fashion with dimensionality equivalent to the number of feature maps of the corresponding convolutional layer. Pooling strategies usually are utilized to further compress and analyze the convolutional layer. For instance, a number of approaches use pooling strategies to obtain image representation for content-based image retrieval (Zheng et al., 2018).

- **Codebook representation**, constructs codebook with the feature maps of fully-connection layer, and obtains image representation in a data reconstruction manner. For instance, (Jégou et al., 2010) proposes vector of locally aggregated descriptor (VLAD) method,

which aggregates local descriptors into a compact image representation. (Perronnin et al., 2010) proposes fisher vector (FV) approach which combines the benefits of generative and discriminative approaches.

- **Hash representation**, obtains compact binary codes for image representation. For instance, Bai et al. (Bai et al., 2017) present a recursive hashing scheme based on progressively expanded salient regions for hash representation. Calonder et al. (Calonder et al., 2010) propose to use binary strings as an efficient feature point, which show a highly discriminative. However, for some fine-grained tasks, the classes is relatively close, and the performance is discount, especially on some landmark data (Wu et al., 2019).

- **Mixed representation**, features being from multiple view and different dimension semantic information, are superimposed together to make a final representation for certain tasks, which is employed for complex real-world scenarios (Zhu et al., 2019; Chen and Deng, 2019; Zhou and Gu, 2020; Wang et al., 2019a). For example, Wang et al. (Wang et al., 2018) represent the image with the feature vector by multi-feature fusion and feature aggregation. Cheng et al, (Cheng et al., 2018) introduce a static object images representation for videos, which jointly imitate under the similarity network with reconfigurable deep tree structure.

In summary, the aforementioned methods achieve image representation in float data form, which occupy high memory and computational cost. Though hash methods use hamming distance and binary representation of image, they usually sacrifice performance on certain tasks. In this paper, we propose an efficient image representation, which can achieve the same accuracy with low computational and memory cost on image retrieval task.

2.3. Distance measurement

For data in continuous or discrete values, different distance measurements need to be used. For instance, Euclidean Distance, Manhattan Distance, Chebyshev Distance, Correlation coefficient, Mahalanobis Distance and Cosine distance, etc., are common metrics for continuous data. For discrete data, Hamming distance, Jaccard similarity, Dynamic Time Warp can be used. More specifically, Euclidean distance and cosine distance are commonly utilized in image retrieval to obtain the similarity between two image vectors (Gordo et al., 2017). Hash representation methods adopt Hamming distance to calculate the similarity of between binary image representations (Calonder et al., 2010; Wu et al., 2019). Jaccard similarity coefficient (Capra, 2005) calculates the distance between sets. Dynamic Time Warp (Keogh and Pazzani, 2001; Claure et al., 2018) measures the similarity between time series, mainly used in the field of speech recognition. Compared to these paper (Sun et al., 2020, 2019; Sun et al., 2019), this calculation method is similar to some extent. However, the metrics for discrete data neglects the order of the elements in the sets, while the position of matched elements may be critical to measure the similarity between representation in discrete values.

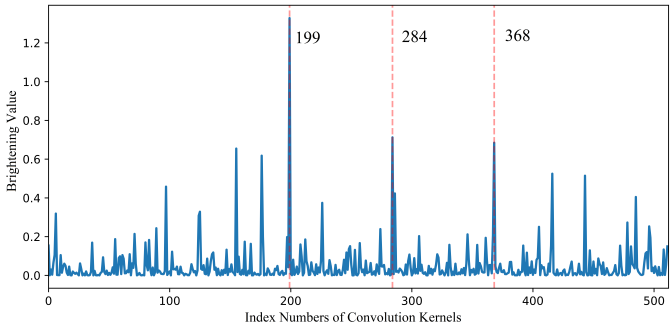
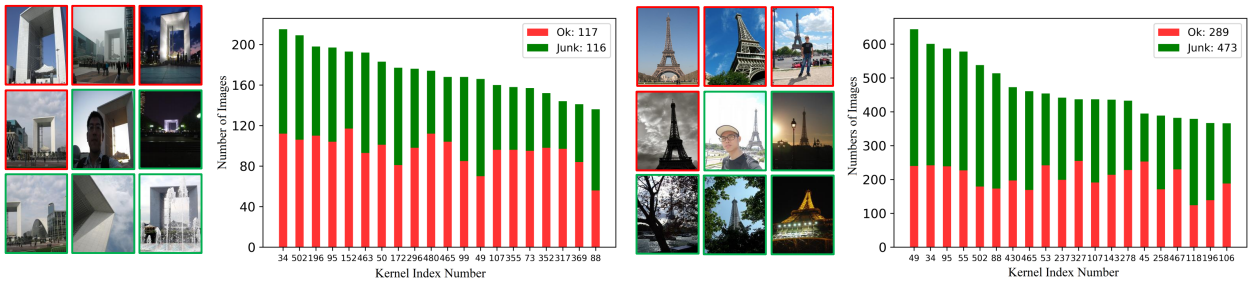


Fig. 2. Brightening statistics for Fig. 1 (a) on Conv-5-3 of pretrained model VGG-16 in ImageNet.

3. Aggregate convolution kernels (ACK) for image representation

In the convolutional neural networks, CNN can be seemed as a multi-stage distillation of information, in which information is continuously filtered and purified (Springenberg et al., 2015; Zheng et al., 2018) by convolution kernels. Each convolution kernel can be understood as a feature template. For example, Fig. 1 (b), (c) and (d) can be seen as the main inner feature of (a) in the vision. Generally speaking, a CNN model achieving competitive performance usually is with relatively stable convolution kernels, so as to extract image features gradually. Intuitively, the convolution kernel gradually transits from low-level features (e.g., color, texture, etc.) to high-level semantic features. To some extent, CNN convolution kernel is equivalent to a feature extractor, which activates the units corresponding to certain patterns.

Therefore, we propose to aggregate convolution kernels (ACK) as image representation in **Algorithm 1**. More specifically, we use brightening to activate the convolution kernels with maximum response to image intrinsic feature, and select the convolution kernels according to the brightness intensity. The ordered index numbers of the convolution kernels can be regarded as a new image representation and different from the vector forms. More specifically, an input image can be fed into the network models. For a certain layer, we perform feature extraction through forward operation, and obtain the target layer. Furthermore, we calculate the brightness intensity for convolution kernels, which can be obtained by using global pooling method, such as the max-pooling convolutional method (MAC) (Razavian et al., 2014), the regional maximum activation of convolutions method (RMAC) (Tolias et al., 2016), the cross-dimensional weighting for aggregating deep convolutional features method (CroW) (Kalantidis et al., 2016), the sum-pooling convolutional method



(a) The statistics on 233 Paris_Defense images

(b) The statistics on 762 Paris_Eiffel_Tower images

Fig. 3. Statistics of the top-20 ranked kernels on the images in the categories of "Paris_Defens" and "Paris_Eiffel_Towe". The left is some examples, and the right is statistical results. OK indicates easy examples, while Junk indicates hard examples. Easy examples: More than 25% of the object is clearly visible. Hard examples: Less than 25% of the object is visible, or there are very high levels of occlusion or distortion.

(SPOC) (Babenko and Lempitsky, 2015), Gem (Radenovic et al., 2019), etc. Finally, the rank list of the kernels is taken as the feature representation of the image.

Algorithm 1 Aggregating Convolution Kernels Method

1: Selecting one convolutional layer X in the trained CNN model, and then obtain the total channel number n of the selected X. The selected CNN model can be a network that trained from scratch, or be the pre-trained model with ImageNet.
2: Input image, and perform feature extraction with the selected CNN.
3: Brightening: First, calculating the response result of X's each convolution kernels by the global pooling method, called brightness values of kernels. Second, the brightness values of X's each convolution kernels are sorted from large to small. Finally, obtaining rank list of the convolution kernels' index.
4: Adopting the top-n as the image representation.

Intuitively, we visualize some examples to show the rationale of brightening and ACK. In terms of brightening, we take the image sample in Fig. 1 as an example, and select the Conv5-3 of the VGG-16 network for brightening statistics. As shown in Fig. 2, we calculate the response of brightening for each of the 512 kernels. The top-3 kernel numbers are 199, 284, 368, which are visualized in Figs 1(b), 1(c), and 1(d), respectively. Different convolution kernels may focus on extracting certain patterns in images. Therefore, we use the response intensity information of each convolution layer to different kernels as image representations.

Furthermore, for verifying effectiveness, we conduct statistics of the top-20 ranked kernels by ACK method on the images in two categories from the Paris building data set, as shown in Fig. 3. We conclude three observations. 1). For the same category, by observing index number Figs. 3(a) or 3(b), the top-20 convolution kernels in a total number of 512 kernels are quite widespread, and most index numbers of images in this category are relatively concentrated, which means that images of the same category have most similar intrinsic kernel features. 2). For different categories by comparing Figs. 3(a) and 3(b), most of the top ranked kernels are quite different, benefiting to the discrimination of images categories, which means that images of the different categories have their own unique internal characteristics. This result exactly shows our index sequence has ability to distinguish details. 3). We have similar observations on both easy and hard examples in Figs. 3(a) or 3(b), having an approximate distribution on statistics results, which illustrates that our ACK method can obtain a robust representation.

4. Position-sensitive distance measurement for ACK-like image representation

4.1. Method

The image representation achieved by ACK is a set of discrete and ordered index numbers for each image, i.e., there is no distance relation between the index numbers. Therefore, vector-based metrics, such as cosine distance, cannot be used directly. In addition, the index numbers in the representation are position-sensitive, making the metrics like Jaccard similarity cannot be adopted as well. To this end, we propose a distance measurement according to dissimilarity in different convolution kernel extractors in Section 3. We construct a position-to-position weight distribution, named pseudo-Gaussian distribution matrix, which is similar to quarter of 4-dim Gaussian distribution in Fig. 4, and calculate the score between two elements in **Algorithm 2**. The similarity of two ACK representations A and B is defined as discrete ordered sequences distance (denoted as $dosd(A, B)$). Then we add up all the position scores of the two vector elements. Finally, for convenient comparisons, the final result $dosd(A, B)$ is normalized to $[0, 1]$. In theory, the time complexity of **Algorithm 2** is $O(n^2)$.

4.2. Details

(1) For two n-dimensionally ordered index sequences for image representation A and B, if there exist the same elements in two sequences, the difference of the elements at positions between A and B should be inversely proportional to the metric. In other words, the smaller $abs(i - j)$ is,

the higher contribution of this elements is. Therefore, we construct a symmetric matrix K, i.e. $k_{i,j} = 1 - \frac{|i-j|}{n}$. In practice, for the two ACK representation A and B, if the position of the same convolution kernel is closer, it means the images A and B are higher similar.

(2) For a single representation index sequence of images, the anterior index sequence is more important than posterior sequence, so the front elements of index sequence should be higher weight. Therefore, we construct a weighted vectors $w_i = 1 - \frac{i}{n} = \frac{n-i}{n}$. In practice, for the index sequence of image representation, the front convolution kernels means they are more able to express their image characteristics. For example, the top convolution kernels usually extract some of the appearance characteristics of the buildings in Paris building dataset. In Fig. 1, a cylindrical convolutional kernel extractor only extracts the cylindrical characteristics of a building, rather than extracting some animal features.

(3) For the contribution score s, we define $s = (n - |i - j|) \cdot G(i, j)$, which means if the position of the corresponding convolution kernel extractor between images A and B is closer, the contribution score should be higher for this position. Theoretically, if the two convolution kernels are the same, i.e. $i = j$, the fraction is given by $s = (n - abs(i - j)) \cdot G(i, j) = n \cdot wi = n - i$. In practice, the position difference of the same convolution kernel is smaller, which indicates that the images A and B are more similar.

4.3. Discussions

We visualize the pseudo-Gaussian distribution matrix G, taking the maximum channel number $c = 512$ as an example in Fig. 4. It synthesizes the ideas ideas of the above two matrices, which can describe the response of different locations to the index sequence for image representation. The matrix distribution G affects the similarity between sequences, which can guide the weights for the same convolution kernel extractor between two images and further obtain the distance of two sequences.

We propose a measurement for discrete ordered sequences according to the characteristics of the convolution extractor, which can give an objective measurement of the two proposed image representations. This metric consists of two aspects. (1) The front convolution kernel sequences mean that current image mainly include this features, which can be used as a key representation. (2) The position of the same convolution kernel between two image representations is positively related to its similarity, i.e. a close position means a high similarity. We use the pseudo-Gaussian distribution matrix to describe the above two objective facts, and we can effectively calculate the similarity of two discrete random sequences by this method.

5. Experiments

In this section, we evaluate the proposed ACK method and the distance measurement on image retrieval tasks.

Algorithm 2 Discrete Ordered Sequences Distance Metric

Input: The channel numbers of selected convolutional layer is c (normally, $c = 2^k$), two images A and B, and their feature representation dimension is n.

Output: calculating $dosd(A, B)$

Apply c for natural number coding, obtaining an ordered set $X = \{0, 1, 2, \cdots, c-1\}$

Obtaining two images representation $A^{1\times n}$ and $B^{1\times n}$ by using the ACK method:

$$A = \{a_0, a_1, \cdots, a_{n-1}\}, a_n \in X$$
$$B = \{b_0, b_1, \cdots, b_{n-1}\}, b_n \in X$$

Construct a symmetric matrix $K^{n\times n}$ and a weight sequence $W^{1\times n}$, which are given by

$$k_{i,j} = 1 - \frac{|i-j|}{n}, w_i = 1 - \frac{i}{n}$$
$$i, j = (0, 1, 2, \cdots, n-1)$$

Calculate the weighted pseudo-Gaussian distribution matrix $G^{n\times n}$,

$$g_{i,j} = k_{i,j} \cdot w_i = (1 - \frac{|i-j|}{n}) \cdot \frac{n-i}{n}$$

$Sum = 0$

for $i \leftarrow 0$ to $n-1$ **do**
  for $j \leftarrow 0$ **to** $n-1$ **do**
    if $a_i == b_j$ **then**
      Contribution score $s = (n - |i-j|) \cdot G(i, j)$
      $Sum = Sum + s$
    else
      Contribution score $s = 0$
      $Sum = Sum + s$
    end if
  end for
end for

Calculate the maximum of the $distance(A, B)$, is given by $Sum_{max} = n + (n-1) + \ldots + 1 = n(n+1)/2$

Normalized the Sum, then obtaining the result $dosd(A, B) = Sum/Sum_{max}$, is simplified by

$$dosd(A, B) = \frac{\sum_{i=0, \exists a_i = b_j}^{n-1} 2w_i \cdot k_{i,j}^2}{n+1}$$
$$i, j = (0, 1, 2, \cdots, n-1)$$

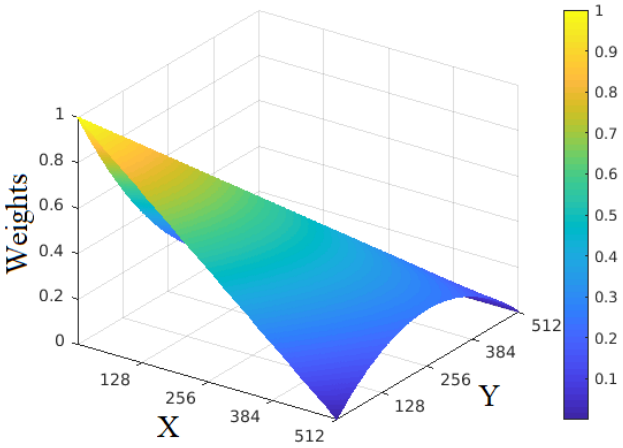


Fig. 4. The visualization of pseudo-Gaussian distribution matrix.

5.1. Datasets and performance metric

The evaluations are conducted on a number of public datasets, including Oxford dataset (Philbin et al., 2007), Paris dataset (Philbin et al., 2008), Holidays dataset (Jégou et al., 2008). These datasets are also combined with 100k distractors from Oxford100k, which can compose a larger scale for broader experiments effectiveness, e.g., the Oxford105k, the Paris106K and the Holidays101K. In terms of performance metric, mean average precision (MAP) is adopted. Following the standard evaluation protocol for Oxford and Paris, we crop the query images with the provided bounding box for multiple perspectives and magnitude, which is fed as input to the CNN.

5.2. Network settings and pooling method

We select the popular networks as our extracted models, including VGG-16 (Simonyan and Zisserman, 2015), Resnet-50 (He et al., 2016), Resnet-101, etc.. Each model comprises the network weights with ImageNet pre-training, and also includes the fine-tuned network weights on three retrieval datasets. We adopt the state-of-the-art fine-tuned method (Radenovic et al., 2019). The following will directly use these pre-training models. In addition, Oxford buildings (Philbin et al., 2007) and Paris datasets (Philbin et al., 2008), are similar to our training data, while the last dataset is quite different because of the containing similar scenes and man-made objects.

Our proposed ACK method is based on pooling, so we select five pooling methods for comparisons.

- The sum-pooling convolutional, SPOC (Babenko and Lempitsky, 2015), calculates the mean value of the feature map on each channel of the convolutional layer to obtain the descriptors as the global representation.
- The max-pooling convolutional, MAC (Razavian et al., 2014), calculates the maximum value of each intermediate feature map and concatenates all these values within a convolutional layer.
- The regional maximum activation of convolutions method, RMAC (Tolias et al., 2016), applies a group of the sliding windows on the feature map. The max values of the sliding windows are calculated and averaged to capture the local details as the local representation.
- The cross-dimensional weighting for aggregating deep convolutional features method, CroW (Kalantidis et al., 2016), is constructed by the spatial weight and the channel weight on the feature maps, which can increase the weight of the region of interest and reduce the weight of the non-object region.
- The generalized-mean pooling method, Gem (Radenovic et al., 2019), adds a trainable pooling parameter to further optimize and generalizes max and average pooling by generalized mean.

5.3. Retrieval task description

For retrieval tasks, there are some additional techniques to improve the final accuracy of the vector methods, such as whitening (Babenko and Lempitsky, 2015; Tolias et al., 2016), extending query (Chum et al., 2011), and derived principal component analysis (PCA) whitening (Jégou and Chum, 2012), trainable whitening (Gordo et al., 2016), average query expansion (a-AQE) (Gordo et al., 2017; Radenovic et al., 2019) and so on. However, these methods are all designed for the image representation methods in vector forms. For fairness, all the following experiments do not consider the additional techniques as mentioned above, in order to compare the feature representation ability of the image descriptors. Since whitening and extending query methods have not been taken into account, the results of the same approach reproduced by us are a little bit lower than those published in the corresponding papers.

5.4. Hardware consumption performance

We compare hardware consumption, data memory usage, and overall query time between two methods. The fine-tuned VGG-16 network is used to conduct tests on Oxford5K and Oxford105K, and all the tested images are scaled to 224*224. The feature extraction of vector method is adopted by the Gem pooling directly, and the ACK method is based on the improved Gem with brightening method. Besides, the activation of last convolutional layer is selected as the feature. After feature extraction from query image and library dataset, the features are saved into .TXT file, and the size of the .TXT file is used to represent the memory occupation. Our online query environment is: Window10, 64bit, CPU: Intel(R) Core(TM) i7-9750H, GPU: NVIDIA GeForce GTX 1660 Ti, 16G RAM.

5.4.1. Memory resources consumption of data

According to the results in Table 1, with the same level of the retrieval accuracy, the memory footprint of our ACK method is much lower than the image representation methods in vector forms. Being able to import the library

Table 1: Performance comparison of two representations under the same conditions on Oxford dataset. Vector is the Gem pooling Radenovic et al. (2019), Ours is the ACK method.

	Oxford5K					Oxford105K				
Method	Vector	Ours				Vector	Ours			
Dim	512	64	128	256	512	512	64	128	256	512
Data Type	float	int	int	int	int	float	int	int	int	int
Memory	64.8Mb	1.6Mb	3.1Mb	6.2Mb	12.4Mb	1345.7Mb	32.7Mb	64.2Mb	128.6Mb	257.7Mb
Time/Image	0.77ms	0.06ms	0.19ms	0.81ms	2.89ms	0.73ms	0.06ms	0.20ms	0.82ms	2.86ms
MAP	64.75	55.86	62.56	**64.96**	64.27	57.98	48.46	54.79	**58.78**	57.02

dataset into random access memory (RAM) at once would greatly reduce the online query time, since it does not have to read the library data from hardware memory. In practice, our method would greatly reduce the online query time on a large-scale datasets, because we could import the library dataset into RAM at one time because of the low memory consumption, rather than read the library data from hardware memory for each inquire.

5.4.2. Query time consumption

Theoretically, the time of data query is linear with the input dimension for a sufficiently large input, so dimensions of the vector are almost proportional to the query time of the image, i.e. the high dimensions need more time for querying. However, as can be seen from Table 1, in the same hardware environment, the time consumed by the 512-dimensional vector method is not much different from that of the 256-dimensional sequence method. This is because in the current python development package, the developers have accelerated the matrix operation, while currently there is no acceleration for the operations of our ordered set. In addition, we can clearly see that when the data volume expands, the memory consumption and query time of vector method are larger than that of the sequence method with the same precision. It means that facing the practical application of computer vision, the index sequence for image representation will have higher accuracy and less query time than the image representation methods in vector forms.

5.4.3. Discussions

This section compares the memory occupation and query time required by the vector forms and the sequence forms. Our proposed method has better feature representation ability, when facing ultra-large data sets. Compared with the image representation methods in vector forms, our method has lower dimensions, less memory consumption and low computational complexity forwhile achieving the same accuracy on image retrieval task. In actual computer vision tasks, these are inevitable problems especially in the era of big data.

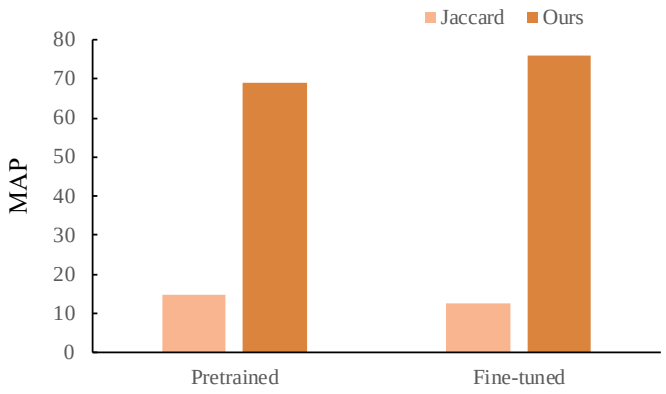


Fig. 5. Performance of two metrics to calculate distance of two index sequences for image representation on Paris dataset.

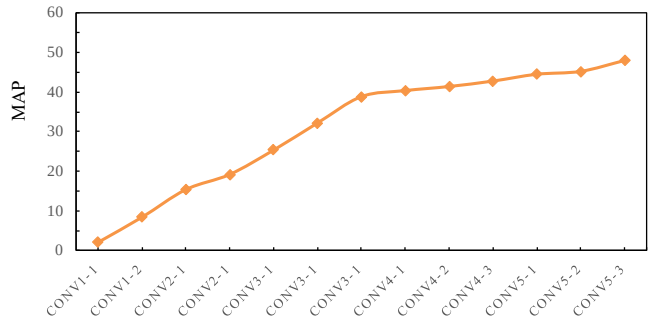


Fig. 6. Performance of different convolutional layer of VGG-16 pre-train model in ImageNet on Paris dataset.

5.5. Metric method for comparison

We compare the proposed metric with Jaccard similarity on Paris dataset in Fig. 5. We use the pre-trained ResNet-50 on ImageNet and the fine-tuned ResNet-50 model for the experiment and the sequence features are obtained by the variant Gem method with brightening. The proposed measurement method is much higher than Jaccard similarity, because every element of disordered sequences is equally important in Jaccard similarity, which means it is not position-sensitive.

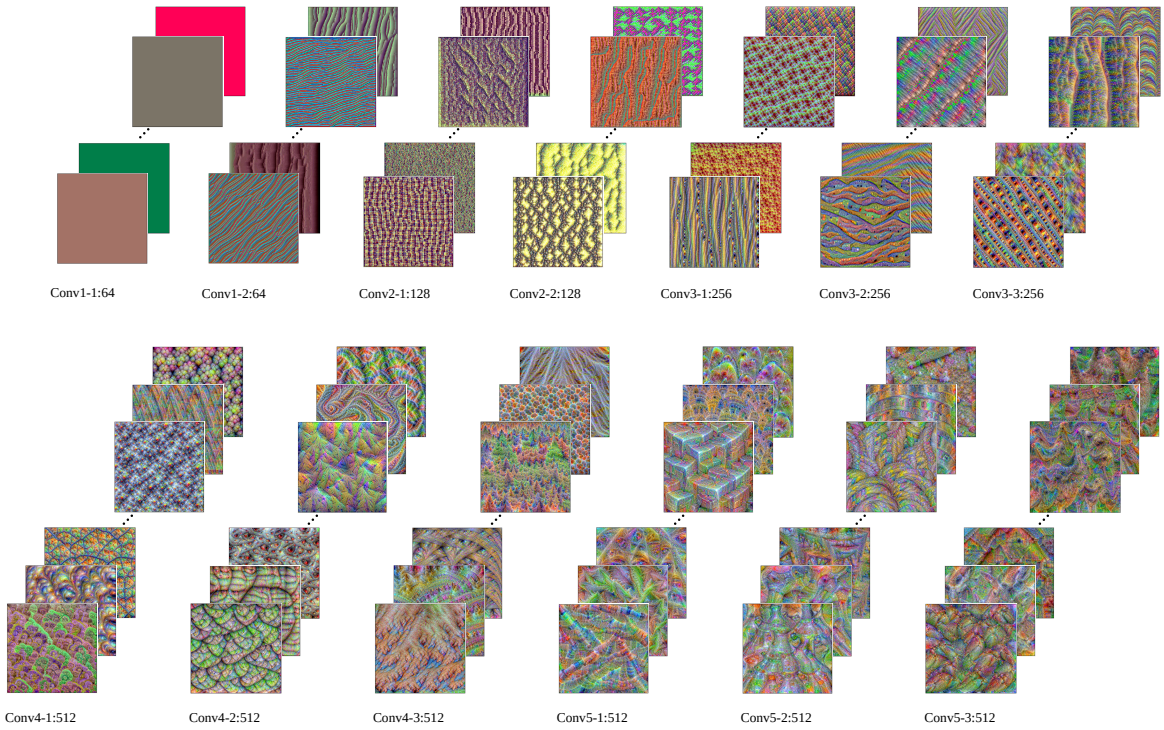


Fig. 7. The visualization of convolution kernels for VGG-16 pretained model on ImageNet based on maximum activation method (Erhan et al., 2009).

5.6. Performance of different feature maps

To verify the effectiveness of the image representation of ACK method, we compare different convolution feature maps by the VGG-16 pre-trained model on ImageNet and on Oxford datasets in Fig. 6. As the number of layers in the convolutional layer increases, the accuracy of the dataset becomes higher, which explains that high-level features have strong representation.

For further exploring the kernel, we visualize all convolution kernels of the VGG-16 model in Fig. 7. We visualize intermediate convnet outputs, which can be obtained through maximizing the mean activation (Erhan et al., 2009), to illustrate the function of the convolution kernel. For one layer, the same index sequence number indicates common features, while different index sequence number demonstrates different details. For example, for cats and dogs, they have the same animal characteristics, but the details are different.

The visualization shows different shapes and semantic for specific objects, such as a bird image, and the core features may be represented by a bird's head, e.g. first feature map in Fig. 7 Conv5-3-512 can be as a bird image representation. As shown in Fig. 7, from which we have two observations. (1) As the layers going deeper, the output of convolution kernel becomes more abstract and semantic, which matches the result of Fig. 6. (2) Each convolution kernel is almost different, which can be regarded as the basic feature units. Therefore, for image representation, it can be considered as a combination of these basic units, which is also the principle of our ACK method.

5.7. Dimension analysis

To compare the representation ability of the image representation methods in vector forms and ACK method, we test the ability of index sequences representation in different dimensions.

5.7.1. Dimension comparison between two image representation

As shown in Table 2, we use the pre-trained Resnet-50 model on ImageNet and the fine-tuned Resnet-50 model, and the adopted datasets are Holidays6K and Holidays106K. The last convolutional layer is selected as the feature extractor, and then vector features are obtained by Gem pooling Radenovic et al. (2019), and sequence features are obtained by the variant Gem brightening method.

First, a very interesting observation is that our sequence representation method only needs 4 dimensions integer numbers to achieve 60% MAP, and the vector method that needs to achieve the same result currently requires 32 dimensions float data for feature representation. This also confirms that the features of an image can be represented by several corresponding convolutional kernels, capturing most of the feature information of the image on a fixed dataset. Second, it also shows that our method has better representation ability than the image representation methods in vector forms under the same dimension. Moreover, as the dimension of the sequence method increases, the MAP actually decreases, which reflects that the extractors of the later convolution kernel can interfere with the front extractors that may have crucial feature information. This

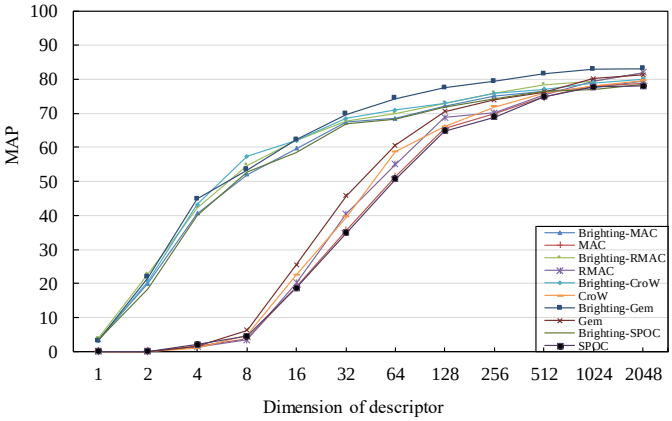


Fig. 8. The dimension comparison of different pooling method on Paris106K datasets.

phenomenon is consistent with the principle of our method, that is, after brightening operation the higher ranking of elements means the more importance for convolution kernel extractor of the correspond index, which indicates that the current kernel has strong representation ability for this image.

Table 2: Different dimension between ACK and the image representation methods for vector forms. ACK's dimension is the result of top-n. The image representation methods in vector forms is the result of PCA.

Dim	Holidays			Holidays101K		
	Ours		Vector	Ours		Vector
	Pretrained	Fine-tuned		Pretrained	Fine-tuned	
1	12.33	23.27	0	4.15	3.19	0
2	33.48	47.52	0.88	22.57	21.74	0.04
4	54.72	60.71	9.84	49.7	44.78	1.59
8	64.34	68.15	31.28	61.73	53.42	6.26
16	71.9	75.38	50.84	71.31	62.18	25.46
32	77.35	80.79	65.94	76.79	69.72	45.75
64	80.45	83.49	74.32	80.44	74.28	60.61
128	84.71	85.39	80.34	84	77.43	70.54
256	86.72	86.33	81.96	86.69	79.42	73.94
512	88.61	87.3	83.5	87.64	81.53	76.92
1024	89.57	87.62	85.98	88.11	82.89	80.24
2048	90.26	87.37	87.35	88.62	83.08	81.32

5.7.2. Dimension comparison of different pooling method

As shown in Fig 8, we use the fine-tuned Resnet-50 model and the adopted datasets are Paris106K. The last convolutional layer is selected as the feature extractor, and then vector features are obtained by pooling method, and sequence features are obtained by the variant brightening method. In Fig 8,our method has better stability than the naive method, which illustrates that our method can still obtain key feature information with low dimension.

5.7.3. Discussions

This section compares the representation ability of the proposed sequence method and the vector method in the corresponding dimension, and the experimental results show that our method outperforms the vector method. Meanwhile, as the dimension of the ACK method increases, its retrieval accuracy decreases in varying degrees, which also confirms that the sensitivity of convolution kernels in network is different. For instance, image features may only need a few main convolution kernel extractors to characterize the main feature information.

Table 3: Comparison of different brightening method in different networks.

	VGG-16	Resnet-50	Resnet-101	Resnet-152
Brighting-SPOC	72.53	70.16	70.2	72.63
Brighting-MAC	74.43	75.2	75.69	76.12
Brighting-RMAC	74.09	73.19	73.13	74.73
Brighting-CroW	74.32	73.2	73.98	75.03
Brighting-Gem	75.33	75.76	76.49	77.28

5.8. Comparison among different networks and methods

To test versatility of our ACK method, we conduct experiments on Paris6k dataset and report the best performance of our proposed ACK method under different networks and different pooling methods in Table 3. The dimension of all index sequences is 128, and all network structures are consistent with the network structure described in the paper (Georgakis et al., 2018). First, it can be seen that the mean brightening method has the worst performance combined with different networks, while Gem method with brightening can obtain the highest accuracy. The brightening can highlight the image intrinsic features combining with the Gem's trainable pooling parameter, which achieves the best performance among these methods. Second, the performance is better when the network is deeper. In addition, if we want to improve the accuracy of retrieval, we need a stronger network. However, from the practical point of view, expanding the network structure will increase the training difficulty and time, but the actual overall accuracy improvement is relatively small.

5.9. Comparison with the state-of-the-art image retrieval methods

In Table 4, we compare some of the classic feature extraction methods including MAC Razavian et al. (2014), RMAC Tolias et al. (2016), CroW Kalantidis et al. (2016), SPOC Babenko and Lempitsky (2015), Gem Radenovic et al. (2019), after fine-tuning as well as the direct feature extraction methods. The fine-tuned VGG-16 network is adopted in all experiments of this section. We compare the feature representation capabilities of different brightening methods, different pooling methods and some existing image retrieval methods. We adopt the existing vector

Table 4: Performance comparison with the state-of-the-art image retrieval method using VGG deep networks.

	Fine-tuned	Dimension	Data Type	Oxford5K	Oxford105K	Paris6K	Paris106K	Holidays	Holidays101K
SPOC	no	512	float	39.61	36.77	63.52	53.5	78.86	75.83
SPOC	yes	512	float	58.36	51.31	74.09	61.16	82.86	73.66
Ours	yes	256	int	62.15	55.79	74.6	62.34	84.64	77.19
MAC	no	512	float	43.06	39.66	67.02	57.38	83.1	75.68
MAC	yes	512	float	64.98	58.56	78.73	68.5	83.33	75.35
Ours	yes	256	int	63.41	57.91	75.69	65.24	83.65	75.11
RMAC	no	512	float	49.53	45.59	72.02	62.78	85.21	77.39
RMAC	yes	512	float	62.17	59.12	77.94	67.26	84.13	74.73
Ours	yes	256	int	64.04	57.91	75.76	63.95	84.72	76.21
CroW	no	512	float	43.79	40.52	68.94	59.3	83.05	74.99
CroW	yes	512	float	65.32	53.03	77.48	65.96	82.59	73.97
Ours	yes	256	int	63.15	56.62	75.94	64.32	83.97	75.93
Gem	yes	512	float	64.75	57.98	79.67	69.64	84.02	76.29
Ours	yes	256	int	64.98	57.87	76.26	65	84.36	76.47

method with the proposed index sequence method on three retrieval datasets. First, the results shows that under the same experiment conditions, our index sequence method could achieve better results than the image representation methods in vector forms, and improve on the basis of various existing pooling methods. Second, our method can achieve the best results in all datasets, because aggregating the image intrinsic features can receive the strong capability of feature representation, which are achieved by brightening on each kernel. Finally, the dimension of feature representation of our method is low. On large-scale datasets, the results of our sequence method are more robust than the image representation methods in vector forms.

6. Conclusion

In this work, we propose a new image representation method, aggregating convolution kernels (ACK), inspired by the visualizations of CNN convolution kernel, which shows strong discrimination and robustness. In details, the convolution kernels in a layer are ranked based on their response intensity to image intrinsic features, which are achieved by brightening on each kernel, i.e. finding the index that maximizes the strength of the convolution kernel response to image feature, respectively. Furthermore, to objectively evaluate the performance of the proposed image sequence representation, a similarity measurement of two index sequences is designed for the proposed image representation. Besides, we implement it on image retrieval tasks, and compare it with the current CNN-based vector representation method, and achieve the better performance in experiments. The proposed image representation method can alleviate the problems of high dimension, high memory and high computational complexity existing in the current image representation methods for vector forms, which can promote the application of computer vision to some practical large projects with limited memory. We believe that our index sequence for image representation may serve as a new field of basic research. In future work, we will explore the common algorithms in CNN for index sequence for image representation, such as classifier, regression, approximate nearest neighbor (ANN) search, principal component analysis (PCA), whitening, extended query and etc..

References

A. Babenko and V. S. Lempitsky. Aggregating deep convolutional features for image retrieval. *CoRR*, abs/1510.07493, 2015.

J. Bai, B. Ni, M. Wang, Y. Shen, H. Lai, C. Zhang, L. Mei, C. Hu, and C. Yao. Deep progressive hashing for image retrieval. In *ACM on Multimedia Conference*, pages 208–216, 2017. doi: 10.1109/TMM.2019.2920601.

A. Bhat. Makeup invariant face recognition using features from accelerated segment test and eigen vectors. *Int. J. Image Graphics*, 17(1):1–11, 2017. doi: 10.1142/S021946781750005X.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: binary robust independent elementary features. In *ECCV*, pages 778–792, 2010. doi: 10.1007/97836421556l1_56.

M. G. Capra. Factor analysis of card sort data: an alternative to hierarchical cluster analysis. *Human Factors & Ergonomics Society*, 49(5):691–695, 2005. doi: 10.1177/154193120504900512.

B. Chen and W. Deng. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00286.

Z. Cheng, X. Wu, Y. Liu, and X. Hua. Video2shop: Exactly matching clothes in videos to online shopping images. *CoRR*, abs/1804.05287, 2018.

O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total recall II: query expansion revisited. In *CVPR 2011*, pages 889–896, 2011. doi: 10.1109/CVPR.2011.5995601.

Y. N. Claure, E. T. Matsubara, C. Padovani, and R. C. Prati. Polywatt: A polynomial water travel time estimator based on derivative dynamic time warping and perceptually important points. *Computers & Geosciences*, 112:54–63, 2018. doi: 10.1016/j.cageo.2017.12.002.

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, volume 1, pages 1–2, 2004.

A. ElAdel, M. Zaied, and C. B. Amar. Fast DCNN based on fwt, intelligent dropout and layer skipping for image retrieval. *Neural Networks*, 95:10–18, 2017. doi: 10.1016/j.neunet.2017.07.015.

D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018.

G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Košecká. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In *CVPR*, pages 1965–1973, June 2018. doi: 10.1109/CVPR.2018.00210.

A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, pages 241–257, 2016. doi: 10.1007/9783319464664_15.

A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. doi: 10.1007/s1126301710168.

H. Guo, K. Zheng, X. Fan, H. Yu, and S. Wang. Visual attention consistency under image transforms for multi-label image classification. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00082.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, pages 774–787, 2012. doi: 10.1007/9783642337093_55.

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008. doi: 10.1007/9783540886822_24.

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010. doi: 10.1109/CVPR.2010.5540039.

Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCV*, pages 685–701, 2016. doi: 10.1007/9783319466040_48.

E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *SIAM*, pages 1–11, 2001. doi: 10.1137/1.9781611972719.1.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. *CoRR*, abs/1703.04730, 2017.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. doi: 10.1145/3065386.

K. Liu, H. Wang, F. Nie, and H. Zhang. Learning multi-instance enriched image representations via non-greedy ratio maximization of the l1-norm distances. In *CVPR*, pages 7727–7735, 06 2018. doi: 10.1109/CVPR.2018.00806.

Y. Liu, F. Nie, Q. Gao, X. Gao, J. Han, and L. Shao. Flexible unsupervised feature extraction for image classification. *Neural Networks*, 115:65–71, 2019. doi: 10.1016/j.neunet.2019.03.008.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

J. Y. Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval. In *CVPR*, pages 53–61, 2015. doi: 10.1109/CVPRW.2015.7301272.

D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006. doi: 10.1109/CVPR.2006.264.

K. Pang, K. Li, Y. Yang, H. Zhang, T. M. Hospedales, T. Xiang, and Y.-Z. Song. Generalising fine-grained sketch-based image retrieval. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00077.

F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010. doi: 10.1007/9783642155611_11.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. doi: 10.1109/CVPR.2007.383172.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. doi: 10.1109/CVPR.2008.4587635.

F. Radenovic, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1655–1668, 2019. doi: 10.1109/TPAMI.2018.2846566.

A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *Ite Transactions on Media Technology and Applications*, 4, 2014. doi: 10.3169/mta.4.251.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443, 2006. doi: 10.1007/11744023_34.

P. C. Roy and V. N. Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00269.

M. Sarigul, B. M. Ozyildirim, and M. Avci. Differential convolutional neural network. *Neural Networks*, 116:279–287, 2019. doi: 10.1016/j.neunet.2019.04.025.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. URL `http://arxiv.org/abs/1409.1556`.

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003. doi: 10.1109/ICCV.2003.1238663.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR*, 2015. URL `http://arxiv.org/abs/1412.6806`.

K. Sun, S. Mou, J. Qiu, T. Wang, and H. Gao. Adaptive fuzzy control for nontriangular structural stochastic switched nonlinear systems with full state constraints. *IEEE Transactions on Fuzzy Systems*, 27(8):1587–1601, 2019. doi: 10.1109/TFUZZ.2018.2883374.

K. Sun, J. Qiu, H. R. Karimi, and H. Gao. A novel finite-time control for nonstrict feedback saturated nonlinear systems with tracking error constraint. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–12, 2019. doi: 10.1109/TSMC.2019.2958072.

K. Sun, L. Liu, J. Qiu, and G. Feng. Fuzzy adaptive finite-time fault-tolerant control for strict-feedback nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 2020. doi: 10.1109/TFUZZ.2020.2965890.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. URL `http://arxiv.org/abs/1312.6199`.

C. Tian, Y. Xu, and W. Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. doi: 10.1016/j.neunet.2019.08.022.

G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*, 2016. URL `http://arxiv.org/abs/1511.05879`.

N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays. Composing text and image for image retrieval - an empirical odyssey. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00660.

L. Wang, B. Wang, Z. Zhang, Q. Ye, L. Fu, G. Liu, and M. Wang. Robust auto-weighted projective low-rank and sparse recovery for visual representation. *Neural Networks*, 117:201–215, 2019a. doi: 10.1016/j.neunet.2019.05.007.

Q. Wang, J. Lai, K. Xu, W. Liu, and L. Lei. Beauty product image retrieval based on multi-feature fusion and feature aggregation. In *ACM Multimedia*, pages 2063–2067, 2018. doi: 10.1145/3240508.3266431.

Y. Wang, X. Tao, X. Shen, and J. Jia. Wide-context semantic image extrapolation. In *CVPR*, June 2019b. doi: 10.1109/CVPR.2019.00149.

D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang. Deep incremental hashing network for efficient image retrieval. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00928.

X. Yang, N. Wang, B. Song, and X. Gao. Bosr: A cnn-based aurora image retrieval method. *Neural Networks*, 116:188–197, 2019. doi: 10.1016/j.neunet.2019.04.012.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolu-

tional networks. In *ECCV*, pages 818–833, 2014. doi: 10.1007/978-3319105901_53.

F. Zhan and S. Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, June 2019. doi: 10.1109/CVPR.2019.00216.

L. Zheng, Y. Yang, and Q. Tian. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40 (5):1224–1244, 2018. doi: 10.1109/TPAMI.2017.2709749.

L. Zhou and X. Gu. Embedding topological features into convolutional neural network salient object detection. *Neural Networks*, 121: 308–318, 2020. doi: 10.1016/j.neunet.2019.09.009.

Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He. Multi-representation adaptation network for cross-domain image classification. *Neural Networks*, 119:214–221, 2019. doi: 10.1016/j.neunet.2019.07.010.