# Image retrieval using deep statistical feature descriptor

**Zhou Lu**
Guangxi Normal University
**Guang-Hai Liu** ( liuguanghai009@163.com )
Guangxi Normal University https://orcid.org/0000-0002-1558-2694

**Research**

**Keywords:** image retrieval, CNN, deep feature, texture feature, deep statistical feature descriptor

**Posted Date:** June 1st, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1675224/v1

# Image retrieval using deep statistical feature descriptor

**Zhou Lu, Guang-Hai Liu**

**liuguanghai009@163.com**

**Abstract.** The use of deep neural networks to extract deep high-level features has provided excellent image retrieval performance. However, few studies have analyzed low-level features, which have their own advantages in image representation. Combining the advantages of low-level features and deep features remains a challenging problem. To solve it, we propose a deep feature aggregation method that includes texture features. Its highlights are: (1) A novel covariance weight is proposed to select some key feature maps. This enhances the discriminativeness of feature maps. (2) A novel response-region weighting scheme is proposed to balance various response-regions weights. It improves the positive role of some valuable feature maps in retrieval. (3) A novel method is proposed to extract texture features together with deep features. It can combine deep and texture features into a compact representation, and improve the performance of image retrieval. Experiments were conducted with five popular benchmark datasets to show that our method can significantly improve image retrieval performance.

**Keywords:** image retrieval, CNN, deep feature, texture feature, deep statistical feature descriptor.

## 1 Introduction

After decades of development, image retrieval methods have become divided into two categories: text-based and content-based. The former requires manual design of image labels, which is difficult to implement for large databases. Nowadays, content-based image retrieval has become the mainstream method. The main feature of this kind of method is to design the image representation vector based on the features of the image. With the development of deep learning, image retrieval based on convolutional neural network has attracted close attention. In recent work, researchers have proposed some feature aggregation methods based on the deep features of convolutional neural networks, and achieved good retrieval performance [1][2].

Although deep features have advantages in the representation of high-level semantics, low-level features such as color and texture still have an important role in image retrieval. The researchers extract visual information of the entire image, such as color, texture, and other features. These features provide good distinguishability [3][4]. Liu et al. constructed some effective image representations based on these features, which significantly improved the performance of image retrieval [5][6].

Texture features are commonly used visual features in the field of computer vision. It provides useful information when analyzing different image regions or object classes. For example, in the two images of **Fig. 1**, the building in the left image has a tall spire and the roof of the building has the visual feature of uneven height, while in the right image, the top of the

building is smooth. Humans can accurately use these visual features to distinguish between the two types of objects. In fact, these visual features can be identified because the various spatial distributions of pixel blocks produce various textural clues. Inspired by this view, we combine texture and deep features to improve the representation.

The covariance between texture and deep features has close relationship with the key feature maps. Aggerating the key feature maps can provide a robust descriptor and enhance the representation. In this paper, we propose a novel descriptor named the *deep statistical feature* (DSF) descriptor to represent deep convolutional features, and apply it to image retrieval.

The main contributions of this paper are as follows: 1) A *covariance weight* is proposed to select some key feature maps, it can enhance the discriminativeness of feature maps, 2) A novel *response-region weighting scheme* is proposed to balance various response-regions weights. It improves the positive role of some valuable feature maps in retrieval. 3) A novel method is proposed to extract texture features together with deep features. It can combine deep and texture features into a compact representation, and improve the performance of image retrieval.

The rest of this paper is organized as follows. In Section 2, we summarize related research on image retrieval. In Section 3, we describe our proposed method. Section 4 presents the experiments and compares their results with those of other methods. Section 5 concludes the paper.

**Fig 1.** Images from datasets Oxford5k [41] (left) and Paris6k [42] (right). Note the obvious differences in texture features.

## 2 Related works

The key to image retrieval technology is the representation of image content. Research on image representation methods has gone through three stages.

In the first stage, global feature representation was used to describe image content. Global features are based on the whole attributes of an image, including color, texture, shape, and spatial layout, etc. Based on color features, Swain proposed a color histogram based on the proportions of different colors in an image [7]. Based on the color histogram, Lai proposed the color correlation map, which adds spatial features to color features [8]. In terms of texture features, the GLCM [9] proposed by Haralick et al. contains the spatial relationships between image grey levels and is a classic texture-feature algorithm. In addition, Markov random field model [10], Gabor filtering [11], local binary pattern [12], etc. are also commonly used texture feature algorithms. Shape feature is one of the important low-level features for object recognition and has been widely used in shape analysis and object recognition applications [13-16]. Commonly used shape feature

descriptors include region-based shape descriptor, and curvature scale space descriptor [17], etc.

In the second stage, local feature representation was used to represent image content. Compared with global features, local features are more robust. Among them, the SIFT feature proposed by David Lowe is a typical local-feature representation [18]. It can construct a descriptor that is robust to image rotation and scaling. The researchers constructed bag-of-visual words model (BOW) [19] by borrowing text retrieval technology. Nowadays, methods based on bag-of-visual words model are widely used in image retrieval, object recognition and other fields [20-23]. However, when the local feature method is used for large-scale image datasets, it will take up a lot of memory and the computational cost is high. To improve this situation, researchers have devised various encoding methods. Typical methods include the vector of locally aggregated descriptors (VLAD) [24], and the Fisher vector (FV) [25].

The research of Krizhevsky et al. promoted the development of deep convolutional neural networks for image retrieval [26]. Subsequently, image representation entered the third stage—deep feature representation. In order to obtain better deep features, researchers have developed many CNN models. For example, the commonly used VGGNet [27] increases the depth of AlexNet [26] and modifies the size of the convolution kernel. GoogLeNet [27] reduces the number of parameters used. These models have achieved good results in many fields of computer vision, and increasing numbers of researchers are exploring the use of deep features. Through many experiments, researchers proved that deep features based on convolutional neural networks (CNNs) are very effective in image retrieval and classification [28-37]. For example, Babenko et al. used a direct superposition summation of the feature maps of the last convolutional layer of a CNN and assigned a Gaussian weight to each feature map, which is referred to as SPoC [28]. Tolias et al. proposed to sample local features of the feature map by using a sliding window, and it can provide a global representation, namely regional maximum activation of convolutions (R-MAC) [29], to image retrieval. Kalantidis et al. proposed al deep convolution feature aggregation method based on spatial and channel weights, named Crow [30]. Xu et al. proposed that channels with large variances have better target discrimination capabilities [36], but they did not consider the differences between them when aggregating feature maps. Recently, Liu et al. [38] proposed a method for selecting deep features using low-level features by studying the human visual system. This method constructs a new image feature descriptor, which provides ideas for combining low-level features with deep features.

To sum up, there is still room for improvement in the processing of image features by previous deep feature-based image retrieval techniques, and they lack research on low-level features. Low-level features and deep features have their own advantages in image representation, but it is still challenging to combine their advantages to construct image representations. Aiming at these problems, this paper proposes a feature extraction method based on the correlation between deep features and texture features. Through this method we improve the recognition ability of image representation.

## 3 The proposed method

In this section, we introduce our proposed unsupervised method, which can generate powerful and distinguishable feature representations for image retrieval. Its main concept is to combine the texture features with deep features to calculate the *response area weights* (RAW), which are used to enhance the deep features together with one of the GLCM statistics.

In here, the deep features are extracted from the feature maps of pool5 layer in a pre-trained VGG16 model, because it is very popular CNN model applied in instance retrieval [29,30,36,39,40]. The framework of the proposed method is illustrated in **Fig. 2**.

Our CBIR framework is divided into four steps: (1) We calculate the covariance weights to select some key feature maps. (2) then the response area weights are calculated in the select some key feature maps. (3) Construct the difference weights based on the differences between deep features by computing the differences between deep features using the GLCM statistics. (4) Aggregating the weighted feature maps and the *difference weights* (DW) into a compact representation, the DSF descriptors are obtained after PCA Whitening, and then are utilized for image retrieval.

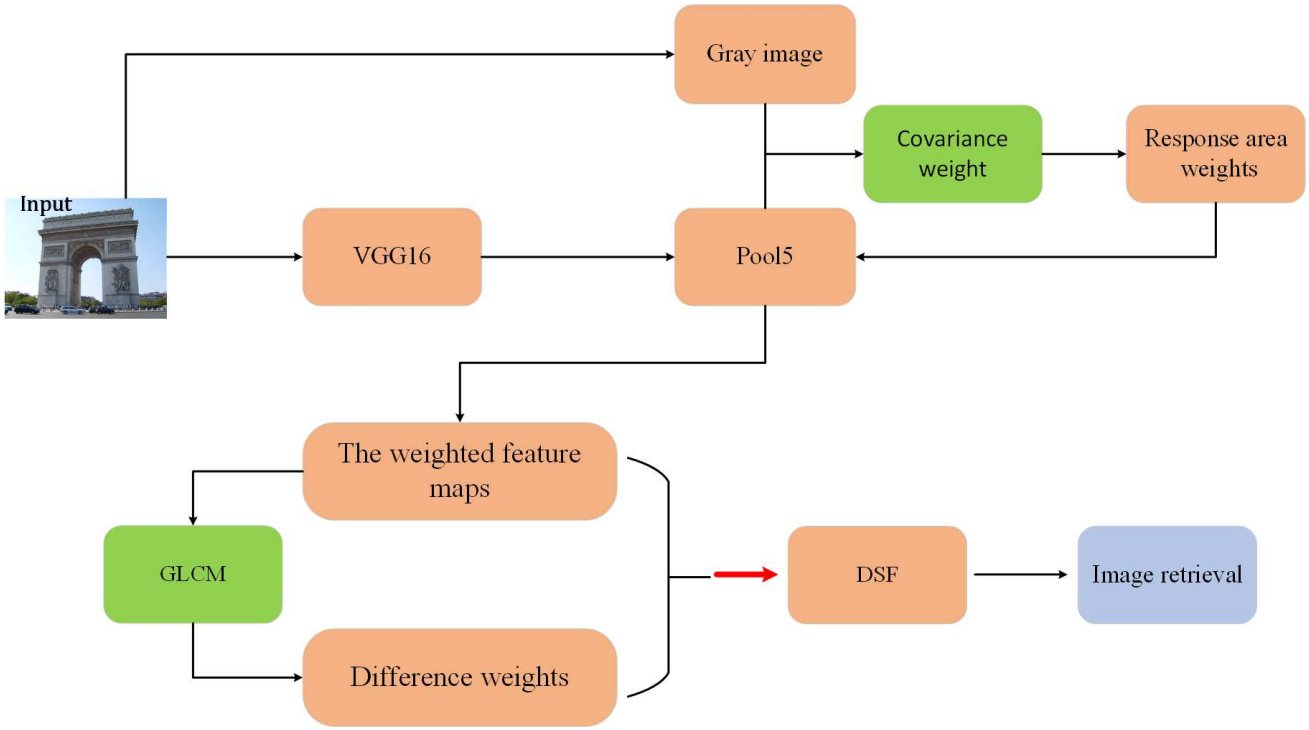


**Fig 2.** Basic framework of the proposed method.

### 3.1 Covariance weight

The covariance value can be used to measure the overall error of two variables. Specifically, if the covariance value of the two variables is positive, the variables have the same trend; if negative, the opposite is true. We can calculate the covariance between texture features and deep features to distinguish their correlation and select the key feature maps. Deep features with high correlation have a better ability to represent images.

The feature maps of pool5 layer are defined as $X \in \mathcal{R}^{K\times W\times H}$, and $X = \{x_k\}_{k=1}^{K}$, where x denotes a certain feature map, $W$ and $H$ denote its width and height, respectively, and $K$ denotes the number of feature maps or channels.

We transform the original image into a grayscale image. In order to improve the compatibility of grayscale images and feature maps, we normalize the grayscale images. We resize it to match the size of the feature map, where we also scale its grayscale value and make it has the same values range as the feature map. The normalized grayscale image is more conducive to calculating the similarity. In here, we denote the grayscale image as $I$.

We transform the feature maps $X = \{x_k\}_{k=1}^{K}$ and *I* into a column vector via the flattening method and define the results as $q = \{q_k\}_{k=1}^{K}$ and Q:

$$\begin{cases} q_k = flatten(x_k), k \in [1,2,..,K] \\ Q = flatten(I) \end{cases} \tag{1}$$

Then, let $\bar{q}$ and $\bar{Q}$ denote the averages of q and Q, respectively. The calculation is:

$$\begin{cases} \bar{q} = \dfrac{\sum q}{W \times H} \\ \bar{Q} = \dfrac{\sum Q}{W \times H} \end{cases} \tag{2}$$

The covariance of the feature maps $X$ and *I* is denoted as $C = \{c_k\}_{k=1}^{K}$, and it is calculated as follow:

$$c_k = \frac{(Q - \bar{Q})^T \times (q_k - \bar{q})}{W \times H}, k \in [1,2, \dots, K] \tag{3}$$

where T represents the matrix transpose operation. We take the average value of $C$ and denote it as $C_{avg}$:

$$C_{avg} = \frac{\sum_{k=1}^{K} max\{c_k, 0\}}{\sum_{k=1}^{K} N'\{c_k > 0\}} \tag{4}$$

where $max\{\}$ denotes the maximum, and $N'\{\}$ denotes the number of $c_k > 0$. Thus, highly correlated feature maps with a covariance greater than $C_{avg}$ can be selected. The selected correlated feature maps are compacted to a covariance weight (CW):

$$CW = \frac{\sum_{k=1}^{K_1} x_k}{K_1} \tag{5}$$

where $K_1$ expresses their selected numbers. The CW is used to enhance the correlation representation of the feature maps. We combine CW with the feature map in a weighted manner, and denote the weighted feature map as $x_{cw}$. The calculation process is as follows:

$$x_{cw} = CW \cdot x_k \tag{6}$$

We aggregate the $x_{cw}$ into a K-dimensional vector denoted as $V_{cw}$ and the result as the vector set $V = \{V_{cw}^n\}_{n=1}^{N}$, where $N$ represent the number of images in the dataset.

We select the key feature maps to calculate the response area weights, and the selection strategy is divided into three steps as follows:

(1) Calculate the variances of $V$ and obtain a K-dimensional vector denoted as $V' = \{v_k\}_{k=1}^{K}$.

(2) Select the $K_1$ numbers of $V'$ with values greater than the average value of $V'$, and denote the result as $V'' = \{v_k\}_{k=1}^{K_1}$.

(3) Continue selecting $K_2$ numbers of $V''$ with values greater than the average of $V''$, and denote the result as $V''' = \{v_k\}_{k=1}^{K_2}$.

Thus, the feature maps corresponding to the values of $V'''$ are the key feature maps. We denote the selected key feature maps as $\{x_k\}_{k=1}^{K_2}$.

**3.2 Response area weights**

The selected key feature maps can provide various object information, so it is necessary to filter them. We obtain a filter in two ways.

In the first way, let $x'$ represent the compression of the selected key feature maps as:

$$x' = \frac{\sum_{k=1}^{K_2} x_k}{K_2} \tag{7}$$

Then, the first filter, which is denoted $m_1$, is generated as:

$$m_1 = x' - \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(x')_{i,j}}{W \times H} \tag{8}$$

In $m_1$, there may be positive numbers, negative numbers or zero. A positive number means that the value of $x'$ in the response region is greater than those in other regions, so that region is more likely to be the target object or key feature region.

In the second way, we mark locations with activation values > 0 as 1, and other locations as 0. Let M denote the marked result:

$$M_{(i,j)} = \begin{cases} 1, & if \ x_{(i,j)} > 0 \\ 0, & otherwise \\ where & x_{(i,j)} \in \{x\}_{k=1}^{K_2} \end{cases} \tag{9}$$

Then, the second filter is generated by calculating the numbers of marked results, and is denoted as $m_2$:

$$m_2 = \sum_{k=1}^{K_2} M_k \tag{10}$$

Before calculating the response area weights, we first quantify $x'$. We denote the quantized result by $S$. The formula is as follows:

$$S = \sqrt{\frac{x'}{\sum_{i=1}^{W}\sum_{j=1}^{H}(x')_{i,j}}} \tag{11}$$

We use $m_1$ and $m_2$ as conditions for using the tanh function at each location in $S$. The weight of the response area (RAW) is obtained after using the tanh function at each location. The calculation is:

$$RAW = \begin{cases} tanh(S_{i,j}), if \;\; m_1(i,j) > 0 \;\; and \;\; m_2(i,j) > t \\ \dfrac{tanh(S_{i,j})}{2}, \qquad otherwise \end{cases} \tag{12}$$

where parameter *t* represents the threshold of the number of responses and is the average of the maximum and minimum values on $m_2$.

The *response area weights* (RAW), which includes abundant feature correlation information, is used to enhance the feature maps. The Tanh function has a maximum of 1 and a minimum of 0 in the first quadrant, and is monotonically increasing. Thus, when using the response area weights to allocate weights, it not only gives the eligible regions of the feature map with large weights, but also avoids the variety caused by excessive weighting of a certain region.

### 3.3 The deep statistical feature descriptor

We weight each feature map with the RAW and obtain a new weighted feature map as follow:

$$\tilde{x} = RAW \cdot x \tag{13}$$

The weighted feature maps are denoted as $X' = \{\tilde{x}\}_{k=1}^{K}$.

In the feature map of pool5 layer, there will be some target objects with lower response values. However, they can still play a positive role in image retrieval. Based on this point, we need to further design the difference weights from the entropy statistic vector (ESV) and the element value ratio vector (EVRV) in each feature map:

**(a) The entropy statistic vector**

The texture features can play an important role in differentiating various images. In here, the GLCM of each feature map is calculated to reflect the spatial distribution relationships, and further to exploit the differences between deep features. Commonly, four GLCM statistics are extracted to represent the texture features in traditional methods.

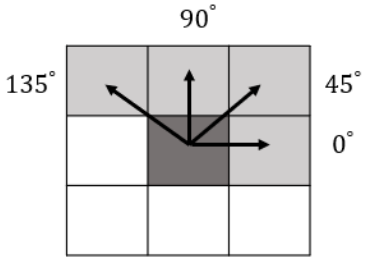


**Fig 3.** Orientation selection between adjacent grayscale values.

To help understand this technique, we illustrate the GLCM in more detail. In a GLCM, the spatial relationship between adjacent grayscale values can be represented quantitatively based on human visual system's orientation-selection. As is shown in **Fig. 3**, the four directions are chosen to represent the spatial relationship between two pixels, which are $\{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\}$. According to these four directions, different spatial relationships of a GLCM can be constructed and, therefore, the four GLCM statistics have different values. The average of the four directions of a certain statistic is used as the final texture feature vector.

Here, let P denote the GLCM of each feature map transformed from a weighted feature map $\tilde{x}$, and let M denote the gray level. In this paper, we take M = 8. The four statistics are calculated as follows:

Angular second moment $(ASM)$:

$$ASM = \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij}^2 \tag{14}$$

Contrast $(CON)$:

$$CON = \sum_{i=1}^{M}\sum_{j=1}^{M} (i-j)^2 P_{ij} \tag{15}$$

Inverse different moment $(IDM)$:

$$IDM = \sum_{i=1}^{M}\sum_{j=1}^{M} \frac{P_{ij}}{(1+(i-j)^2)} \tag{16}$$

Entropy $(ENT)$:

$$ENT = -\sum_{i=1}^{M}\sum_{j=1}^{M} P(i,j)\log_2 P(i,j) \tag{17}$$

Among these four statistics, ENT is a measure of information and can reflect the degree of non-uniformity in deep feature textures. Experiments show that ENT has a good effect; more detail is provided in Section 4.1. Thus, in this paper, we used ENT as the texture feature descriptor and gathered the ENT of each feature map to obtain an entropy statistic vector (ESV) as:

$$ESV = \{ENT_k\}_{k=1}^{K} \tag{18}$$

**(b) The element value ratio vector**

The activation values of the feature map reflect the sensitivity of a target object. Therefore, we calculate the proportion of the activation values as the element value ratio vector (EVRV), and $EVRV = \{av_k\}_{k=1}^{K}$

$$av_k = \frac{F_k}{\sum_{k=1}^{K} F_k}, k \in [1,2,..,K] \tag{19}$$

Where $\{F_k\}_{k=1}^{K}$ is a vector obtained by using sum-pooling in the weighted feature maps $X'$.

**(c) The difference weights**

In here, we define the difference weights $DW = \{dw_k\}_{k=1}^{K}$ as:

$$dw_k = \begin{cases} \log_e\left(\frac{\sum_{k=1}^{K}(av_k \cdot ENT_k)}{av_k \cdot ENT_k}\right), av_k \cdot ENT_k > 0 \\ 0\ , \qquad otherise \end{cases} \tag{20}$$

in which using a logarithmic function achieves the purpose of assigning large weights to insignificant features.

Finally, we define the feature vectors $V = \{v_k\}_{k=1}^{K}$ as:

$$v_k = F_k \cdot dw_k \tag{21}$$

The *deep statistical feature* (*DSF*) descriptor can be obtained by applying the L2-normalization and PCA Whitening method to the $V$.

## 4 Results and discussion

We conducted experiments on five datasets, namely, Oxford5k [41], Paris6k [42], Holidays [43], Oxford105K, and Paris106k. Oxford105k and Paris106k are based on the Oxford5k and Paris6k datasets with added Flickr100k interference images.

For Oxford5k and Oxford105k, we used the cropped areas of each image as a query, as provided by the oxford5k dataset protocols. We think that this complicates feature extraction, so this method was not used with the other datasets. For the Holidays dataset, we considered the 500 query images as "junk" images, as described by the authors.

For the CNN model, we used the public pre-trained VGG16 network model to extract deep features. We used the mean average precision (mAP) as the evaluation metric to measure retrieval performance, where the evaluation code is provided by the authors and a higher mAP indicates better performance. We used the L2 distance to calculate the similarity between feature vectors.

### 4.1 Texture feature testing

The statistics representing texture features designed by Haralick et al. [9] based on the GLCM matrix method have different physical meanings. Previous methods would combine them into a feature vector to represent texture features. But we think that their uses should be distinguished according to the physical meaning they represent. For example, the entropy statistic has higher values for more complex images [9][44]. According to this, we can judge the information content of the feature map by the entropy statistic. The larger the entropy, the more image features the feature map contains. In order to analyze the effect of using the four GLCM statistics, we conducted several experiments to test them. The step of GLCM calculation

involves the ASM, CON, IDM, ENT, and EVRV. Here, we set up two types of experiments: with and without the use of EVRV.

Firstly, without EVRV, the ENT, ASM, CON, and IDM were used as texture feature descriptors to calculate ESV to obtain deep feature descriptors for image retrieval. According to the combinations of strategies, we conducted three groups of tests.

The first group used a single statistic for calculation. According to the experiment, it is best to use ENT on Oxford5k and Paris6k. The second group of tests used a two-by-two unordered combination of the four statistics for calculation. The combination of IDM and ENT (IDM · ENT) on Oxford5k was found to provide better performance. On Paris6k, CON · ENT is recommended. The third group of tests combined three or four statistics. According to the experiment, the ASM · IDM · ENT combination had the best effect on Oxford5k, while CON · IDM · ENT was better with Paris6k.

**Table 1.** Comparison of mAP results obtained using different statistical combinations on the Oxford 5k and Paris6k datasets.

| Statistic | Dimensions | Oxford5k | Paris6k |
|---|---|---|---|
| ENT | 128 | **0.697** | 0.793 |
| IDM · ENT | 128 | 0.695 | 0.792 |
| CON · ENT | 128 | 0.684 | **0.797** |
| ASM · IDM · ENT | 128 | 0.690 | 0.786 |
| CON · IDM · ENT | 128 | 0.687 | **0.797** |
| ENT | 256 | **0.724** | 0.818 |
| IDM · ENT | 256 | **0.724** | 0.817 |
| CON · ENT | 256 | 0.713 | **0.819** |
| ASM · IDM · ENT | 256 | 0.721 | 0.813 |
| CON · IDM · ENT | 256 | 0.714 | **0.819** |
| ENT | 512 | **0.747** | **0.844** |
| IDM · ENT | 512 | 0.746 | **0.844** |
| CON · ENT | 512 | 0.738 | 0.841 |
| ASM · IDM · ENT | 512 | 0.743 | 0.841 |
| CON · IDM · ENT | 512 | 0.739 | 0.841 |

According to these three groups of tests, we derived the best combination of statistics for each group. Next, we compare their performance to pick the best strategy. The results are compared in **Table 1**. It can be seen that using ENT as a texture feature vector performs better than using other statistics in three dimensions on Oxford5k. On Paris6k, in the 128- and 256-dimension cases, using ENT is slightly worse than the other two combinations (0.1%, 0.4%). However, with 512-dimensions, using ENT is better than the other combinations. Moreover, some combinations that performed well on Paris6k did not perform well on Oxford5k. Therefore, we conclude that ENT is the best texture feature vector for calculating the ESV.

**Table 2.** mAP values of retrieval using EVRV and EVRV · ESV.

| Method | Dimensions | Oxford5k | Paris6k |
|---|---|---|---|
| *EVRV* | 128 | 0.705 | 0.801 |
| *ESV · EVRV* | 128 | **0.724** | **0.825** |
| *EVRV* | 256 | 0.732 | 0.830 |
| *ESV · EVRV* | 256 | **0.741** | **0.842** |
| *EVRV* | 512 | 0.754 | 0.857 |
| *ESV · EVRV* | 512 | **0.766** | **0.860** |

In addition, in order to test the effectiveness of using *ESV* with *EVRV*, we conducted a comparative experiment. The results are compared in **Table 2.** and show that better performance can be obtained using *ESV·EVRV*. Thus, the *DW* generated by combining *ESV* with *EVRV* is better in improving retrieval accuracy, and the effect of *DW* is more obvious in low-dimensional cases. This suggests that the information provided by texture features can complement depth features. Texture features are more obvious in lower dimensions, which can enhance the recognition ability of image representation.

### 4.2 Method testing and discussion

To illustrate the superiority of our method and the effects of its three parts *(CW, RAW, DW*), we conducted several sets of experiments.

To test the effect of the *CW*, we removed it kept the other parts unchanged to obtain feature representation for image retrieval. The experimental results (**Fig. 4**) show that the mAP of retrieval decreases obviously after removing the *CW*, especially in low-dimension cases. The covariance weight can enhance the correlation between the feature map and the original image, which is conducive to the selection of key features. Removing the covariance weight affects the selection of key features, thereby reducing the accuracy. This indicates that the *CW* part of our method is important.

Next, we tested the effect of removing the *RAW*; i.e., without considering the number of activation values of the feature maps, the *RAW* is degenerated into a spatial weight that only considers the magnitude of the activation value. Here, we used the spatial weight of the Crow method for comparative testing [30]. The experimental results are shown in **Table 3.** It is obvious that using the *RAW* in the proposed method can improve the retrieval effect, especially in low-dimension cases. The response region weights (RAW) can enhance the target features in the feature map, thereby improving the recognition ability of the image representation.

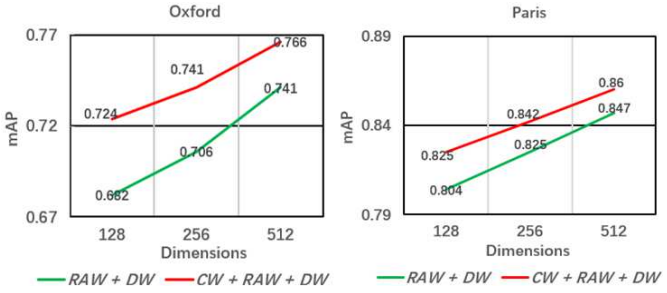


**Fig. 4.** mAP values of retrieval methods, with and without the use of *CW*, applied to the Oxford5k and Paris6k datasets.

**Table 3**. mAP comparison of retrieval with 128, 256, and 512 dimensions, where *CW + SP + DW* denotes that our Tanh weight was not used but the spatial weight (*SP*) was used.

| Method | Dimensions | Oxford5k | Paris6k |
|---|---|---|---|
| *CW + SP + DW* | 128 | 0.705 | 0.815 |
| *CW+RAW+DW* | 128 | **0.724** | **0.825** |
| *CW + SP + DW* | 256 | 0.737 | 0.838 |
| *CW + RAW + DW* | 256 | **0.741** | **0.842** |
| *CW + SP + DW* | 512 | 0.764 | 0.857 |
| *CW + RAW + DW* | 512 | **0.766** | **0.860** |

Lastly, we tested the effect of excluding the *DW*. We compared the performance of the two weighting methods from two

perspectives.

In the first sub-experiment, in order to test whether *DW* can also work when applied to other algorithms, we replaced the channel weight with *DW* and kept the other parts unchanged for retrieval, and denoted it as Crow (*DW*). The experimental results (**Table 4.**) show that our *DW* can be used with other methods and can improve their performance.

In the second sub-experiment, in order to test the effect of *DW*, we removed it and retained the other parts for retrieval. The experimental results (**Fig. 5**) indicate that *DW* increases performance significantly on the Oxford5k and Paris6k datasets. Thus, the two sub-experiments demonstrate that our *DW* can effectively improve retrieval performance and has enough generality to be used with similar methods.

**Table 4.** mAP comparison of the two methods, where Crow represents the original method.

| Method | Dimensions | Oxford5k | Paris6k |
|---|---|---|---|
| Crow (*DW*) | 128 | **0.659** | **0.775** |
| Crow [25] | 128 | 0.641 | 0.746 |
| Crow (*DW*) | 256 | **0.696** | **0.794** |
| Crow [25] | 256 | 0.691 | 0.765 |
| Crow (*DW*) | 512 | **0.725** | **0.820** |
| Crow [25] | 512 | 0.708 | 0.797 |

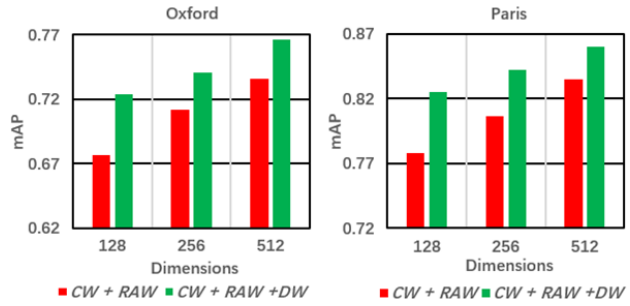


**Fig 5.** mAP comparison of retrieval performance with (green bars) and without *DW* (red bars).

All in all, these experiments demonstrate that the *CW, RAW*, and *DW* are all important parts of our method and greatly improve retrieval performance.

### 4.3 Retrieve performance and comparison

In the proposed method, we also perform PCA Whitening [45] on the feature vectors to obtain the DSF. For better comparison with other experiments, in this paper, we also use different datasets to learn the whitening parameters. For Oxford5k and Oxford105K datasets, we learn the whitening parameters using Paris6k dataset, while we use Oxford5k dataset to learn the whitening parameters on Paris6k and Paris106k datasets. On Holidays dataset, we used Oxford105k dataset to learn the whitening parameters, as others do.

To further improve the retrieval performance, we also use query expansion techniques [46]. The first query results are sorted in ascending order of similarity, then we aggregate and average the first $m$ query results into a feature vector, and re-query after performing L2 normalization. **Fig. 6** shows the test results of query expansion. Experimental results show

that using the first seven query results works best on Oxford5k dataset, while using the first 50 query results is best on Paris6k dataset. These two parameters will also be applied to Oxford105k and Paris106k datasets. So, combining query expansion with our method can improve retrieval performance. For better comparison with other experiments, we uniformly select the top 10 query results.

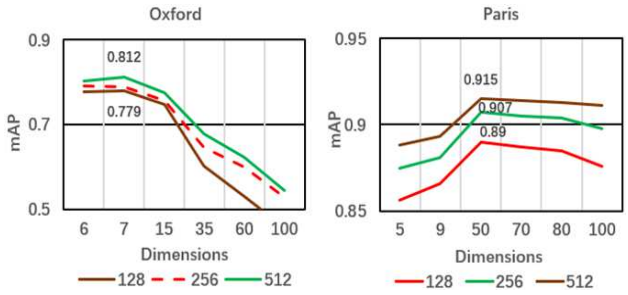


**Fig 6.** mAP of retrieval performance using query expansion on Oxford5k and Paris6k.

Overall, our method achieves good results on all five public datasets. Compared with the classical deep feature methods (e.g. Crow, SBA), our method highlights the target object in the feature map through the correlation of the deep features with the original image, thereby improving the positive role of key features in retrieval. Not only that, we incorporate the information of texture features into the final image feature representation, which improves the recognition ability of low-dimensional image representations. Comparative experiments show that on the Oxford5k dataset, DSF is on average 6% higher than other methods. On the Paris6k dataset, DSF also has a strong advantage, outperforming other methods by 4-8%. DSF also has good results for Oxford105k and Paris6k datasets. On the Holidays dataset, our method is slightly lower than some methods, however, we do not take preprocessing operations on the holidays data. Thus, our method has great advantages in retrieval performance. The specific experimental results are shown in **Table 5**.

**Table 5.** mAP comparison of the proposed method and other classic methods on five datasets. QE = query expansion.

| Method | Dimensions | Oxford5k | Oxford105k | Paris6k | Paris106k | Holidays |
|---|---|---|---|---|---|---|
| Neural Codes [32] | 512 | 0.435 | 0.392 | / | / | / |
| R-MAC [29] | 512 | 0.669 | 0.616 | 0.830 | 0.757 | / |
| NetVLAD [47] | 512 | 0.676 | / | 0.749 | / | 0.861 |
| CroW [30] | 512 | 0.708 | 0.653 | 0.797 | 0.722 | 0.851 |
| CAMs [48] | 512 | 0.712 | 0.672 | 0.805 | 0.733 | / |
| SBA [36] | 512 | 0.720 | 0.662 | 0.823 | 0.758 | 0.852 |
| ReSW [39] | 512 | 0.726 | / | 0.824 | / | 0.853 |
| HeW [40] | 512 | 0.728 | / | 0.824 | / | 0.884 |
| LASC [49] | 512 | 0.671 | / | / | / | 0.909 |
| AILIR(Cross) [50] | 512 | 0.707 | 0.687 | 0.852 | 0.847 | / |
| **DSF** | 512 | **0.766** | **0.670** | **0.860** | **0.769** | **0.841** |
| Neural Codes [32] | 256 | 0.435 | 0.392 | / | / | 0.749 |
| Razavian et al. [31] | 256 | 0.533 | 0.489 | 0.670 | / | 0.716 |
| SPoC [28] | 256 | 0.531 | 0.501 | / | / | 0.802 |
| R-MAC [29] | 256 | 0.561 | 0.470 | 0.729 | 0.601 | / |
| NetVLAD [47] | 256 | 0.635 | / | 0.735 | / | 0.843 |
| CroW [30] | 256 | 0.684 | 0.637 | 0.765 | 0.691 | 0.851 |
| **DSF** | 256 | **0.741** | **0.640** | **0.842** | **0.736** | **0.844** |
| Neural Codes [32] | 128 | 0.433 | 0.384 | / | / | / |
| NetVLAD [47] | 128 | 0.614 | / | 0.695 | / | 0.826 |
| CroW [30] | 128 | 0.641 | 0.590 | 0.746 | 0.670 | 0.828 |

| | | | | | | |
|---|---|---|---|---|---|---|
| SBA [36] | 128 | 0.648 | 0.587 | 0.765 | 0.685 | 0.830 |
| DSFH [38] | 128 | 0.622 | 0.4764 | 0.607 | 0.7016 | 0.7476 |
| **DSF** | 128 | **0.724** | **0.603** | **0.825** | **0.710** | **0.824** |
| CroW + QE | 128 | 0.670 | 0.641 | 0.793 | 0.728 | / |
| CroW + QE | 256 | 0.718 | 0.676 | 0.815 | 0.753 | / |
| CroW + QE | 512 | 0.749 | 0.706 | 0.848 | 0.794 | / |
| CAMs + QE | 512 | 0.730 | 0.712 | 0.836 | 0.791 | / |
| SBA + QE | 512 | 0.748 | 0.726 | 0.860 | 0.807 | / |
| **DSF + QE** | 128 | **0.772** | **0.671** | **0.868** | **0.759** | / |
| **DSF + QE** | 256 | **0.789** | **0.707** | **0.884** | **0.799** | / |
| **DSF + QE** | 512 | **0.808** | **0.722** | **0.895** | **0.829** | / |

## 5 Conclusion

This paper proposed a new method to combine deep features and low-level features into a compact representation. In the proposed method, a covariance weight is proposed to select some key feature maps, which are utilized to calculate the response-regions weights. It can enhance the discriminativeness of feature maps. A novel method is proposed to extract texture features together with deep features. It improves the positive role of some valuable feature maps in retrieval. Experimental results with five public datasets show that the proposed method outperforms other similar methods.

However, methods of further combining deep and low-level features still need further research. In the future, while maintaining the advantages of this method, we plan to reduce the computational burden of the algorithm and explore the retrieval effects of other neural networks.

## Availability of data and materials

The author has used third party data and therefore do not own the data, and those benchmark datasets are public. The code of the proposed method available on request.

## Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.


## Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61866005.


## Authors' contributions

**Zhou Lu:** Conceptualization, Software, Validation, Writing - Original Draft, Resources, Data Curation. **Guang-Hai Liu:** Methodology, Review & Editing, Supervision, Revision, Funding acquisition, Formal analysis.


## Acknowledgements

The authors would like to thank the anonymous reviewers of IJVP journal for their constructive comments in the revision.

## References


[1] F. Lu, and G.-H. Liu. "Image retrieval using contrastive weight aggregation histograms." Digital Signal Processing 123 (2022): 103457.

[2] B.-J. Zhang, G.-H. Liu, and J.-K. Hu. "Filtering Deep Convolutional Features for Image Retrieval." International Journal of Pattern Recognition and Artificial Intelligence 36(01) (2022): 2252003.

[3] Ji-Zhao Hua, Guang-Hai Liu, Shu-Xiang Song, Content-based image retrieval using color volume histograms, International Journal of Pattern Recognition and Artificial Intelligence, 33(9) (2019)1940010.

[4] C. Singh, E. Walia, K.P. Kaur, Color texture description with novel local binary patterns for effective image retrieval, Pattern Recognition, 76(2017)50- 68.

[5] G.-H. Liu, J.-Y. Yang, Z.Y. Li. Content-based image retrieval using computational visual attention model, Pattern Recognition, 48(8) (2015) 2554- 2566.

[6] G.-H. Liu, Z. Wei, Image Retrieval Using the Fused Perceptual Color Histogram. Computational Intelligence and Neuroscience, vol. 2020, Article ID 8876480, 10 pages, 2020.

[7] M.J. Swain, D.H. Ballard, Color Indexing, International Journal of Computer Vision, 7(1991)11-32.

[8] C-H Lai, A colour image retrieval scheme based on Z-scanning technique[J]. Image Science Journal, 2013, 61(3): 320-333.

[9] R. Haralick, K. Shanmugam, and I. Dinstein, Texture features for image classification. IEEE Trans. on System, Man and Cybernetics, 1973.

[10] G. Cross, A. Jain, Markov random field texture models, IEEE Trans. Pattern Anal. Mach. Intelligence. 5 (1) (1983) 25–39.

[11] B.S. Manjunathi, W.Y. Ma., Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intelligence. 18 (8) (1996) 837–842.

[12] T. Ojala, M. Pietikanen, T. Maenpaa, Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intelligence. 24 (7) (2002) 971–987.

[13] M. Clement, C. Kurtz, L. Wendling, Learning spatial relations and shapes for structural object description and scene recognition, Pattern Recognition, 84(2018)197-210.

[14] B. Hong and S. Soatto, Shape matching using multiscale integral invariants, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(1) (2015)151-160.

[15] J. Žunić, P.L. Rosin, V. Ilić, Disconnectedness: A new moment invariant for multi-component shapes, Pattern Recognition, 78(2018)91-102.

[16] G. Malu, S. Elizabeth, S. M. Koshy, Circular mesh-based shape and margin descriptor for object detection, Pattern Recognition, 84(2018)97-111.

[17] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley & Sons Ltd., New York, 2002.

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

[19] J. Sivic and A. Zisserman, Video Google: A text retrieval approach to object matching in videos. In Proceedings of the International Conference on Computer Vision, October 2003.

[20] D. Nister, H. Stewenius. Scalable recognition with a vocabulary tree, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.

[21] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2169–2178.

[22] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of features image classification, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 490–503.

[23] J. Sivic, A. Zisserman, Efficient visual search of videos cast as text retrieval, IEEE Trans. Pattern Anal. Mach. Intelligence. 31 (4) (2009) 591–606.

[24] H. J´egou, M. Douze, C. Schmid, P. Perez, Aggregating local descriptors into a compact image representation. In: CVPR (2010) 3304-3311.

[25] F. Perronnin, Y. Liu, J. Sanchez, H. Poirier, Large-scale image retrieval with compressed Fisher vectors. In: CVPR (2010) 3384-3391.

[26] A. Krizhevsky, I. Sutskever, G.-E. Hinton, Imagenet classification with deep convolutional neural networks. In: NIPS (2012) 1097-1105.

[27] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations arXiv preprint arXiv:1409.1556, 2015.

[28] A. Babenko, V. Lempitsky, Aggregating local deep features for image retrieval, in: International Conference on Computer Vision, 2015, pp. 1269–1277.

[29] G. Tolias, R. Sicre, H. Jégou, Particular object retrieval with integral max-pool- ing of CNN activations, in: International Conference on Learning Representa- tions, 2016, pp. 1–12.

[30] Y. Kalantidis, C. Mellina, S. Osindero, Cross-dimensional weighting for aggregated deep convolutional features, Proceedings of the European Conference on Computer Vision, Springer, Cham (2016) 685-701.

[31] 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(3) (2016) 251-258.

[32] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, Proceedings of the European Conference on Computer Vision (2013), pp. 584–599.

[33] E. Mohedano, K. McGuinness, E. O'Connor N, et al. Bags of local convolutional features for scalable instance search. Proceedings of the ACM on International Conference on Multimedia Retrieval, ACM (2016), pp. 327-331.

[34] M. Tzelepi, A. Tefas, Deep convolutional learning for content-based image retrieval, Neurocomputing, 275(31) (2018)2467-2478.

[35] H. Zhang, S. Wang, X. Xu, T. W. S. Chow and Q. M. J. Wu, Tree2Vector: Learning a vectoral representation for tree-structured data, IEEE Transactions on Neural Networks and Learning Systems, 29(11) (2018) 5304-5318.

[36] J. Xu, C. Wang, C. Qi, et al, Unsupervised Semantic-Based Aggregation of Deep Convolutional Features, IEEE Transactions on Image Processing, vol.28, no.2, 2019, pp. 601-611.

[37] H. Zhang, Y. Ji, W. Huang, et al. Sitcom-star-based clothing retrieval for video advertising: a deep learning framework, Neural Computing and Applications,31(2019) 7361–7380.

[38] G-H Liu, J-Y Yang. Deep-seated features histogram: A novel image retrieval method. Pattern Recognition, 116 (2021), 107926.

[39] S. Pang, J. Zhu, J. Wang, V. Ordonez, J. Xue. Building discriminative CNN image representations for object retrieval using the replicator equation, Pattern Recognition, 83(2018) 150-160.

[40] S. Pang, J. Ma, J. Xue, J. Zhu and V. Ordonez, Deep Feature Aggregation and Image Re-Ranking With Heat Diffusion for Image Retrieval, IEEE Transactions on Multimedia, 21(6) (2019) 1513-1523.

[41] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[42] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: improving particular object retrieval in large scale image databases, in 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1–8.

[43] M. Douze, H. Jegou, C. Schmid, Hamming embedding and weak geometry consistency for large scale image search, in: Proceedings of the 10th European conference on Computer vision, October 2008.

[44] R.M. Haralick Statistical and structural approaches to texture. Proceedings of the IEEE, 1979, 67(5):786-804.

[45] H. Jégou and O. Chum, Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening, in Proc. 12th Eur. Conf. Computer. Vis., Florence, Italy, Oct. 2012, pp. 774– 787.

[46] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in Proc. 11th IEEE Int. Conf. Computer. Vis., Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[47] R. Arandjelović, P. Gronat, A. Torii, et al NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 40, no. 6, pp. 1437-1451, 2016.

[48] A. Jiménez, J. Alvarez and X. Giro, Class-weighted convolutional features for visual instance search. Proceedings of the 28th British Machine Vision Conference (BMVC), London, (2017) pp. 1-12.

[49] B. Zhang, Q. Wang, X. Lu, F. Wang, P. Li, Locality-constrained affine subspace coding for image classification and retrieval, Pattern Recognition. 100 (2020) 107167.

[50] C. Bai, H. Li, J. Zhang, et al. Unsupervised Adversarial Instance-level Image Retrieval. in *IEEE Transactions on Multimedia*, vol. 23, (2021) pp. 2199-2207.

**Zhou Lu**, He is currently a postgraduate of the School of Computer Science and Engineering, Guangxi Normal University. His current research interests are in the areas of image processing, pattern recognition and artificial intelligence.

**Guang-Hai Liu** received his Ph. D degree from the School of Computer Science and Technology, Nanjing University of Science and Technology (NUST). He is currently a professor with the College of Computer Science and Information Technology, Guangxi Normal University in China. His current research interests are in the areas of image processing, pattern recognition, and artificial intelligence.