mathematics

MDPI

Article

Near-Duplicate Image Detection System Using Coarse-to-Fine Matching Scheme Based on Global and Local CNN Features

Zhili Zhou [1], Kunde Lin [1], Yi Cao [1,*], Ching-Nung Yang [2,*] and Yuling Liu [3]

[1] Jiangsu Engineering Centre of Network Monitoring and School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; zhou_zhili@nuist.edu.cn (Z.Z.); LinKunde@nuist.edu.cn (K.L.)

[2] Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 97047, Taiwan

[3] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410208, China; yuling_liu@hnu.edu.cn

* Correspondence: caoyinuist@163.com (Y.C.); cnyang@gms.ndhu.edu.tw (C.-N.Y.)

Received: 26 February 2020; Accepted: 17 April 2020; Published: 22 April 2020

Abstract: Due to the great success of convolutional neural networks (CNNs) in the area of computer vision, the existing methods tend to match the global or local CNN features between images for near-duplicate image detection. However, global CNN features are not robust enough to combat background clutter and partial occlusion, while local CNN features lead to high computational complexity in the step of feature matching. To achieve high efficiency while maintaining good accuracy, we propose a coarse-to-fine feature matching scheme using both global and local CNN features for real-time near-duplicate image detection. In the coarse matching stage, we implement the sum-pooling operation on convolutional feature maps (CFMs) to generate the global CNN features, and match these global CNN features between a given query image and database images to efficiently filter most of irrelevant images of the query. In the fine matching stage, the local CNN features are extracted by using maximum values of the CFMs and the saliency map generated by the graph-based visual saliency detection (GBVS) algorithm. These local CNN features are then matched between images to detect the near-duplicate versions of the query. Experimental results demonstrate that our proposed method not only achieves a real-time detection, but also provides higher accuracy than the state-of-the-art methods.

Keywords: convolutional feature maps; deep convolutional neural network; CNN features; sum-pooling; near-duplicate image detection; digital forensics

1. Introduction

With the rapid development of Internet technology and the increasing popularity of mobile devices, it is very easy for users to capture, transmit and share images through the networks. In these image data, near-duplicate images occupy a significant proportion. The task of near-duplicate image detection is to efficiently and effectively detect near-duplicate versions of a given query image from a large-scale image database. Near-duplicate image detection has been successfully applied in many applications, such as image copyright protection [1–3], coverless information hiding [4–6], secret image sharing [7] and redundancy elimination [8].

In recent years, deep learning techniques such as convolutional neural networks (CNNs) have received extensive attention in the area of computer vision [9,10]. In view of this fact, some researchers tend to use the features extracted from CNNs instead of hand-crafted features for the tasks of

near-duplicate image detection or content-based image retrieval [11–17]. In literature, it has been proven that CNN-based features achieve superior performance than the traditional hand-crafted features. In general, the existing CNN-based features can be divided into two categories: global CNN features and local CNN features.

The global CNN features are extracted by feeding the whole region of an image into a pretrained CNN model and pooling the outputs of the intermediate layers such as convolutional layers or fully connected layers. The most typical pooling methods used for global CNN feature extraction include max-pooling [18–28], sum-pooling [29–32], and average-pooling [33]. Researchers have proposed some improved versions of these pooling methods to extract the global CNN features, such as centering prior-based sum-pooled convolution (SPoC) feature [29], the cross-dimensional weighting (CroW) feature [30], and the regional maximum activations of convolution (R-MAC) [18].

Recently, some researchers have focused on training deep learning models to extract image features [34–37]. However, since these methods fail to sufficiently consider the influence of background clutter and partial occlusion on the final representation, the extracted global CNN features are not robust enough to combat these attacks.

Instead of capturing the characteristics of the whole image region, the local CNN features [19–22,38–42] characterize the local image regions. Generally, similar to the traditional hand-crafted local features such as scale-invariant feature transform (SIFT) [43], the extraction of local CNN features consists of two stages: image region detection and descriptor generation. A number of image regions are first detected from each image, and the local descriptors are generated from the outputs of intermediate layers of a pretrained CNN model within the regions. In [44], the local CNN features were proven to perform better than the traditional hand-crafted local features in many image retrieval/detection tasks. However, since a large number of local regions are usually detected from each image, the extraction and matching of local features usually have high computational complexity. To reduce the computational complexity, local features are usually integrated into a single image representation using a variety of integration methods such as bag of words (BOW) model [45], fisher vector (FV) [46], and vector of locally aggregated descriptors (VLAD) [47] for near-duplicate image detection. However, the integration process causes a lot of important information to be lost and thus decreases the detection accuracy significantly.

In summary, although the extraction and matching of global CNN features are computationally efficient, the global CNN features suffer from the robustness problem in near-duplicate image detection. On the contrary, the local CNN features achieve higher robustness, but the matching of local CNN features between images has high computational complexity since a large number of local CNN features are extracted from each image.

In order to exploit the advantages of both global features and local features, we propose a coarse-to-fine feature matching scheme using both global and local CNN features for near-duplicate image detection. The main contributions of our method are summarized as follows:

(1) A coarse-to-fine feature matching scheme is proposed. The proposed coarse-to-fine feature matching scheme consists of a coarse matching stage and a fine matching stage. In the coarse matching stage, we match the global CNN features between a given query image and database images to filter most of irrelevant images of the query from an image database. In the fine matching stage, we extract and match the local CNN features between images to find the near-duplicate versions of the query. The proposed coarse-to-fine feature matching scheme allows a real-time and accurate near-duplicate image detection. Thus, it has important significance in practical applications of content-based image detection/retrieval.

(2) The saliency map-based local CNN features are extracted. In the tasks of facial expression recognition and image classification, the introduction of attention mechanisms leads to the significant improvements [48–51]. Motivated by these works, after the global CNN feature matching, we detect the saliency map by the graph-based visual saliency detection (GBVS) algorithm [52] and extract the local CNN features from the local regions surrounding the maximum values of the saliency map. The extracted local CNN features not only have high robustness to background clutter and partial

occlusion, but also achieve high repeatability due to the good stability of the maximum values of the saliency map. Consequently, in the fine feature matching stage, the local CNN feature matching further improves the accuracy of near-duplicate image detection.

The rest of this paper is organized as follows. Section 2 introduces the related works. The details of the proposed detection method are presented in the Section 3. Section 4 displays and analyzes the experimental results. Finally, conclusions are drawn in Section 5.

2. Related Works

With the increasing popularity of CNNs, the recent near-duplicate image detection methods tend to use the features extracted from pre-trained CNN models instead of the traditional hand-crafted features. The existing CNN-based features can be roughly categorized into global CNN features and local CNN features.

The global CNN features are usually extracted by feeding the whole region of an image into a pre-trained CNN model and then pooling the outputs of the intermediate layers such as convolutional layers and fully connected layers. In literature, the popular pooling methods include max-pooling [18–28], sum-pooling [29–32], and average-pooling [33]. Generally, the outputs of a convolutional layer are a set of convolutional feature maps (CFMs), and the global CNN features are extracted by implementing a pooling operation on the CFMs. The max-pooling method computes the maximum value of each CFM and concatenates all the maximum values to form the global CNN features, while sum-pooling and average-pooling methods compute the sum and the average value of each CFM, respectively. To improve the performance of the extracted global CNN features on near-duplicate image detection, researchers have proposed some improved versions of these pooling methods to extract the global CNN features. Babenko et al. [29] proposed the SPoC descriptor, which is generated by an improved version of the sum-pooling, i.e., centering prior-based sum-pooling. In particular, instead of directly computing the sum of all activations of each CFM, SPoC [29] is extracted by performing centering prior-based sum-pooling on the CFMs, where the activations near to the center of feature maps are assigned larger weighting coefficients. Kalantidis et al. [30] generated CroW by computing the weighted sum values on the CFMs, where the weights of detected interest regions are set to be larger, while the weights of other regions are set to lower. Tolias et al. [18] proposed an aggregation method based on variable sliding windows to generate the global CNN feature, i.e., R-MAC, where the max-pooling operation is implemented to aggregate activations of CFMs within sliding windows. The above global feature extraction methods have improved the performance of near-duplicate image detection to some extent.

Recently, some researchers have focused on training deep learning models to extract global image features. Lia et al. [34] evaluated a set of CNN-learned descriptors and concluded that the features learned from fine-tuned CNNs perform better than the off-the-shelf features. Shervin et al. [35] gave a summary of promising works that use deep learning-based models for biometric recognition. Shervin et al. [36] identified mild traumatic brain injury patients by combining a bag of adversarial features (BAF) and unsupervised feature learning techniques. Zhang et al. [37] learned a general straightforward similarity function from raw image pairs for near-duplicate image detection.

However, since those global CNN features are extracted from the whole image region, they show weak robustness in regards to the background clutter and partial occlusion, which negatively influence on the performance of near-duplicate image detection.

In order to address the problem of weak robustness, one possible solution is to extract local CNN features for near-duplicate image detection. In literature, a variety of local CNN features have been proposed. Generally, similar to the traditional hand-crafted local features, the extraction of local CNN features consists of two steps: region detection and descriptor generation. In the image region detection, there are three kinds of popular regions: the image patches, interest point-based regions, and the object region proposals. In the descriptor generation stage, by feeding a given image into a pretrained CNN model, the local CNN features are extracted from the outputs of convolutional layers or fully connected

layers within each image region. Gong et al. [38] detected local image patches by adopting a multi-scale sliding window strategy on CFMs, and then concatenated the local CNN features extracted from all the image patches. In contrast from the R-MAC, the feature extraction is implemented at a multi-scale level. Razavian et al. [19] divided images into a set of patches at the multi-scale level, the union of which covers the whole image, for local CNN feature extraction. Zagoruyko et al. [20] detected the regions surrounding the difference of Gaussian (DOG) feature points, while Fischer et al. [44] detected the maximally stable extremal regions (MSER). Mopuri et al. [21] extracted image patches using selective search [39]. In [40], Uricchio et al. utilized the EdgeBox algorithm proposed by [41] for region generation. By using the edge information, the EdgeBox first determines the number of contours in image boxes and the number of edges that overlap the edge of the boxes to score these boxes for generation of object region proposals. Salvador et al. [22] located the potential object regions in an image by employing the region proposal network (RPN) [42]. Besides these region detection algorithms, attention mechanisms have been introduced to capture local characteristics in image classification and facial expression recognition tasks [48,49,51]. Assaf et al. [48] improved the classical Capsule Network (CapsNet) architecture by embedding the self-attention module between the convolutional layers and the primary CapsNet layers for image classification. Shervin et al. [49] proposed the spatial transformer network [50] to detect important face parts for facial expression recognition. Wang et al. [51] built the residual attention network by stacking multiple attention modules within the feed forward network architecture for image classification. Since the local CNN features are extracted at the region-level and some regions still survive after the attacks of background clutter and partial-occlusion, the local CNN features show much higher robustness than the global CNN features. However, due to the large number of local features, the extraction and matching of these local features is very time-consuming, which leads to the limited efficiency for near-duplicate image detection. Although some aggregation methods, such as the bag of words (BOW) model [45], fisher vector (FV) [46], and the vector of locally aggregated descriptors (VLAD) [47] integrate local features into a single image representation to improve the efficiency, the detection accuracy will decrease significantly due to the information loss caused by the aggregation process. In summary, it is hard to directly use these CNN features to achieve a real-time and accurate near-duplicate image detection.

According to the above, there is still a lot of room for improvement in the performance of near-duplicate image detection. To achieve a real-time and accurate near-duplicate image detection, we attempt to take the advantages of both global and local CNN features. In this paper, we propose a coarse-to-fine matching scheme using global and local CNN features for near-duplicate image detection. In the coarse matching stage, we implement the sum-pooling operation on whole region of each image to extract global features and then match them between images. Since only a single global CNN feature is extracted from each image, the coarse matching stage can efficiently filter most of the irrelevant images of a given query image to narrow the search scope largely. In the fine matching stage, motivated by the attention-based image classification and facial expression recognition works [48,49,51], we match the robust and stable local CNN features that are extracted from the regions surrounding the maximum values of CFMs and the saliency map generated by the graph-based visual saliency detection (GBVS) algorithm [52]. Consequently, the proposed approach can effectively and efficiently detect near-duplicate images of a given query from image databases.

3. The Proposed Method

In this section, we introduce the proposed near-duplicate image detection approach in detail. Figure 1 shows the framework of the proposed approach. As shown in Figure 1, the proposed approach consists of two main components, which are the coarse matching stage and the fine matching stage, respectively.

We first generate convolutional feature maps (CFMs) by feeding images into a pre-trained CNN model. Then, in Section 3.2, we extract global features from each image using sum-pooling operation and then match these features between images to obtain the candidate images of a given query from an image database. Finally, in Section 3.3, local CNN features of the query image and candidate images

are extracted and then matched to further detect the near-duplicate versions of the query. The details are given below.

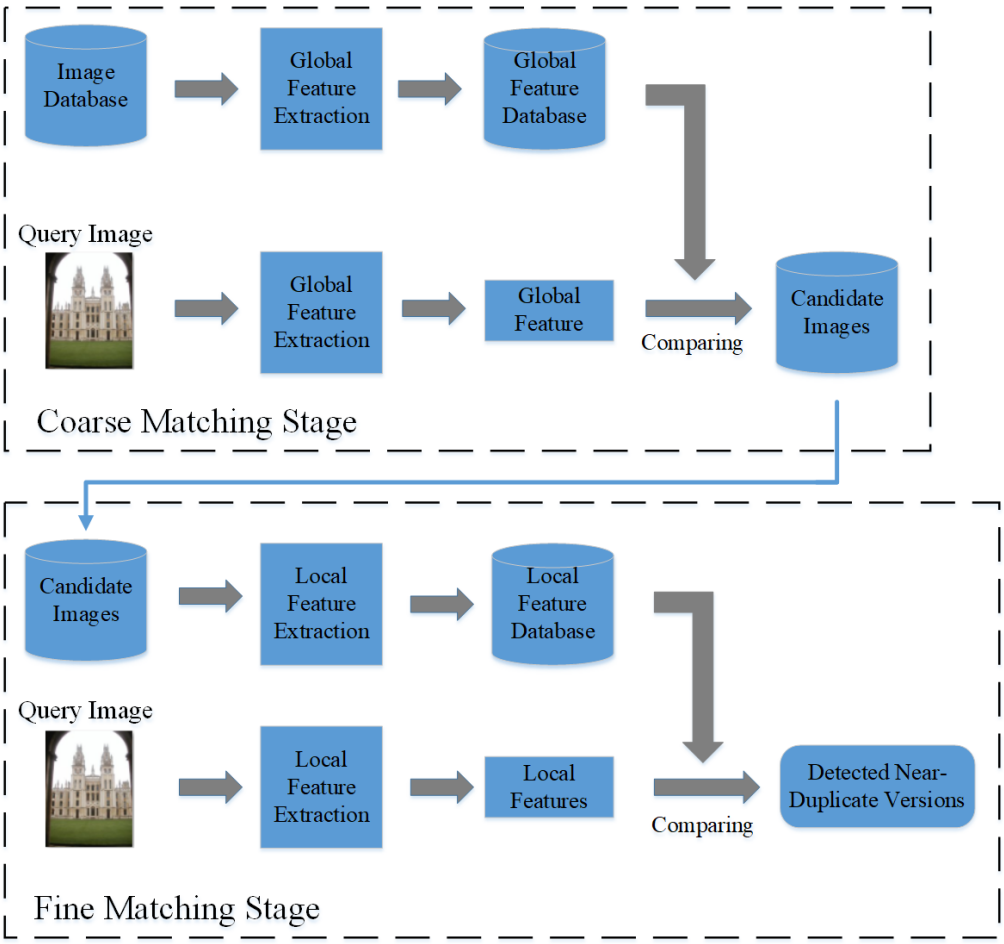

Figure 1. The framework of the proposed near-duplicate image detection system.

3.1. CFM Generation

According to [25], a CNN model is composed of a set of layers including convolutional layers and fully connected layers. In the early years, researchers employed the outputs of fully connected layers to generate image representations. However, some research [23,24,29,44,53] indicates that the features extracted from convolutional layers, especially the last convolutional layer, show better performance than the features extracted from fully connected layers, where the output of a convolutional layer is a set of feature maps, i.e., CFMs. In our approach, we feed each image to a pretrained CNN model, and use the output of the last convolutional layer for feature extraction. Note that we test the performances of our method when using different CNN models in the experimental part. To obtain a good trade-off between accuracy and efficiency, we adopt AlexNet [25] as the pretrained CNN model in our method. Figure 2 shows the 256 CFMs generated from the last convolutional layer after feeding an image into the AlexNet model, where the sizes of CFMs are proportional to the size of the original image.

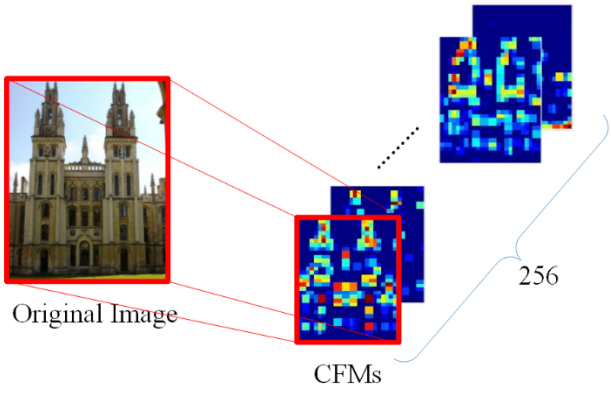

Figure 2. The outputs of the fifth convolutional layer of the AlexNet model, i.e., a set of convolutional feature maps (CFMs), the sizes of which are proportional to the size of the original image.

3.2. Coarse Matching Stage

We will encounter the problem of high computational complexity if the local CNN features are directly extracted and matched between images to detect near-duplicate images from an image database, which usually consists of thousands of images. Due to the high efficiency of global feature extraction and matching, we first extract and match global features to implement a coarse feature matching to efficiently filter most irrelevant images of a given query.

3.2.1. The Extraction of Global CNN Feature

According to [29], the global CNN features generated by sum-pooling of CFMs performs not only better than traditional hand-crafted features, but also better than those generated by max-pooling of CFMs. Thus, we adopt sum-pooling operation on the CFMs to extract the global features.

For a given image I_i, we feed it into the pretrained AlexNet model and collect the output of the fifth convolutional layer to form a set of CFMs, denoted as $MS_i = \left\{M_i^1, M_i^2, \ldots, M_i^k, \ldots, M_i^M\right\}$, where $1 \le k \le M$ and $M = 256$. For each CFM, i.e., M_i^k, its size and activations are denoted as $W \times H$ and $F_i^k = \left\{f_i^k(x,y) \middle| 1 \le x \le W, 1 \le y \le H\right\}$, respectively. Subsequently, we use Equation (1) to extract global features by sum-pooling operation:

$$\phi\left(F_i^k\right) = \sum_{y=1}^{H}\sum_{x=1}^{W} f_i^k(x,y) \tag{1}$$

After this, we concatenate all of these feature values to obtain a 256-dimensional feature vector $\partial(I_i) = \left(\phi\left(F_i^1\right), \phi\left(F_i^2\right), \ldots, \phi\left(F_i^k\right), \ldots, \phi\left(F_i^M\right)\right)$, and normalize the feature vector as $V(I_i) = \frac{\partial(I_i)}{\|\partial(I_i)\|_2}$.

3.2.2. Global Feature Matching

For a given query image I_q and a database image I_d, we can obtain two corresponding normalized feature vectors $V\left(I_q\right)$ and $V(I_d)$ by the above feature extraction process. Then, we employ Equation (2) to compute the inner product of the two feature vectors to measure the global similarity between the two images.

$$SIM\left(I_q, I_d\right) = \left\langle V\left(I_q\right), V(I_d)\right\rangle \tag{2}$$

After computing the similarity, we sort all similarity values in descending order $\left\{SIM_1, SIM_2, \ldots, SIM_{N_D}\right\}$, where $SIM_1 > SIM_2 > \ldots > SIM_n > SIM_{n+1} > \ldots > SIM_{N_D}$ and N_D means the number of database images. In our method, we only keep $N_{Top} = 1000$ detected images of the query as its candidate images, and remove the others.

3.3. Fine Matching Stage

Due to the weak robustness of global CNN features, the detection accuracy of the coarse matching stage is limited. Therefore, in the fine matching stage, we extract and match local features between images to further increase the detection accuracy. It is worth noting that, since only a small number of candidate images need to be verified to confirm whether they are the near-duplicate versions of the query, the efficiency of the fine matching is relatively high.

3.3.1. Central Cropping

Due to the fact that the target objects tend to be located near to the geometrical center of an image, we propose a central cropping strategy on CFMs to reduce the influence of irrelevant background before local feature extraction. As the feature extraction is based on CFMs, we implement the central cropping on the CFMs. In the cropping process, we make the sizes of the cropped CFMs proportional to the sizes of the original CFMs. Denote the ratio between the area of each cropped CFM and that of each original CFM as α. If the area of an original CFM is $S_{CFM} = W \times H$, the area of a cropped CFM is $S'_{CFM} = \alpha \times S_{CFM}$, where $0 < \alpha < 1$. Thus, the width and height of the cropped CFM are $W' = \sqrt{\alpha} \times W$ and $H' = \sqrt{\alpha} \times H$, respectively. To generate the cropped CFM, we set the coordinates of the central point of the cropped CFM by

$$\begin{cases} x_c = \left\lfloor \frac{W}{2} \right\rfloor + 1 \\ y_c = \left\lfloor \frac{H}{2} \right\rfloor + 1 \end{cases} \tag{3}$$

Thus, the cropped CFM can be denoted as $R = [(x_c, y_c), W', H']$, which will be used for local region detection.

3.3.2. Local Region Detection

Since the maximum activations of a CFM are stable and their surrounding regions contain rich information, we detect the points with maximum activations, i.e., maximum points on the cropped CFMs, and then generate the regions surrounding these maximum points for local CNN feature extraction. Figure 3 shows the flowchart of local region generation on cropped CFMs. For a CFM M_i^k, we select the maximum value among all the activations by

$$\mathrm{m}_i^k = \max\left\{ f_i^k(x, y) \middle| 1 \le x \le W, 1 \le y \le H \right\} \tag{4}$$

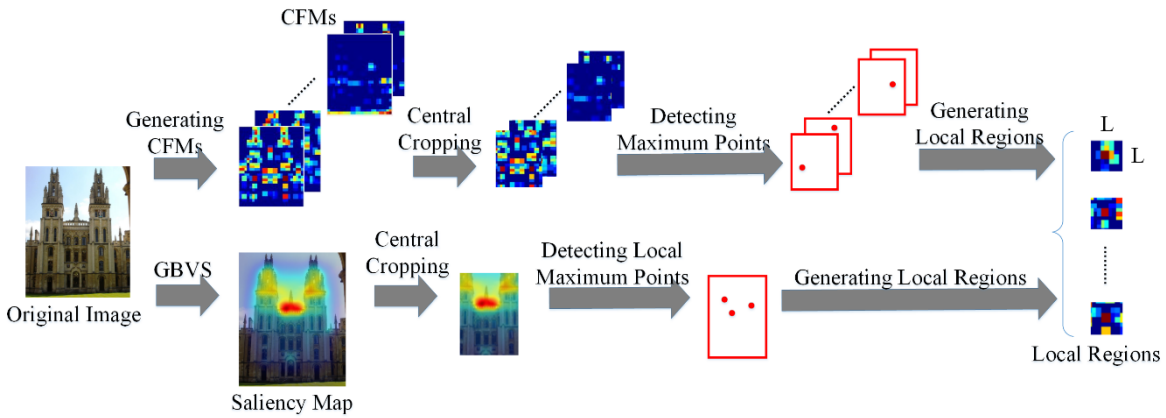

Figure 3. The flowchart of local region generation.

In addition, we generate the saliency map for local feature extraction. In our method, we generate the saliency map by the famous graph-based visual saliency detection (GBVS) algorithm [52].

Then, we also apply central cropping on the saliency map, and detect the local maximum values of the saliency map, as illustrated in Figure 3.

Next, by setting each detected point as a center, the patch surrounding the point is used as the local region, where the side length of the patch is denoted as L. Suppose N_P maximum points are generated from the CFMs in total. Thus, N_P corresponding local regions are generated. To reduce storage memory and computational complexity in feature extraction and matching, we will not use all the regions generated from CFMs for feature extraction. Instead, we sort these regions in descending order according to the activation values of the corresponding maximum points, and select the first N regions for local feature extraction. Since the number of local maximums of the saliency map is limited, we use all the regions generated from the saliency map for local feature extraction.

3.3.3. Local Features Extraction and Matching

After generating a set of local regions, we extract a 256-dimensional feature vector by sum-pooling the activations of CFMs within each local region rather than the whole image region, and then normalize it. Thus, for a given image I_i, we can extract a set of 256-dimensional normalized local feature vectors $VS_{I_i} = \{V_1(I_i), V_2(I_i), \ldots, V_M(I_i)\}$, the number of which is denoted as M. Next, we match these local features between images. For a query image I_q and a candidate image I_c, by the above local feature extraction method, we can obtain their M 256-dimensional local feature vectors, denoted as $VS_q = \left\{V_1\left(I_q\right), V_2\left(I_q\right), \ldots, V_M\left(I_q\right)\right\}$ and $VS_c = \{V_1(I_c), V_2(I_c), \ldots, V_M(I_c)\}$, respectively. Then, we sequentially compare each pair of feature vectors to compute their similarity by inner products. By comparing a query feature vector $V_k\left(I_q\right)$ to each feature vector $V_j\left(I_q\right)$ in VS_c, where $1 \le j \le M$, we can obtain M similarity scores and then select the maximum score as the matching score of $V_k\left(I_q\right)$.

$$\max\left\{\left\langle V_k\left(I_q\right), V_j(I_c)\right\rangle \middle| j = 1, 2, \ldots, M\right\} \tag{5}$$

Thus, there are M matching scores in total. Next, we sum up all the matching scores as the final similarity between the query image I_q and the candidate image I_c by Equation (6).

$$SIM\left(I_q, I_c\right) = \sum_{k=1}^{M} \max\left\{\left\langle V_k\left(I_q\right), V_j(I_c)\right\rangle \middle| j = 1, 2, \ldots, M\right\} \tag{6}$$

Finally, we compare the similarity score to a pre-set threshold to determine whether the candidate image I_c is a near-duplicate version of the query I_q.

4. Experiments

In this section, we first introduce the public datasets and the evaluation criteria used in our method. Second, for the three parameters of our method, i.e., the cropping ratio α between areas of cropped CFMs and original CFMs, the maximum number of regions N, and the side length of regions L, we determine the parameter settings to achieve the optimal performance of our proposed method. Third, we measure the detection performance of the proposed method and compare it with those of its two versions, which separately use global CNN features or local CNN features, as well as the state-of-the-art features.

4.1. Datasets and Evaluation Criteria

In this experiment, we adopt three near-duplicate image detection datasets, i.e., the Oxford5k dataset [54], the Holidays dataset [55], and the Paris6k dataset [56]. The Oxford5k dataset consists of 5062 pictures of Oxford buildings collected from the Flickr website. These images have been manually labeled as one of 11 different landmarks, each of which contains five query images. The Holidays dataset contains a total of 1491 pictures, which are divided into 500 groups. Each group of images

contains a specific object or scene captured by different viewpoints. The first image of each group is used as the query image. The Paris6k dataset is composed of 6412 pictures of Paris buildings from the Flickr website. There are 500 query images in total. The above three public datasets are used to test the detection performance of different methods. We use the mAP value, which represents the average detection accuracy at different recall rates, to measure the detection accuracy. In addition, we adopt the average query time to test the detection efficiency.

4.2. Parameter Determination

In this subsection, we observe the effects of the three parameters of the proposed method, and then find the proper parameter settings for the proposed method. The three important parameters are α, N and L, representing the cropping ratio between areas of cropped CFMs and CFMs, the maximum number of regions, and the side length of regions, respectively. We implement the experiment on the Oxford5k dataset. We first fix parameters L and N to the default values, 3 and 100, respectively, to test the impact of the parameter α in the aspects of the accuracy and efficiency. The effects of α are illustrated in Figures 4 and 5. From Figure 4, we can clearly observe that a larger α is helpful for performance improvement, because larger cropped CFMs contain more crucial content. However, increasing α does not consistently improve the performance. This might be because larger cropped CFMs ($\alpha > 0.5$) also contain more irrelevant background clutter, which would introduce more noises in the image features. From Figure 5, it is clear that the increase of α leads to the increase of detection time, because more features will be extracted from larger cropped CFMs. Therefore, to find a good trade-off between accuracy and efficiency, we set the parameter α as 0.5, which provides good accuracy and high efficiency.

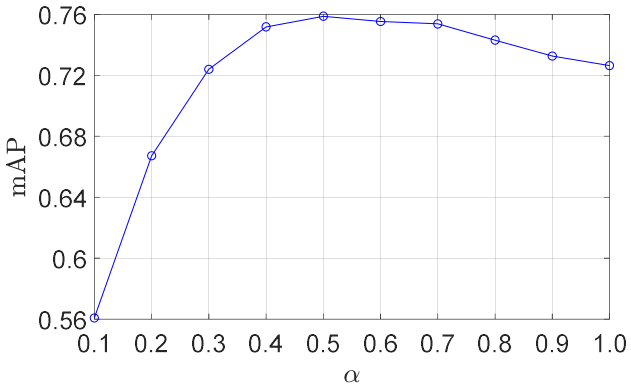

Figure 4. The effect of α on the detection accuracy.

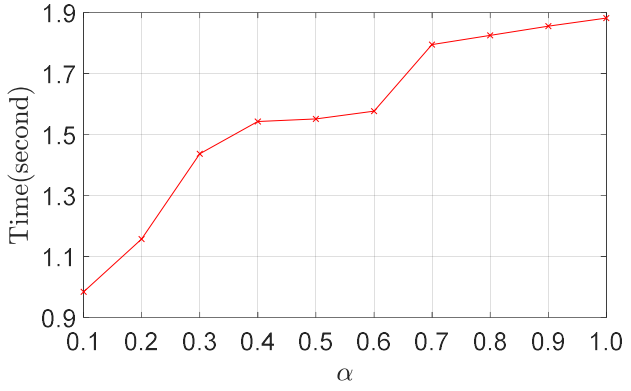

Figure 5. The effect of α on the detection efficiency.

The effects of N on the detection accuracy and efficiency are illustrated in Figures 6 and 7, respectively. It can be clearly observed that a larger N is helpful for performance improvement. However, when N is too large, it becomes very likely that many irrelevant local regions are used for feature extraction. Thus, we set the parameter N as 100 for the following experiments.

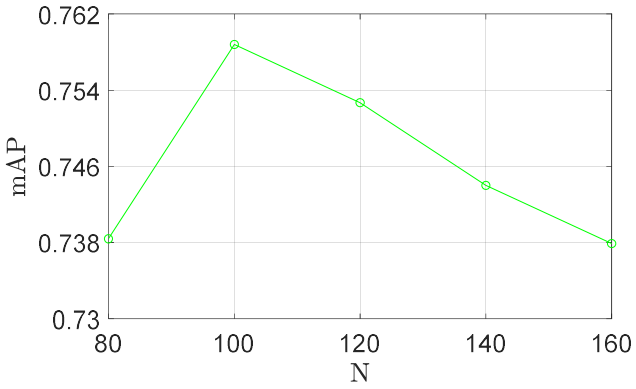

Figure 6. The effect of N on the detection accuracy.

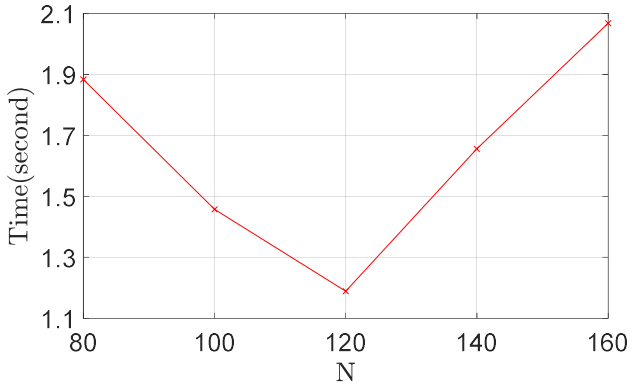

Figure 7. The effect of N on the detection efficiency.

The effects of L are illustrated in Figures 8 and 9. It is clear that the detection performance degrades if L is too large or too small. That is because a smaller L results in smaller local regions, which contain insufficient visual content; If the area of local regions is too large, these regions would be sensitive to background clutter and partial occlusion. To find a good trade-off between accuracy and efficiency, we set the value of L as five.

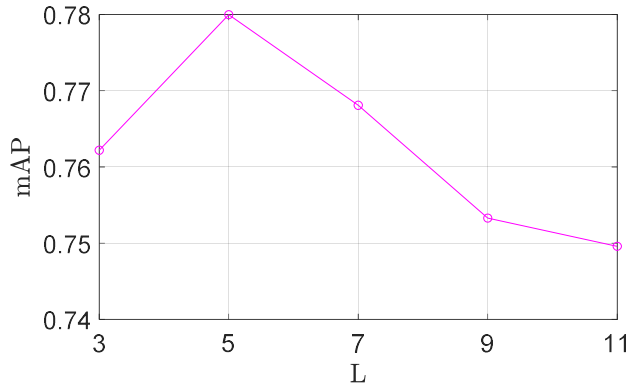

Figure 8. The effect of L on the detection accuracy.

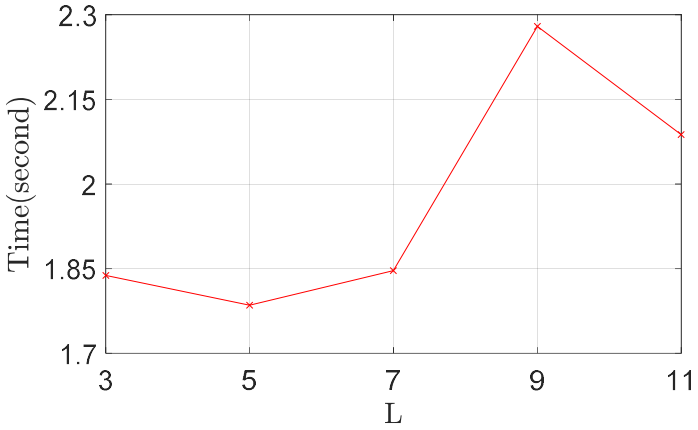

Figure 9. The effect of L on the detection efficiency.

In the proposed method, we adopt the optimal settings of these three parameters, i.e., 0.5 for α, 100 for N and 5 for L in the following experiments.

4.3. Performance When Using Different Pre-Trained Networks

In the experiments, we also test the performances of our method when using different kinds of pre-trained networks to observe their impacts on our method. We chose four famous convolutional neural networks including AlexNet [25], VGG16 [57], VGG19 [57], and ResNet-18 [58] to implement the test, where the last convolutional layers, or ReLu layers, of these networks are adopted. Table 1 shows the mAP values and time costs of our method when using different pre-trained CNNs on Oxford5k dataset.

Table 1. Performances of our method when using different pre-trained convolutional neural networks (CNNs) on Oxford5k.

	Layer	Number of Feature Maps	mAP	Time (Second)
AlexNet	Conv5_3	256	0.715	**0.2253**
Vgg16	Relu5_3	512	**0.728**	1.7124
Vgg19	Relu5_4	512	0.719	1.9412
ResNet	Conv5_x	512	0.725	1.9728

In Table 1, it is clearly observed that the detection accuracy when using Vgg16, Vgg19, and ResNet-18 is slightly higher than that when using AlexNet. However, AlexNet leads to much higher time efficiency than the other networks, due to the fewer feature maps needed to be processed for feature extraction. Thus, to find a good trade-off between accuracy and efficiency, we chose AlexNet in our method.

4.4. Performance Comparison

After selecting the parameters, we use Oxford5k as the baseline dataset to compare the detection performances between our method and its two other versions in the aspects of detection accuracy, average time cost, and average memory consumption.

The two versions are the methods that separately use the extracted global CNN features or local CNN features. In Table 2, the "global CNN features" and the "local CNN features" denote the methods using the extracted global CNN features and local CNN features, respectively. The "Time" means the average time cost for a query image, while the "Memory" represents the average memory required to store the features of an image.

Table 2. Comparison between the proposed method and its two versions on Oxford5k.

	Global CNN Features	Local CNN Features	The Proposed Method
mAP	0.5271	**0.780**	0.715
Time (seconds)	0.0376	1.6259	**0.2253**
Memory (bytes)	256		

As shown in Table 2, the proposed method achieves a significant improvement in detection accuracy compared to the method only using global CNN features, and it has much higher detection efficiency compared to the method only using local CNN features. Moreover, since our method stores two types of CNN features, the memory consumption of the proposed method is higher than that of the two other methods. However, its total memory is slightly higher than that of the method using CNN local features, since the number of candidate images has been greatly reduced by the coarse feature matching. Additionally, the detection accuracy of the proposed method is comparable to that of the method only using local CNN features. That is because most of potential near-duplicate versions of a given query are kept by the coarse feature matching.

Also, we compare our method with five state-of-the-art methods: max-pooling [18], VLAD-CNN [24], SPoC [29], R-MAC [18], and CroW [30]. Table 3 shows the mAP values of these methods on three different datasets. From Table 3, it is clear that our method achieves higher detection accuracy than all of those methods on Oxford5k and Holidays. The detection accuracy of our method is only slightly lower than that of R-MAC on Paris6k. Overall, the proposed method generally outperforms these state-of-the-art methods.

Table 3. Comparison of detection accuracy between our method and the state-of-the-art methods on three different datasets.

Methods	Oxford5k (mAP)	Holidays (mAP)	Paris6k (mAP)
max-pooling [18]	0.524	0.711	——
VLAD-CNN [24]	0.558	0.836	0.583
SPoC [29]	0.589	0.802	——
R-MAC [18]	0.669	0.852	0.830
CroW [30]	0.684	0.851	0.765
Ours	0.715	0.886	0.772

Figure 10 shows some examples of detection results of the proposed method on Oxford5k. In summary, our method achieves high detection efficiency and meets real-time detection demand, while maintaining good accuracy in the task of near-duplicate image detection.

Figure 10. Several examples of detection results of the proposed method. The four queries are listed in the first column, and the corresponding top four ranked detection results of each query are shown in the following columns.

5. Conclusions

We presented a coarse-to-fine feature matching scheme using both global feature and local feature for near-duplicate image detection. By exploiting the advantages of both global and local CNN features, the proposed method can achieve real-time and accurate near-duplicate image detection. In the coarse matching stage, we extract global features to quickly filter most of the irrelevant images. In the following fine matching stage, we detect CFMs and use the saliency map to extract and match the proposed local CNN features to obtain the final detection results. The experimental results show our method achieves desirable performances in both accuracy and efficiency, which makes it appealing for practical applications of content-based image detection/retrieval tasks.

In our method, we directly use the pre-trained CNNs for near-duplicate image detection, but these CNNs are originally designed for image classification. Thus, it might be more effective to adopt the transfer learning methods to generate a fine-tuned CNN model for near-duplicate image detection. Additionally, the saliency map is generated by an unsupervised method to locate potential object regions for local feature extraction. In future work, we will study the supervised object recognition methods to accurately locate the object regions for local feature extraction to further improve the detection performance.

Author Contributions: Conceptualization, Y.C. and C.-N.Y.; Methodology, Y.C.; Software, K.L. and Y.L.; Validation, K.L., Y.C. and Z.Z.; Formal analysis, Y.C.; Investigation, K.L.; Resources, Y.L.; Data curation, K.L. and Z.Z.; Writing—Original draft preparation, Z.Z.; Writing-Review and editing, K.L., Y.C., C.-N.Y. and Y.L.; Visualization, K.L. and Z.Z.; Supervision, Z.Z.; Project administration, Z.Z.; Funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded in part by the National Natural Science Foundation of China under Grant 61972205, 61602253, U1836208, U1836110, 61872134, in part by the National Key R&D Program of China under Grant 2018YFB1003205, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China, and in part by Ministry of Science and Technology (MOST) under contracts 1092634-F-259-001-through Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, Z.; Wang, Y.; Wu, Q.M.J.; Yang, C.N.; Sun, X. Effective and efficient global context verification for image copy detection. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 48–63.
2. Zhou, Z.; Yang, C.N.; Chen, B.; Sun, X.; Liu, Q.; Wu, Q.M.J. Effective and efficient image copy detection with resistance to arbitrary rotation. *IEICE Trans. Inf. Syst.* **2016**, *99*, 1531–1540. [CrossRef]
3. Zhou, Z.; Wu, Q.M.J.; Yang, Y.; Sun, X. Region-level visual consistency verification for large-scale partial-duplicate image search. *ACM Trans. Multimedia Comput. Commun. Appl.* **2020**. [CrossRef]
4. Cao, Y.; Zhou, Z.; Sun, X.; Gao, C. Coverless information hiding based on the molecular structure images of material. Comput. *Mater. Continua* **2018**, *54*, 197–207.
5. Zhou, Z.; Cao, Y.; Wang, M.; Fan, E.; Wu, Q.M.J. Faster-RCNN based robust coverless information hiding system in Cloud Environment. *IEEE Access.* **2019**, *7*, 179891–179897.
6. Cao, Y.; Zhou, Z.; Yang, C.N.; Sun, X. Dynamic content selection framework applied to coverless information hiding. *J. Int. Technol.* **2018**, *4*, 1179–1186.
7. Liu, Y.; Yang, C.N.; Wu, C.; Sun, Q.; Bi, W. Threshold changeable secret image sharing scheme based on interpolation polynomial. *Multimedia Tools Appl.* **2019**, *13*, 18653–18667.
8. Zhou, Z.; Wu, Q.M.J.; Huang, F.; Sun, X. Fast and accurate near-duplicate image elimination for visual sensor networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717694172.
9. Yang, Y.; Wu, Q.M.J.; Feng, X.; Akilan, T. Recomputation of dense layers for the performance improvement of DCNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]
10. Yang, Y.; Wu, Q.M.J. Features combined from hundreds of mid-layers: Hierarchical networks with subnetwork nodes. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3313–3325. [CrossRef]
11. Lin, G.; Liu, B.; Xiao, P.; Lei, M.; Bi, W. Phishing detection with image retrieval based on improved texton correlation descriptor. *Comput. Mater. Continua* **2018**, *57*, 533–547. [CrossRef]
12. Xu, W.; Xiang, S.; Sachnev, V. A cryptograph domain image retrieval method based on Paillier Homomorphic block encryption. *Comput. Mater. Continua* **2018**, *55*, 11–21.
13. Zheng, L.; Song, C. Fast near-duplicate image detection in Riemannian space by a novel hashing scheme. *Comput. Mater. Continua* **2018**, *56*, 529–539.
14. Zhou, Z.; Mu, Y.; Wu, Q.M.J. Coverless image steganography using partial-duplicate image retrieval. *Soft Comput.* **2019**, *23*, 4927–4938. [CrossRef]
15. Zhou, Z.; Wu, Q.M.J.; Wan, S.; Sun, W.; Sun, X. Integrating SIFT and CNN feature matching for partial-duplicate image detection. *IEEE Trans. Emerging Top. Comput. Intell.* **2014**, *23*, 3368–3380. [CrossRef]
16. Zhou, Z.; Wu, Q.M.J.; Sun, X. Multiple distance-based coding: Toward scalable feature matching for large-scale web image search. *IEEE Trans. Big Data* **2019**. [CrossRef]
17. Zhou, Z.; Wu, Q.M.J.; Sun, X. Encoding multiple contextual clues for partial-duplicate image retrieval. *Pattern Recognit. Lett.* **2018**, *109*, 18–26. [CrossRef]
18. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
19. Sharif Razavian, A.; Azizpour, H.; Sullivan, J. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
20. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
21. Reddy Mopuri, K.; Venkatesh Babu, R. Object level deep feature pooling for compact image representation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 62–70.
22. Salvador, A.; Giró-i-Nieto, X.; Marqués, F.; Satoh, S.I. Faster r-cnn features for instance search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 9–16.
23. Razavian, A.S.; Sullivan, J.; Carlsson, S.; Maki, A. Visual instance retrieval with deep convolutional networks. *ITE Trans. Media Technol. Appl.* **2016**, *4*, 251–258. [CrossRef]

24. Yue-Hei Ng, J.; Yang, F.; Davis, L.S. Exploiting local features from deep networks for image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 53–61.
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, USA, 3–6 December 2012; pp. 1097–1105.
26. Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; Schmid, C. Deep matching: Hierarchical deformable dense matching. *Int. J. Comput. Vision* **2016**, *120*, 300–323. [CrossRef]
27. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Babenko, A.; Slesarev, A.; Chigorin, A.; Lempitsky, V. Neural codes for image retrieval. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 584–599.
29. Babenko, A.; Lempitsky, V. Aggregating deep convolutional features for image retrieval. *arXiv* **2015**, arXiv:1510.07493.
30. Kalantidis, Y.; Mellina, C.; Osindero, S. Cross-dimensional weighting for aggregated deep convolutional features. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; pp. 685–701.
31. Rezende, R.S.; Zepeda, J.; Ponce, J.; Bach, F.; Perez, P. Kernel square-loss exemplar machines for image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2396–2404.
32. Cao, J.; Liu, L.; Wang, P.; Huang, Z.; Shen, C.; Shen, H.T. Where to focus: Query adaptive matching for instance retrieval using convolutional feature maps. *arXiv* **2016**, arXiv:1606.06811.
33. Zhi, T.; Duan, L.Y.; Wang, Y.; Huang, T. Two-stage pooling of deep convolutional features for image retrieval. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, Arizona, USA, 25–28 September 2016; pp. 2465–2469.
34. Morra, L.; Lamberti, F. Benchmarking unsupervised near-duplicate image detection. *Expert Syst. Appl.* **2019**, *135*, 313–326. [CrossRef]
35. Minaee, S.; Abdolrashidi, A.; Su, H.; Bennamoun, M.; Zhang, D. Biometric recognition using deep learning: A survey. *arXiv* **2019**, arXiv:1912.00271.
36. Minaee, S.; Wang, Y.; Aygar, A.; Chung, S.; Wang, X.; Lui, Y.W.; Rath, J. MTBI identification from diffusion MR images using bag of adversarial visual features. *IEEE Trans Med. Imaging.* **2019**, *38*, 2545–2555. [CrossRef]
37. Zhang, Y.; Zhang, Y.; Sun, J.; Li, H.; Zhu, Y. Learning near duplicate image pairs using convolutional neural networks. *Int. J. Perform. Eng.* **2018**, *14*, 168–177. [CrossRef]
38. Gong, Y.; Wang, L.; Guo, R. Multi-scale orderless pooling of deep convolutional activation features. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 392–407.
39. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
40. Uricchio, T.; Bertini, M.; Seidenari, L.; Bimbo, A. Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 13–16 December 2015; pp. 9–15.
41. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
43. Lowe, D.G. Local feature view clustering for 3D object recognition. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; p. 1-1.
44. Fischer, P.; Dosovitskiy, A.; Brox, T. Descriptor matching with convolutional neural networks: A comparison to sift. *arXiv* **2014**, arXiv:1405.5769.
45. Yang, J.; Jiang, Y.G.; Hauptmann, A.G.; Ngo, C.W. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, Augsburg, Germany, 28–29 September 2007; pp. 197–206.

46. Sánchez, J.; Perronnin, F.; Mensink, T.; Verbeek, J. Image classification with the fisher vector: Theory and Practice. *Int. J. Comput. Vis.* **2013**, *105*, 222–245. [CrossRef]
47. Jégou, H.; Douze, M.; Schmid, C.; Pérez, P. Aggregating local descriptors into a compact image representation. In Proceedings of the CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3304–3311.
48. Hoogi, A.; Wilcox, B.; Gupta, Y.; Rubin, D.L. Self-Attention Capsule Networks for Image Classification. *arXiv* **2019**, arXiv:1904.12483.
49. Minaee, S.; Abdolrashidi, A. Deep-emotion: Facial expression recognition using attentional convolutional network. *arXiv* **2019**, arXiv:1902.01019.
50. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: Montreal, QC, Canada, 2015; pp. 2017–2025.
51. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
52. Harel, J.; Koch, C.; Perona, P. Graph-based visual saliency. In *Advances in Neural Information Processing Systems*; Van Can: San Diego, CA, USA, 2007; pp. 545–552.
53. Azizpour, H.; Sharif Razavian, A.; Sullivan, J.; Maki, A.; Carlsson, S. From generic to specific deep representations for visual recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 36–45.
54. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Object retrieval with large vocabularies and fast spatial matching. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
55. Jegou, H.; Douze, M.; Schmid, C. Hamming embedding and weak geometric consistency for large scale image search. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 304–317.
56. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
57. Simonyan, K.; Andrew, Z. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556 1409.
58. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.