mathematics

MDPI

Article

CNN Feature-Based Image Copy Detection with Contextual Hash Embedding

Zhili Zhou [1,2,*], Meimin Wang [1,2], Yi Cao [1,2,*] and Yuecheng Su [1,2]

[1] Jiangsu Engineering Centre of Network Monitoring & School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; wang_meimin@nuist.edu.cn (M.W.); su_yuecheng@nuist.edu.cn (Y.S.)

[2] Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing 210044, China

* Correspondence: zhou_zhili@nuist.edu.cn (Z.Z.); caoyi@nuist.edu.cn (Y.C.)

Received: 18 June 2020; Accepted: 13 July 2020; Published: 17 July 2020

Abstract: As one of the important techniques for protecting the copyrights of digital images, content-based image copy detection has attracted a lot of attention in the past few decades. The traditional content-based copy detection methods usually extract local hand-crafted features and then quantize these features to visual words by the bag-of-visual-words (BOW) model to build an inverted index file for rapid image matching. Recently, deep learning features, such as the features derived from convolutional neural networks (CNN), have been proven to outperform the hand-crafted features in many applications of computer vision. However, it is not feasible to directly apply the existing global CNN features for copy detection, since they are usually sensitive to partial content-discarded attacks, such as copping and occlusion. Thus, we propose a local CNN feature-based image copy detection method with contextual hash embedding. We first extract the local CNN features from images and then quantize them to visual words to construct an index file. Then, as the BOW quantization process decreases the discriminability of these features to some extent, a contextual hash sequence is captured from a relatively large region surrounding each CNN feature and then is embedded into the index file to improve the feature's discriminability. Extensive experimental results demonstrate that the proposed method achieves a superior performance compared to the related works in the copy detection task.

Keywords: image copy detection; convolutional neural networks (CNN); contextual hash; local CNN features; bag-of-visual-words (BOW)

1. Introduction

Due to the rapid development of Internet technology and the increasing popularity of personal digital camera devices, the amount of digital media (images, audio, and video) grows exponentially on the Internet [1–3]. With the help of various image processing tools such as Photoshop, it is very easy for users to modify a copyrighted image (an original image) with a variety of manipulations such as rescaling, rotation, cropping, noise addition, and text addition to produce various kinds of copy versions of the image for illegal use. Figure 1 shows the toy examples of an original image and its copies. In view of this, detecting image copies has become the first and key step for copyright protection.

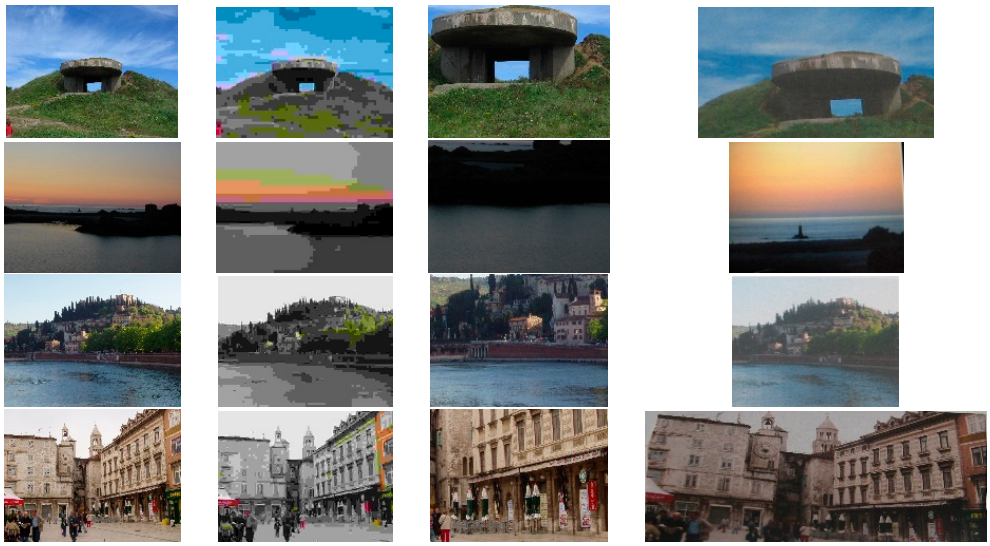

Figure 1. The toy examples of an original image and its copies. The left column is the original image, and the right three columns correspond to the copy versions generated by three different kinds of attacks, which are JPEG compression; cropping; and the combination of several manipulations, such as rotation, cropping, noise addition, and screen-shooting.

Generally, two typical techniques are popularly applied to detect illegal copies: digital watermarking [4] and content-based copy detection [5–8]. Digital watermarking embeds a watermark into the image file before its distribution. Consequently, all the copies of the marked image contain the watermark, which can be extracted and used as the proof of ownership. Instead of embedding additional information into the image, content-based copy detection directly relies on the image itself. Generally, a content-based copy detection system works as follows. It first collects numerous images downloaded from the networks to build a database, and extracts content-based features from the database images as their unique information. Then, for a given copyrighted image, the system compares its features to the features of the database images to determine whether there are copy versions of the copyrighted image in the database. Compared to the watermarking, the content-based copy detection does not need to embed the extra information but the image itself, and copy detection can be conducted after distribution [5,9]. Therefore, this paper focuses on content-based copy detection.

To resist various common copy attacks, the traditional content-based copy detection methods [6–14] are usually based on local hand-crafted image features, such as scale-invariant feature transform (SIFT) [15], principal component analysis on SIFT (PCA-SIFT) [16], and speeded-up robust feature (SURF) [17]. However, as hundreds to thousands of high-dimensional hand-crafted features are extracted from each image, directly matching these features between images for copy detection is very time-consuming. To reduce the time consumption of the matching process, a bag-of-visual-words (BOW) model [18] is adopted to quantize these features to visual words to build an inverted index file for copy detection.

Recently, deep learning techniques, particularly convolutional neural networks [19], have achieved great success in many applications of computer vision, such as image or scene classification [20], human activity recognition [21], and object defect detection [22]. Since the CNN features have been proven to be superior to hand-crafted features for content-based retrieval tasks [23], researchers prefer to employ CNN features for content-based retrieval. Some earlier works [24–27] feed an image into a pretrained CNN model and then use the output of the last fully-connected network layer as a global image representation. In some other works [3,23,26], instead of focusing on the features extracted from fully-connected layers, the features extracted from the deep convolutional layers are explored for the tasks of content-based retrieval.

Generally, the convolutional feature maps (CFMs) are first extracted from the deep convolutional layers of CNNs with an input image, and a pooling strategy such as sum-pooling [23] or max-pooling [28] is usually adopted to aggregate the feature maps into a single image representation. In our previous work [3], instead of only generating a single global representation from each image, we extract both the global and local CNN features from the CFMs and match these features between images with a coarse-to-fine matching strategy for near-duplicate image detection. In another work [29], the spatial-temporal CNN features are generated and matched for video copy detection. However, since the global CNN features are sensitive to partial content-discarded attacks such as cropping and occlusion, it is hard for these methods to detect the image copies generated by these attacks. Consequently, the retrieval accuracy is compromised to some extent, and it is not a reasonable choice to directly apply these global CNN features for copy detection.

Therefore, we attempt to propose a novel image copy detection method based on local CNN features with contextual hash embedding. First, we extract the CFMs from the deep convolutional layers of a pre-trained CNN model with an input image. Then, a number of local CNN features are generated by sum-pooling the feature values within the image regions detected by the SURF region detector [17]. Additionally, to improve the discriminability of the features, we extract a contextual hash sequence from a relatively large region surrounding each local feature. Next, these local CNN features are quantized to visual words by the BOW model to build an inverted index file, and the generated hash sequence is embedded into the index file. Finally, the local CNN features are matched efficiently between the images by looking up the inverted index file for copy detection. Our main contributions are summarized as follows.

(1) The extraction and indexing of local CNN features. The local CNN features are extracted by pooling the feature values of the convolutional feature maps (CFMs) within the regions detected by the SURF region detector. In the feature extraction, the regions detected by the SURF region detector change covariantly to the geometric transformations, including scaling and translation, and the feature values in CFMs are robust to a variety of content-preserved attacks due to the powerful training process. Therefore, the extracted local CNN features are not only robust to the partial content-discarded attacks, but also to the common geometric transformations and content-preserved attacks. Therefore, the extracted features have a high robustness, which will be beneficial to the accuracy of copy detection. Then, we index these local CNN features by the BOW model to form an inverted index file. By looking up the inverted index file, the feature matching process can be rapidly implemented for copy detection.

(2) The contextual hash embedding. To improve the discriminability of the quantized features, we also generate a contextual hash sequence for each feature and embed it into the inverted index file. Since the proposed contextual hash sequence is composed of a small number of hash values, it is quite compact and thus does not need too much additional storage space. Moreover, different from the CNN features that usually describe the complex patterns and semantic information of images, the hash sequence captures the correlations between blocks divided from relatively large regions surrounding the local CNN features, which can sufficiently characterize the contextual information of these features and thus improve the features' discriminability significantly. That will lead to a higher detection accuracy.

The reminder of this paper is organized as follows. In Section 2, we introduce the proposed copy detection method in detail. The experimental results and analysis are given in Section 3. Section 4 draws the conclusions.

2. The Proposed Copy Detection Method

In this section, the proposed copy detection method will be introduced. The framework of the proposed copy detection method is illustrated by Figure 2. In Section 2.1, we introduce the generation of the CFMs for a given image. In Section 2.2, we describe how to extract the local CNN features from the CFMs. In Section 2.3, a contextual hash sequence is generated for each local CNN feature. In Section 2.4, the extracted local CNN features are quantized by the BOW model to build an inverted

index file, and the generated hash sequence is embedded into the index file. In Section 2.5, by looking up the inverted index file, we match the features between images for copy detection.

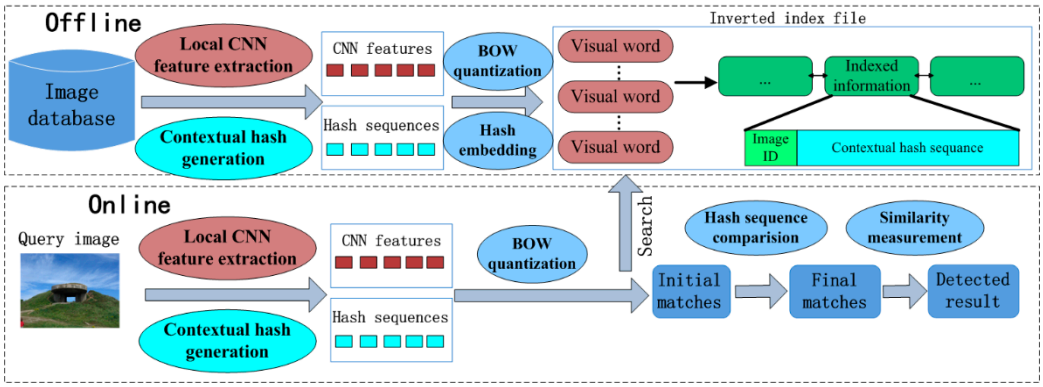

Figure 2. The framework of the proposed copy detection method.

2.1. CFM Generation

Generally, a typical CNN model is composed of a number of layers, including convolutional layers, pooling layers, and fully connected layers. As illustrated in [23], the CNN features extracted from deep convolutional layers perform better than the features from other layers in many retrieval tasks. Thus, for a given image, it is fed into a pretrained CNN model and the output of the fifth convolutional layer—i.e., a set of CFMs—is used for the local CNN feature extraction. In our method, we adopt the famous CNN model—i.e., *AlexNet* [19].

From [19], by feeding an image into the AlexNet model, the output of the fifth convolutional layer is K feature maps with the size of $W \times H$, where $K = 256$ and W and H are proportional to the width and height of the image, respectively. Denote the K feature maps as $CFMs = \{M_1, M_2, \cdots, M_i, \cdots, M_K\}$. These CFMs will be further used for local CNN feature extraction. Figure 3 illustrates the generation of CFMs.

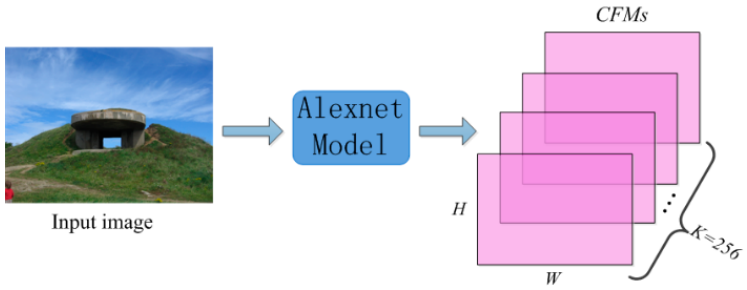

Figure 3. The illustration of convolutional feature maps (CFM) generation.

2.2. Local CNN Feature Extraction

To extract the local CNN features, we first detect a set of regions from the image and then extract the local CNN features by pooling the feature values within the detected regions. To achieve a high robustness of the common geometric transformations including rotation, rescaling, and translation, the regions detected for feature extraction should change covariantly to these transformations. To this end, the SURF region detector [17] is adopted to detect the regions from images, since the SURF detector can efficiently detect the regions that change covariantly to the above transformations, as illustrated in [17].

Since the sizes of CFMs are proportional to the size of the image, for a region detected on the image R, we can map the region to the CFMs to obtain its corresponding region R_M on CFMs, according to the ratio between the size of the image and its CFMs. Then, we adopt the sum-pooling strategy [23] to

aggregate the feature values of each CFM within the region R_M to extract the $K = 256$ dimensional local CNN feature F_{R_M}, by:

$$F_{R_M} = \{\sum_{p \in R} M_i(p) | 1 \le i \le K\}, \quad (1)$$

where, p represents a point located in the region R_M, and $M_i(p)$ means the feature value of p on the i-th feature map. Finally, we normalize the extracted feature by L2-normalization. Figure 4 illustrates the local CNN feature extraction. As hundreds to thousands of SURF regions are detected from each image, the same number of local CNN features can be extracted by the above step.

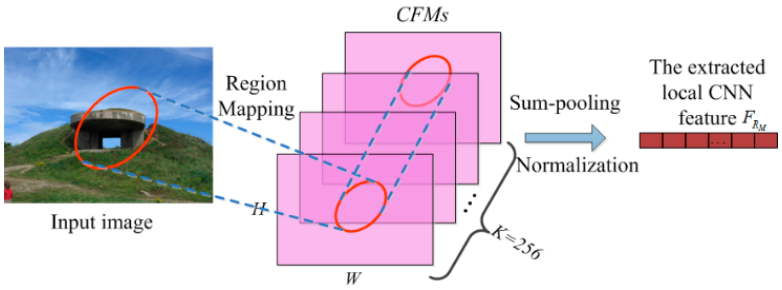

Figure 4. The illustration of local CNN feature extraction.

2.3. Contextual Hash Generation

In our method, the extracted CNN features will be quantized to visual words to construct an inverted index file for rapid image matching. However, the BOW quantization process will decrease the discriminability of the features to some extent. To improve the features' discriminability, we generate a contextual hash sequence of each local CNN feature and then embed it into the index file. In our method, we use the correlations between blocks divided from the relatively large region surrounding each local CNN feature to generate the contextual hash sequence. The algorithm of the contextual hash generation is described as follows.

Suppose the radius of a detection SURF region is r. To capture the contextual information, we expand the region proportionately, and the radius of the extended region $R_M{}'$ is denoted as:

$$r' = r \times \alpha, \quad (2)$$

where α is set as 3.2 by experiments.

For a given expended region $R_M{}'$, we first divide it into $M \times N$ blocks with equal size in the log-polar space and then compute the average gray intensities of these blocks. We denote the average gray intensity of block b in the expended region as G_b, and those of the eight adjacent blocks of block b as $G_b(i)$, where $i \in [1, 8]$. Note that some adjacent blocks of an edge block do not exist, and thus their average gray intensities are set as 0. Then, the hash values of block b denoted by $v_b = \{v_b(1), v_b(2), \cdots, v_b(i), \cdots, v_b(8)\}$ can be generated by Equation (3). Figure 5 shows an example of the extraction of feature values from a block.

$$v_b(i) = \begin{cases} 1, \ if \ G_b(i) > G_b \\ 0, \ otherwise \end{cases} \quad where \ i \in [1, 8] \quad (3)$$

In the above manner, the hash values of all the blocks in the region $R_M{}'$ are computed. Then, we concatenate the hash values of all the blocks to generate the contextual hash sequence of the local CNN feature, denoted as CH, which will be further embedded into the index file. Since the correlations of adjacent blocks that describe the comparative intensity relationships of adjacent blocks are less likely to be changed by various transformations, the generated hash sequence has a high robustness. Moreover, each hash sequence is composed of a small number of hash values, and

thus we do not need too much additional memory space for storing it. In addition, the contextual hash sequence sufficiently captures the contextual information of the local CNN feature, which can significantly improve the feature's discriminability for copy detection.

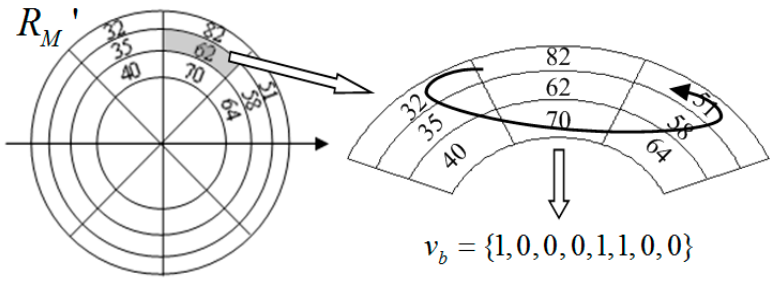

Figure 5. The extraction of the hash values from a block.

2.4. Index File Construction

As hundreds to thousands of local CNN features are detected from each image, it is very time-consuming to directly match these features between images for copy detection. Thus, in this section, we quantize these features to visual words based on the BOW model and then build the inverted index file for efficient copy detection.

Specifically, in the BOW model, numerous sample features are clustered to generate a set of clusters by a clustering algorithm—i.e., K-means—and each cluster center is viewed as a visual word to form a visual vocabulary. Then, we extract the local CNN features from all the database images, and then quantize them to the corresponding nearest visual words of the vocabulary to build the inverted index file.

The structure of the inverted index file of our method is illustrated by Figure 6. Each visual word is followed by the indexed information, each of which stores the ID of the image where the visual word occurs and the contextual hash sequence. Note that the generation of the contextual hash sequence for each local CNN feature is described in next subsection.

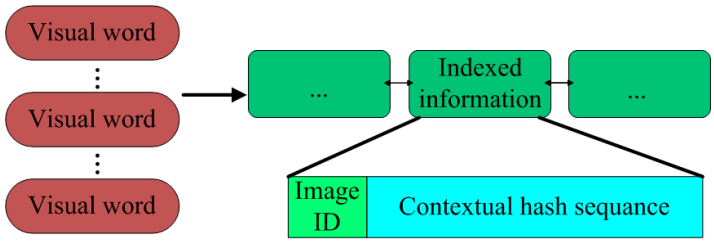

Figure 6. The structure of the inverted index file.

2.5. Copy Detection

By the above steps, we can obtain the inverted index file for the database images. Next, we will introduce the implementation of copy detection in detail.

For a query image, we also extract its local CNN features and the corresponding hash sequences by the algorithms described in Sections 2.3 and 2.4, respectively. Then, by looking up the index file, any two local CNN features from different images quantized to the same visual word are treated as a candidate local match between the images.

Next, we compute the distance between the corresponding contextual hash sequences of initially matched local CNN features, denoted as CH_Q and CH_D, to confirm whether they are a true match. The distance is computed by:

$$Dis(CH_Q, CH_D) = \frac{\sum_{i=1}^{8\times M\times N} \left| CH_Q(i) - CH_D(i) \right|}{8 \times M \times N}, \quad (4)$$

where $CH_Q(i)$ and $CH_D(i)$ are the i-th elements in CH_Q and CH_D, respectively. If the distance is smaller than a preset threshold Dis_T, we determine that the match is a true one. Then, like the tradition BOW-based retrieval methods [7,30], each matched feature casts the corresponding database image a vote weighted by the inverted document frequency (IDF) [30]. The similarity of the query image to a database image is measured by adding up all the weighted votes. Finally, we compare the similarity with a pre-set threshold to determine whether a database image is a copy version of the query.

3. Experiments

In this part, the dataset and evaluation criteria used in our experiments are described first. Second, the optimal parameter setting is determined by experiments. Third, the performance of the proposed method is evaluated and compared to those of the state-of-the-art methods.

3.1. Datasets and Evaluation Criteria

In the experiments, two datasets are adopted, which are detailed as follows.

(1) ***Copydays dataset*** **[31]**. This dataset is composed of 3212 images. There are 157 original ones and 3055 copies that are generated by different kinds of image attacks, such as JPEG compression, cropping, and "strong" attacks. The "strong" attacks mean the different combinations of a variety of manipulations such as scaling, blurring, and rotation. For each original image, it has nine copy versions that are generated by JPEG compression with different quality factors, nine copy versions that are generated by cropping the image from 10% to 80%, and 2 to 6 copy versions that are generated by "strong" attacks. The 175 original images are used as query images for copy detection.

(2) ***DupImage dataset*** [32]. This dataset contains 1104 images. In this dataset, there are 33 image groups. In each group, the first image is an original image, and the other images are the copy regions, which are cropped from the original image with a variety of copy attacks such as rescaling, noise addition, and compression. We use the 33 first images of these groups as query images for copy detection.

In the experiments, we adopt Mean Average Precision (MAP) to test the performances of the different methods. When detecting the copies of a given query in the database, by setting the image similarity threshold to different values we can obtain a set of pairs of precision and recall rates. Thus, we can compute the average precision across all the different recall levels. MAP is obtained by computing the mean value of the average precisions of all queries.

Note that the experiments are implemented on a personal computer (3.2 GHz Core-i5 and 8 GB RAM) with Windows 7 × 64 operation system.

3.2. Parameter Determination

In this part, the impacts of three key parameters are tested: the parameters used for block generation—i.e., M and N—and the threshold used for feature matching—i.e., Dis_T. The size of the visual vocabulary used for the index file construction is set as 20K.

First, the threshold Dis_T is fixed to a default value—i.e., 0.3—to observe the impacts of the parameters M and N on the MAP values. From Figure 7, we can clearly observe that too large or too small M and N lead to an inferior detection performance for the following reasons. A larger M and N lead to more blocks divided from each region and a smaller number of pixels in each block, which will

make the generated contextual hash sequence more sensitive to many copy attacks, such as nosing addition and blurring. A smaller M and N cause fewer blocks from each region, which will decrease the dimensionality of the generated contextual hash sequence. Thus, the accuracy of feature matching will be affected to some extent. According to Figure 7, $M = 4$ and $N = 4$ provide the highest detection accuracy, and thus we use the above settings in the following experiments.

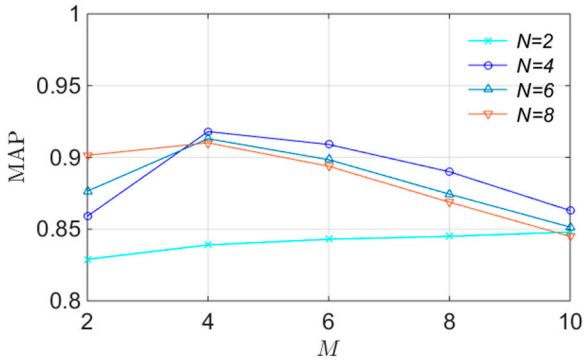

Figure 7. The effects of M and N on accuracy.

Then, we test the impact of the threshold Dis_T. Figure 8 shows the effects of Dis_T on the MAP. From this figure, it can be clearly observed that the detection performance degrades when Dis_T is too small or too large. The reason is that if Dis_T is too small, a considerable number of true feature matches will be detected as false ones, and if Dis_T is too large, many false matches will be determined as true ones. According to this figure, when $Dis_T = 0.25$, we can achieve the highest detection accuracy. Thus, we set $Dis_T = 0.25$ in the following experiments.

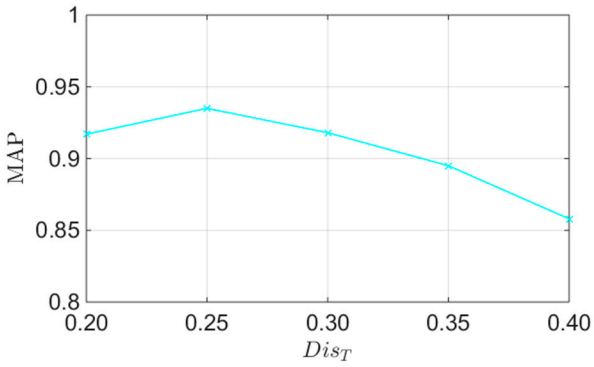

Figure 8. The effects of Dis_T on accuracy.

3.3. Performance Evaluation

In this part, we will compare the performance of the proposed method to those of five other methods, which are listed as follows.

(1) ***SIFT + BOW*** [33]: This is the method based on the hand-crafted local features—i.e., SIFT features [15] and the BOW model [18]. First, a set of SIFT features are extracted from each image, and then they are quantized by the BOW model to build an inverted index file for image copy detection.

(2) ***SIFT + BOW + GC*** [8]: This method is very similar to the previous one, but it has an additional step—i.e., feature match verification by geometric coding (GC). Specifically, after obtaining feature matches between images by inverted index file, a geometric coding algorithm is adopted for feature match verification to filter false matches. The remaining feature matches are used to evaluate the image similarity for copy detection.

(3) ***Global CNN + Local CNN + CF*** [3]: This method extracts both the global and local CNN features from the CFMs generated by a pre-trained CNN model—i.e., Alexnet [19]—and match these features between images with a coarse-to-fine (CF) strategy for copy detection.

(4) ***Global CNN + VDSH*** [27]: In this method, we extract the global CNN features from the last fully connected layer of the same pre-trained CNN model—i.e., Alexnet [19]—and index these global features by the BOW model. Then, the VDSH algorithm [34] is employed to calculate the hash codes of images to improve the discriminability of the global features for copy detection.

(5) ***Local CNN + BOW***: Different from SIFT + BOW, this method uses the local CNN features extracted by the algorithm in Section 2.2 instead of the SIFT features.

(6) ***Local CNN + BOW + CHE***: This method is the proposed method, which extracts the local CNN features and then quantizes them to visual words by the BOW model to build the inverted index file, and the contextual hash sequences are generated and embedded into the index file.

We set the size of visual vocabulary as 20K to test the detection performances of those methods on the two datasets. The comparison results are shown in Tables 1 and 2, where the average time cost per query is adopted to evaluate the time efficiency, while the memory consumption per indexed feature is used to measure the space efficiency of those methods. From Table 1, it can be clearly observed that our method—i.e., **Local *CNN + BOW + CHE***—achieves the highest accuracy among all of these methods on the Copydays dataset. Our method achieves a higher accuracy than the **SIFT + BOW** and **SIFT + BOW + GC**, mainly because our method uses the local CNN features, which have a higher discriminability than the local hand-crafted features. The accuracy of our method is higher than those of **Global CNN + Local CNN + *CF*** and **Global CNN + VDSH**. That is because the proposed local CNN features are more robust than the global CNN features to the partial content-discarded attacks, such as cropping and occlusion. Additionally, our method outperforms **Local CNN + BOW**, since the contextual hash sequence embedded into the index file can significantly improve the discriminability of the local CNN features.

From Table 2, our method still achieves the highest accuracy on the Dupimage dataset. All of the above methods, especially **Global CNN + Local CNN + CF** and **Global CNN + VDSH**, achieve worse performances on the Dupimage dataset than on the Copydays dataset. That is because the Dupimage dataset contains a lot of image copies generated by the partial content-discarded attacks such as cropping and occlusion, and the global CNN features are much more sensitive to these attacks than the proposed local features.

From Tables 1 and 2, we can also observe that the time efficiency of our method is higher than that of **SIFT + BOW + GC** and **Global CNN + Local CNN + CF**, and is slightly lower than that of **Global CNN + VDSH** and **Local CNN + BOW**, since our method needs the additional verification step to confirm the local CNN feature matches by computing the distances between the contextual hash sequences. Our method requires comparable memory space to **SIFT + BOW + GC** and **Global CNN + VDSH**, and a higher memory space than **SIFT + BOW** and **Local CNN + BOW**. That is because the additional contextual hash sequence needs to be embedded into the inverted index file in our method.

In conclusion, our method provides a higher accuracy than the five other methods, while maintaining a desirable performance in the aspects of both space and time efficiency. Some examples of copy detection results of our method are shown in Figure 9.

Table 1. Comparison between different methods on the Copydays dataset.

	SIFT + BOW	***SIFT + BOW + GC***	***Global CNN + Local CNN + CF***	***Global CNN + VDSH***	***Local CNN + BOW***	***Local CNN + BOW + CHE***
MAPs	0.783	0.882	0.762	0.681	0.827	0.935
Average Time cost (second)	0.139	0.385	0.552	0.203	0.258	0.304
Memory consumption per feature (Bytes)	8	16	32	16	8	20

Table 2. Comparison between different methods on the Dupimage dataset.

	SIFT + BOW	***SIFT + BOW + GC***	***Global CNN + Local CNN + CF***	***Global CNN + VDSH***	***Local CNN + BOW***	***Local CNN + BOW + CHE***
MAPs	0.491	0.689	0.543	0.324	0.587	0.861
Average Time cost (second)	0.158	0.866	0.943	0.235	0.284	0.328
Memory consumption per feature (Bytes)	8	16	32	16	8	20

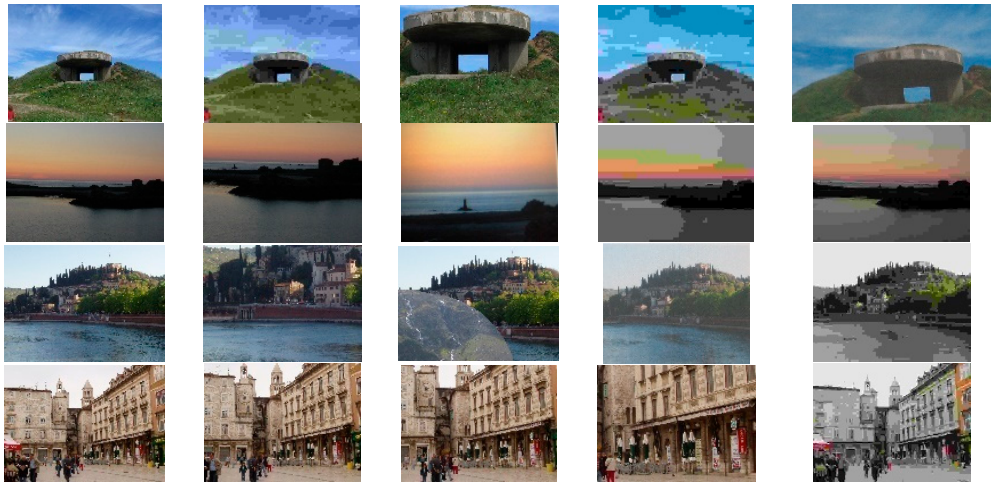

Figure 9. The examples of our detection results (four queries and the corresponding four detected images) on the Copydays dataset. All of these detected images are the image copies.

4. Conclusions

We have presented a local CNN feature-based copy detection method with contextual hash embedding. The local CNN features show a higher discriminability than the traditional hand-crafted features, which is beneficial to the accuracy of copy detection. Thus, instead of using the traditional hand-crafted features, we quantize the extracted local CNN features by the BOW model to build the inverted index file. To further improve the discriminability of CNN features, the corresponding contextual hash sequences of each CNN feature are generated and embedded into the index file. The experimental results show that the proposed copy detection method achieves a promising accuracy, while maintaining good performances in the aspects of time and space efficiency. However, the extracted local CNN features still show limited robustness to the "strong" copy attacks, because the CNN features are extracted from a pre-trained CNN model, which does not take these attacks into account during the training process. To extract more robust CNN features, one of feasible solutions is to train a proper CNN model with a transfer learning technique for feature extraction. Future work will focus on how to further improve the robustness of the local CNN features. Moreover, also a significant research direction is how to extend the proposed method for emerging applications—i.e., cross-media retrieval.

Author Contributions: Conceptualization, Z.Z. and Y.C.; methodology, Z.Z. and Y.C.; software, M.W. and Y.S.; validation, Z.Z., Y.C., M.W. and Y.S.; formal analysis, Y.C.; investigation, M.W.; resources, Z.Z.; data curation, Y.C.; writing–original draft preparation, Z.Z.; writing–review and editing, Y.C., M.W. and Y.S.;visualization, M.W. and Y.S.; 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 work was supported in part by the National Natural Science Foundation of China under Grant 61972205, U1836208, U1836110; in part by the National Key R&D Program of China under Grant2018YFB1003205; 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 under Grant MOST 108-2221-E-259-009-MY2 and 109-2221-E-259-010, Taiwan.

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

References

1. 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] [PubMed]
2. Zhou, Z.; Wu, Q.M.J.; Yang, Y.; Sun, X. Region-level Visual Consistency Verification for Large-Scale Partial-Duplicate Image Search. *ACM Trans. Multimed. Comput. Commun. Appl.* **2020**, *16*, 1–25. [CrossRef]

3. Zhou, Z.; Lin, K.; Cao, Y.; Yang, C.N.; Liu, Y. Near-duplicate Image Detection System Using Coarse-to-Fine Matching Scheme based on Global and Local CNN Features. *Math* **2020**, *8*, 644. [CrossRef]
4. Zhou, Z.; Mu, Y.; Wu, Q.M.J. Coverless image steganography using partial-duplicate image retrieval. *Soft Comput.* **2019**, *23*, 4927–4938. [CrossRef]
5. Kim, C. Content-based image copy detection. *Signal Process. Image Commun.* **2003**, *18*, 169–184. [CrossRef]
6. Zhou, Z.; Wang, Y.; Wu, Q.M.J. Effective and efficient global context verification for image copy detection. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 48–63. [CrossRef]
7. Zhou, Z.; Wu, Q.M.J.; Sun, X. Multiple distances-based coding: Toward scalable feature matching for large-scale web image search. *IEEE Trans. Big Data* **2019**. [CrossRef]
8. Zhou, W.; Li, H.; Lu, Y.; Tian, Q. SIFT match verification by geometric coding for large-scale partial-duplicate web image search. *ACM Trans. Multimed. Comput. Commun. Appl.* **2013**, *9*, 1–18. [CrossRef]
9. Ling, H.; Yan, L.; Zou, F.; Liu, C.; Feng, H. Fast image copy detection approach based on local fingerprint defined visual words. *Signal Process.* **2013**, *93*, 2328–2338. [CrossRef]
10. Ling, H.; Wang, L.; Zou, F.; Yan, W. Fine-search for image copy detection based on local affine-invariant descriptor and spatial dependent matching. *Multimed. Tools Appl.* **2011**, *52*, 551–568. [CrossRef]
11. Ling, H.; Cheng, H.; Ma, Q.; Zou, F.; Yan, W. Efficient image copy detection using multiscale fingerprints. *IEEE Multimed.* **2012**, *19*, 60–69. [CrossRef]
12. Yan, L.; Zou, F.; Guo, R.; Gao, L.; Zhou, K.; Wang, C. Feature aggregating hashing for image copy detection. *World Wide Web* **2016**, *19*, 217–229. [CrossRef]
13. Zhou, Z.; Sun, X.; Wang, Y.; Fu, Z.; Shi, Y. Combination of SIFT Feature and Convex Region-Based Global Context Feature for Image Copy Detection. In Proceedings of the 14th International Workshop on Digital-forensics and Watermarking, Tokyo, Japan, 7–10 October 2015; pp. 60–71.
14. Zhou, Z.; Yang, C.; 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**, *6*, 1531–1540. [CrossRef]
15. Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
16. Yan, K.; Sukthankar, R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In Proceedings of the 2004 Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; pp. 506–513.
17. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2018**, *110*, 346–359. [CrossRef]
18. Sivic, J.; Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the 9th IEEE International Conference of Computer Vision, Nice, France, 10–13 October 2003; pp. 1470–1477.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Image Net Classification with Deep Convolutional Neural Networks. In Proceedings of the IEEE Conference on Neural Information Processing System, Lake Tahoe, CA, USA, 6 December 2012; pp. 1097–1105.
20. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
21. Murad, A.; Pyun, J.Y. Deep recurrent neural networks for human activity recognition. *Sensors* **2017**, *17*, 2556. [CrossRef]
22. Perez, H.; Tah, J.H.; Mosavi, A. Deep learning for detecting building defects using convolutional neural networks. *Sensors* **2019**, *19*, 3556. [CrossRef]
23. Babenko, A.; Lempitsky, V. Aggregating Deep Convolutional Features for Image Retrieval. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1269–1277.
24. 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.
25. Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S. 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.
26. Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-The-Shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 806–813.

27. Liu, R.; Wei, S.; Zhao, Y.; Yang, Y. Indexing of the CNN features for the large scale image search. *Multimed. Tools Appl.* **2018**, *77*, 32107–32131. [CrossRef]
28. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
29. Zhou, Z.; Chen, J.; Yang, C.N.; Sun, X. Video Copy Detection Using Spatio-Temporal CNN Features. *IEEE Access* **2019**, *7*, 100658–100665. [CrossRef]
30. Jegou, H.; Douze, M.; Schmid, C. Hamming embedding and weak geometric consistency for large scale image search. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 304–317.
31. Copydays. Available online: http://lear.inrialpes.fr/~{}jegou/data.php (accessed on 14 July 2020).
32. DupImage. Extraction Code: vwk3. Available online: https://pan.baidu.com/s/1AMT7cdkHVYgMIR8sTeQFSg (accessed on 14 July 2020).
33. Nister, D.; Stewenius, H. Scalable recognition with a vocabulary Tree. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 2161–2168.
34. Zhang, Z.; Chen, Y.; Saligrama, V. Efficient training of very deep neural networks for supervised hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 14–19 June 2016; pp. 1487–1495.