Convolutional Neural Networks for Duplicate Image Detection

Akomolafe, O.P & Adebayo, I.G.
Department of Computer Science
University of Ibadan
Ibadan, Oyo State, Nigeria
E-mails: akomspatrick@yahoo.com; adebayoig@gmail.com
Phones: +2348030848696; +2348067023617

ABSTRACT

The breakthrough in deep neural networks have demonstrated astounding results on image representation and different computer vision tasks. Unfortunately their use in duplicate image detection may lead to a capacity bottleneck as such a system may suffer from the curse of dimensionality, in that, search speed, memory requirement and database size can grow so fast with the data dimension. In this paper, we introduced trainable feature extractors comprising of a convolutional neural network with varying descriptor sizes. The extraction time, descriptor size, duplicate detection accuracy, storage savings and number of misclassified image pairs were recorded. Experimental results show that to obtain high duplicate detection accuracy it is not necessary to use a convolutional neural network's large image descriptor.

Key words: Convolutional neural network, Computer vision, Image descriptor, Feature extractor, Image detection

iSTEAMS Proceedings Reference Format
Akomolafe, O.P & Adebayo, I.G. (2019): Convolutional Neural Networks for Duplicate Image Detection.
Proceedings of the 18th iSTEAMS Multidisciplinary Cross-Border Conference, University of Ghana, Legon, Accra, Ghana. 28th – 30th July, 2019
Pp 423-431 www.isteams.net - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V18N1P47

1. BACKGROUND TO THE STUDY

The rapid developments of the Internet, the increasingly affordable bandwidth, and the widespread adoption of smartphones capable of capturing images and editing have contributed to an explosion in the number and variety of images that are posted online. Reference [1] reports that all Internet users share over 3 billion photos each day translating to over 34,000 images being uploaded per second. These advances in technologies bring along a variety of convenience as well as much challenges to information security. Digital images can easily be duplicated and distributed without the consent of the owner thereby infringing the ownership rights of the owners. These duplicated images may have been manipulated by some image processing and therefore may not be the exact copy of the original one. Hence, duplicate images essentially include the identical copy and digitally altered versions of the original image after the manipulations. In general, being able to detect image duplicates has played an important role in a lot of applications including storage optimisation, identifying copyright violations, video copy detection, image spam detection, and improving image search engine results by grouping related images.

In a typical duplicate image detection system, images are first converted into a particular image representation that can optionally be stored in an indexing structure. The main discerning factor for whether a duplicate image detection system can handle a large dataset is its underlying image representation. For such systems to be highly scalable, the image representation must not be less than a kilobyte (1000 bytes) of storage [2] and robust to image transformations such as rescaling, aspect ratio change, JPEG compressions, colour transforms, right-left flip etc. This means that a duplicate image system faces three conflicting performance requirements: descriptor size, description time and accuracy.

On the one hand, such a system must achieve a high accuracy at identifying duplicate version of an original image which often means high computational cost. On the other hand, if the size of the feature extracted per image (descriptor size) is large, it imposes a high storage and retrieval cost. To meet these performance requirements, a duplicate image detection system needs a good descriptor small enough to tell apart different images and must be robust to different image transformations. The remainder of the paper is organized as follows. In section 2, a review of the related works is presented. The proposed methodology is described in Section 3. After that, application of the proposed algorithm is discussed in section 4, and we draw our conclusion in the last section.

2. RELATED WORKS

Existing solutions to duplicate image detection rely either on the use of watermarks or on features extracted from the image itself. Since the main performance differences between current duplicate image detection methods are about how well they are able to deal with images that are modified using exotic transformations, it is not surprising that most of the state of the art approaches are content-based as they are more robust to image transformations. Content-based copy detection was proposed as an alternative means of identifying illegal image copies. The idea is that, instead of hiding additional information in an image, the image itself can be employed for the same purpose. Content-based methods can be independently used to distinguish illegal copies or can complement digital watermark techniques. However, content-based copy detection methods have a higher computational cost, making scalability more difficult to achieve [3].

Global feature content-based copy detection aims to extract global statics information to represent images. They can be divided into colour features, shape features, texture features and spatial structure features. The major advantage of global features are their simple calculation and less space requirement. However, due to focusing on the overall image information, global features tend to ignore local information in images [4]. This kind of technique is very efficient at finding identical copies but maybe sensitive to the variation of lighting and viewpoint (or occlusion) [5]. Their poor resistance to geometric attacks, especially cropping and rotation, has made scholars prefer detecting image on local features [6].

Local feature content-based copy detection based copy detection first tries to find matches between individual salient details. They first detect the stable regions of an image and then extract high-dimensional feature vectors in the vicinity of each region. Hence, the image is represented as a set of feature vectors. With respect to global features, local features are more suitable to working with local changes in image content. Meanwhile, the matching algorithms of local features are complex and face difficulty in meeting the real-time requirement in large-scale image retrieval [4]. This method can deal with the illumination variation and geometric transformation but at the expense of computational efficiency [5], [7].

Reference [8] proposed RIME (replicate image detector) as an alternative approach to watermarking for detecting unauthorised image copy on the Internet by characterising each image using Daubeclues' Wavelets Transform (DWT). Reference [9]'s system used an image description based on local differential descriptors to describe the visual content of images. Reference [5] presented a key point-based approach that consists of three steps: key points extraction, an estimation of affine invariant to reduce bin space and comparison of colour histograms of areas formed by matched key points. Reference [10] conducted an interest point extraction operation on images and represented each interest point with SIFT descriptor. Reference [11] used local key-points represented by an adapted binary fingerprint to represent images. Reference [12] presented a framework for near-duplicate image detection by combining Bag-of-Words model with spatial pyramid. Their framework consists of three elements – independent multi-codebooks, non-negative sparse coding and an improved intersection kernel function.

Reference [13] combined different descriptors through an alignment procedure based on clusters correspondence to capture different aspects of a considered local region to reduce both visual word ambiguity and the quantisation error in visual book generation. Reference [14] proposed a hashing based method that generates compact fingerprint for image representation to prevents huge semantic loss during hashing for duplicate image detection. Reference [15] extracted Ultra Short Binary (USBs) descriptors from image patches to directly compress their visual clues into a representation as compact as their visual vocabulary. Reference [16] presented a duplicate image detection scheme that adopts multiple hash tables by using Image Secret Sharing (ISS) to generate a black-and-white image which is later used to generate a hash. Reference [17] proposed a feature aggregating method for duplicate image detection using machine learning based hashing. Their motivation was based on the fact that since machine learning based hashing effectively preserves neighbourhood structure of data, it could yield visual words with strong discriminability. Visual vocabulary was constructed by first extracting local features from training image dataset. After which k-means is applied to generate k-clusters. Machine learning based hashing was used to generate binary codes of visual words and forgetting hashing functions to efficiently map local features into binary codes. Histograms of the visual words are then used to construct image representation.

Reference [18]'s system used Locality-Constrained Linear Coding (LLC) with $maxIDF$ cut model was used to represent image features. Map Reduce-based image partitioning method and pairwise merging were further applied to identify image duplicates from a large scale image set. Reference [19] extracted a hash (binary signature) from a resized version of an input image. The constructor of the descriptor is mostly based on pixel value comparisons. Reference [4] used a perception hash tag to generate image signatures. Reference [7] presented an image representation known as Local-based Binary Representation (LBR) is based on the collective information from local regions. Densely local regions from an input image are sampled and converted to alike block-based local binary pattern features. These local features are counted together to form a histogram based on some specific rules. The histogram is then encoded to a 64-bit binary vector to further improve its efficiency for match and online reduced cost. Local features have a good image recognition ability, but its computational complexity is high [6] and also suffer from low matching efficiency [16].

Although most researches in duplicate image domain rarely provide information on their descriptor size, reference [2] used representative techniques to evaluate the performance of content-based duplicate image detection methods in relation to their descriptor size, description time and matching time to assess their feasibility of application to large image collections. Reference [19] aimed at having a descriptor size less than 100 bytes to enable a fast and exhaustive search of a large database. Reference [6] presented a model that automatically differentiates copied versions of original images by learning a comparing function directly from raw image pixels. Unfortunately the use of convolutional neural networks in duplicate image detection may lead to a capacity bottleneck as such a system may suffer from the curse of dimensionality, in that, search speed, memory requirement and database size can grow so fast with the data dimensions.

This paper only focuses on processing of images (feature extraction) and comparison (feature matching) rather than the efficiency of storing and querying feature vectors (image representations) which has its own host of computational challenges.

3. METHODOLOGY

Figure.1 gives a bird's view eye view of the duplicate image detection process. Its components include a trainable feature extractor, a database to store extracted features and a feature matcher which accepts as input, two feature vectors and returns a corresponding output 0 or 1 in which 0 represents the nonexistence of a copy relationship between the two inputs and 1, the opposite.

The following are factors considered in the design of the system:

a) The system be able to identify image distortions without the cost inherent in existing systems that use local, spatial features.
b) The system should be trainable i.e. must be able to accept several images and also categorise them based on "copy" or "not a copy".
c) The feature extraction phase should be separated from the feature matching phase.
d) The size of the features extracted (descriptor) should be as small as possible but should still be good enough to tell apart different images.

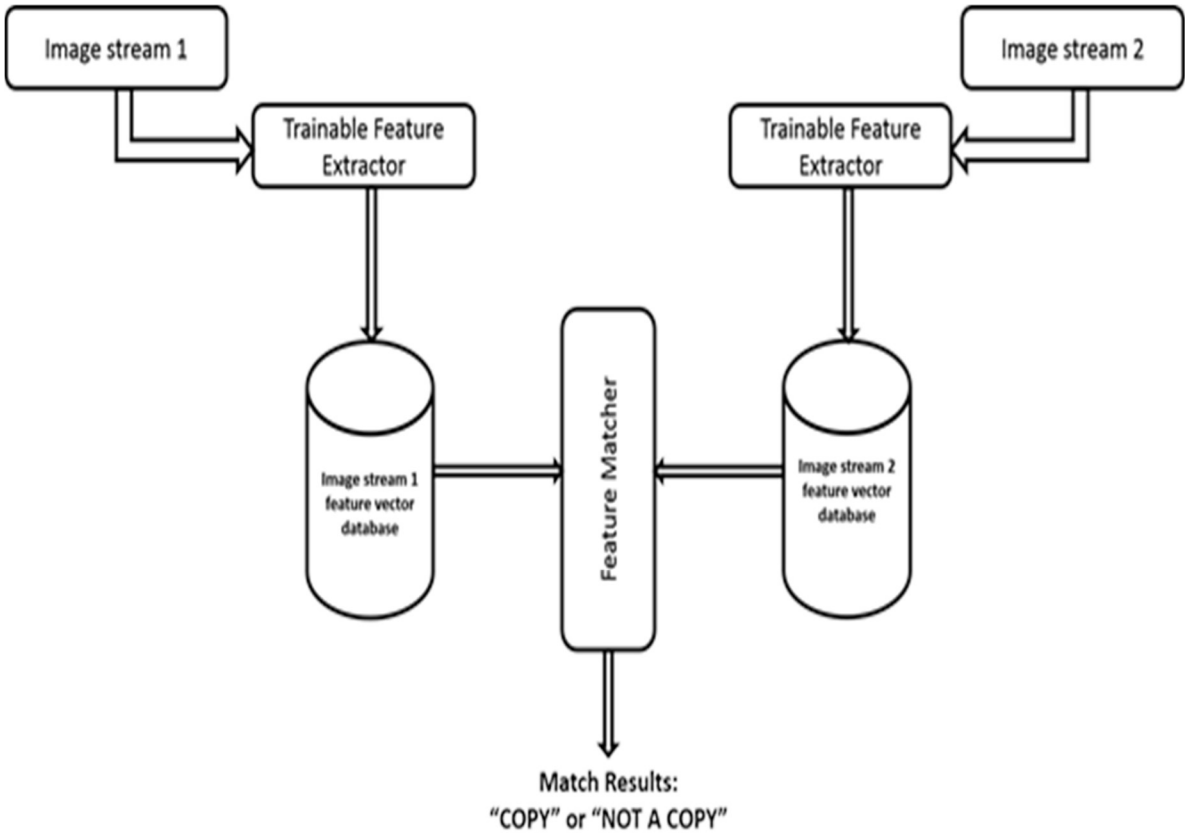

Figure 1: Duplicate image detection system

Feature Extraction

A feature extractor processes the raw data from an image to generate a feature vector (descriptor) which usually has smaller dimension than the original data while still holding the maximum amount of useful information given by the data. The trainable feature extractor (TFE) comprises of a pre-trained convolutional neural network [20] used to extract useful information from the images and a new neural network to reduce the number of output features. The idea is to replace the layer that does the final classification in [20] with a new neural network. Thus the output from pre-trained network become input to the new neural network and then trained to be linearly separable and used as features for the copy decision model. The resulting system is a trainable feature extractor (TFE) as shown in figure 2.

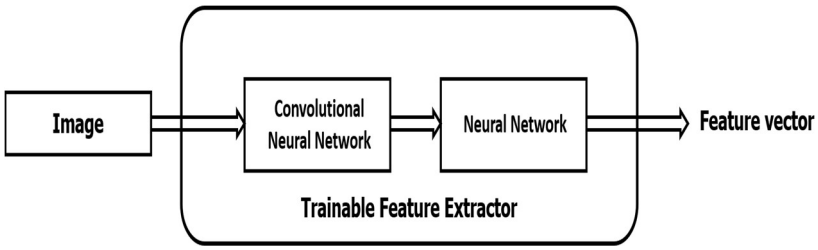

Figure 2 Trainable Feature Extractor

Feature Matching

The feature matching phase makes use of a copy checker shown in figure 3. It is a three-layer vanilla neural network with an input layer, a fully-connected layer and a softmax layer. Our model takes features already extracted from images as input. These are the images on which we want to identify a copy relationship on returning a corresponding output 0 or 1.

The first layer is the input layer of the network with f1+f2 neurons. f1 is the number of features extracted from image 1 and f2 is the number of features extracted from image 2. The second layer of the vanilla neural network is fully connected with n neurons. The number of neurons in this layer is denoted by n as we experimented with different values of n. The third layer of the network is the softmax layer containing two (2) neurons. They represent the goal of the network which is to tell if both features are a copy pair (i.e. a duplicate of each other) or not.

Ten duplicate image detectors which consists of ten feature extractors with different descriptor sizes were used to extract features from images in the dataset and ten copy checkers for each descriptor size. All models were trained with the same dataset. The only difference between the models for feature extraction phases are the numbers of features extracted by each trainable feature extractor and the number of input features used to train each copy checker.

Two main indicators of performance: feature or descriptor size (referring to the amount of memory needed per image) and description time per image (referring to the processing time needed to calculate the descriptor of an image). Together these measurements constitute the most important factors for a system's processing times, main memory and disk storage requirements. The measurements were performed on AMD A6-5200 APU 2.0 GHz machine running Linux Ubuntu 16.04.

To put this memory consumption into perspective, a duplicate image detection system without a dimensionality reduction model would require 30 GB of memory to store the descriptors of one million images a duplicate image detection system with 2048-dimension descriptor would require 12 GB of memory to store the descriptors of one million images, where a duplicate image detection system with 8-dimension descriptor would only need 100 MB. The result also shows that it takes almost the same time to generate different feature sizes i.e. there is no acquired computational overhead in automatically reducing feature size.

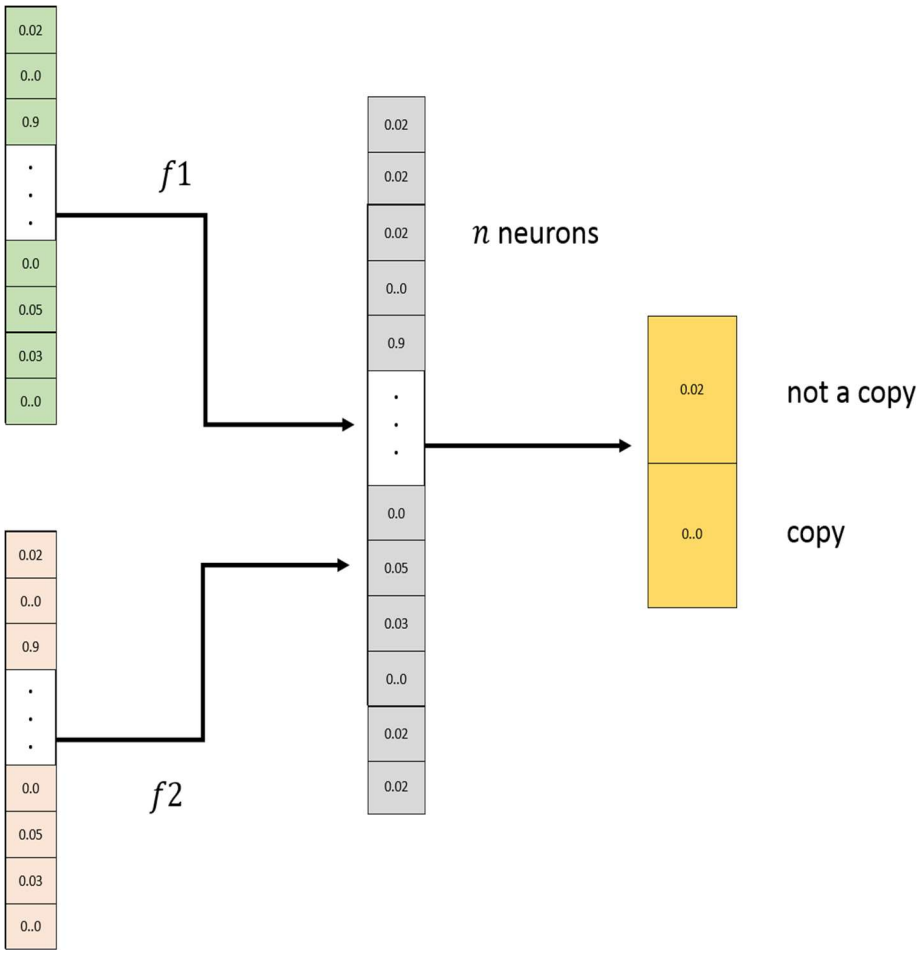

Figure 3: Copy checker

4. METHODOLOGY

To 9970 images were collected from four different datasets to serve as both a training and testing dataset. Several combinations were chosen to finally generate 28978 image pairs for the 'copy' image category which contains a variety of transformations such as scaling, distortions, noise adding and image blurring. The 'not a copy' category also had 14752 image pairs. Sixty percent of the total data for training, twenty percent of the total data for validation and twenty percent of the total data for testing. The dataset used include INRIA Copydays dataset [21], Image Collections [2], and MICC-F220, MICC-F2000 dataset from [22].

In table 1, a duplicate image detection system with 8-dimension descriptor uses 100 bytes, a duplicate image detection system with 16-dimension descriptor uses 190 bytes, a duplicate image detection system with 32-dimension descriptor uses 338 bytes, a duplicate image detection system with 64-dimension descriptor uses 620 bytes, a duplicate image detection system with 128-dimension descriptor uses 1322 bytes, a duplicate image detection system with 256-dimension descriptor uses 2419 bytes, a duplicate image detection system with 512-dimension descriptor uses 3854 bytes, a duplicate image detection system with 1024-dimension descriptor uses 6804 bytes, a duplicate image detection system with 2048-dimension descriptor uses 12456 bytes, a duplicate image detection system without the dimensionality reduction model uses 30753 bytes.

For the success of a method, its accuracy is of paramount importance: if accuracy is low then the method is not very useful even when it demonstrates excellent computational performance. a duplicate image detection system with 8-dimension descriptor has a duplicate detection accuracy of 90% with 46 misclassified image pairs, a duplicate image detection system with 16-dimension descriptor has a duplicate detection accuracy of 95.1% with 17 misclassified image pairs, a duplicate image detection system with 32-dimension descriptor has a duplicate detection accuracy of 96.5% with 1 misclassified image pairs, a duplicate image detection system with 64-dimension descriptor has a duplicate detection accuracy of 98.6% with 2 misclassified image pairs, a duplicate image detection system with 128-dimension descriptor has a duplicate detection accuracy of 98.9% with 0 misclassified image pairs, a duplicate image detection system with 256-dimension descriptor has a duplicate detection accuracy of 99.5% with 0 misclassified image pairs, a duplicate image detection system with 512-dimension descriptor has a duplicate detection accuracy of 99.7% with 0 misclassified image pairs, a duplicate image detection system with 1024-dimension descriptor has a duplicate detection accuracy of 99.5% with 0 misclassified image pairs, a duplicate image detection system with 2048-dimension descriptor has a duplicate detection accuracy of 99.7% with 0 misclassified image pairs, a duplicate image detection system without the dimensionality reduction model has a duplicate detection accuracy of 99.6% with 1 misclassified image pairs.

An ideal duplicate image detection system needs little time to extract features from images and the descriptor uses a minimal number of bytes. The accuracy of the method must also be very high. It is important to take the trade-offs between the various performances indicators into consideration and weigh them accordingly to an intended application need. If we evaluate the results with these trade-offs in mind, we can deduce that a duplicate image detection system with 256-dimension descriptor shows good overall accuracy especially considering that it only requires 2419 bytes for its image descriptors, 0 misclassified image pairs and 92.13% storage savings when compared to a duplicate image detection system without the dimensionality reduction model.

If good computational performance is the most important factor, a duplicate image detection system with 128-dimension descriptor is the best choice since it has a duplicate detection accuracy of 98.9% with 0 misclassified image pairs and 95.7% storage savings when compared to a duplicate image detection system without the dimensionality reduction model.

Table 1: Experiment Results

Number of output features	Accuracy %	Storage Savings (%)	Descriptor size (bytes)	Average Extraction Time (secs)	Number of misclassified image pairs
GoogLeNet + 8-dimension	90.0	99.67	100	0.63	46
GoogLeNet + 16-dimension	95.1	99.38	190	0.62	17
GoogLeNet + 32-dimension	96.5	98.90	338	0.61	1
GoogLeNet + 64-dimension	98.6	97.98	620	0.64	2
GoogLeNet + 128-dimension	98.9	95.70	1322	0.63	0
GoogLeNet + 256-dimension	99.5	92.13	2419	0.63	0
GoogLeNet + 512-dimension	99.7	87.47	3854	0.66	0
GoogLeNet + 1024-dimension	99.5	77.88	6804	0.68	0
GoogLeNet + 2048-dimension	99.7	59.50	12456	0.65	0
GoogLeNet	99.6	0.00	30753	0.61	1

5. CONCLUSION

In this paper we presented a duplicate image detection method based on convolutional neural networks, which learns a comparing function directly from raw image pixels. Images can be described and stored as dominant labels (extracted features). These labels appear robust to distortions, including re-encoding, resizing, cropping, and minor changes to coloration, and is accurate over the limited data set tested for this experiment, both in terms of identifying original and duplicate images. Several architectures are studied and each of them displays extremely good performance. These results indicate that convolutional neural networks based methods are specifically suited to duplicate image detection task and also show that to obtain high accuracy it is not necessary to use a large image descriptor. We also presented results of mismatched image pairs to gain further insight into the strength and weaknesses of the different architectures. This work provides proof of principle by building an accurate and robust-to-distortion duplicate image detection system using features extracted from images using pre-trained neural networks designed for image classification.

The analysis here focused on images that were tangentially related to the training sets used for building the pre-trained network used in extracting features from the images. A more robust solution would look to build a new deep neural network specifically suited to the problem-space using a broad sampling of images for training. Such a network would be better able to encode image-specific visual features. Performance-wise, initial tests showed a feature extraction processing time of about 0.68 seconds per image when using the CPU. Such performance is prohibitive for real-world application. In order to improve performance, the neural network processing would be adapted to use the GPU to drastically lower the per image extraction.

REFERENCES

[1] M. Meeker, “Internet Trends,” 2016.
[2] B. Thomee, M. J. Huiskes, E. M. Bakker, and M. S. Lew, “An Evaluation of Content-Based Duplicate Image Detection Methods for web search,” in IEEE International Conference onMultimedia and Expo (ICME), 2013.
[3] Y. H. Wan, Q. L. Yuan, S. M. Ji, L. M. He, and Y. L. Wang, “A Survey of the Image Copy Detection,” IEEE Conf. Cybern. Intell. Syst., 2008.
[4] M. Chen, Y. Li, Z. Zhang, C.-H. Hsu, and S. Wang, “Real-time, large-scale duplicate image detection method based on multi-feature fusion,” J. Real-Time Image Process., 2016.
[5] S. H. Srinivasan and N. Sawant, “Finding Near-duplicate Images on the Web using Fingerprints,” Proceeding 16th ACM Int. Conf. Multimed. MM 08, p. 881, 2008.
[6] J. Zhang, W. Zhu, B. Li, W. Hu, and J. Yang, “Image Copy Detection Based on Convolutional Neural Networks,” CCIS, pp. 111–121, 2016.
[7] F. Nian, T. Li, X. Wu, Q. Gao, and F. Li, “Efficient near-duplicate image detection with a local-based binary representation,” Multimed. Tools Appl., vol. 75, no. 5, pp. 2435–2452, 2016.
[8] E. Chang, J. Wang, C. Li, and G. Wiederhold, “RIME: A replicated image detector for the world-wide web,” SPIE Symp. Voice, Video, Data Commun., vol. 3527, pp. 58–67, 1998.
[9] S.-A. Berrani, L. Amsaleg, and P. Gros, “Robust content-based image searches for copyright protection,” in Proceedings of the first ACM international workshop on Multimedia databases - MMDB 2003, 2003, no. January.
[10] C. Yang, J. Peng, and J. Fan, “Speed-up Multi-modal Near Duplicate Image Detection,” Proc. Second Int. Conf. Internet Multimed. Comput. Serv., vol. 2013, no. March, pp. 95–98, 2010.
[11] Y. Jinliang, W. Xiaohua, and W. Rongbo, “Copy Image Detection Based On Local Keypoints,” pp. 258–262, 2011.
[12] S. Zhou, J. Li, J. Xing, W. Hu, and J. Yang, “Non-negative Sparse Coding Using Independent Multi-Codebooks for Near-Duplicate Image Detection,” pp. 152–159, 2013.
[13] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì, “Aligning codebooks for near duplicate image detection,” Multimed. Tools Appl., vol. 72, no. 2, pp. 1483–1506, 2014.

[14] L. Yan, H. Ling, F. Zou, and C. Liu, "Iterated local search optimized hashing for image copy detection," Multimed. Tools Appl., vol. 74, no. 21, pp. 9729–9746, 2014.
[15] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui, "USB: Ultrashort binary descriptor for fast visual matching and retrieval," IEEE Trans. Image Process., vol. 23, no. 8, pp. 3671–3683, 2014.
[16] S. Hsieh, C.-C. Chen, and C. Chen, "A novel approach to detecting duplicate images using multiple hash tables," Multimed. Tools Appl., vol. 74, no. 13, pp. 4947–4964, 2015.
[17] L. Yan, F. Zou, R. Guo, L. Gao, K. Zhou, and C. Wang, "Feature aggregating hashing for image copy detection," World Wide Web, vol. 19, no. 2, pp. 217–229, 2016.
[18] W. Zhao, H. Luo, J. Peng, and J. Fan, "MapReduce-based clustering for near-duplicate image identification," Multimed. Tools Appl., 2016.
[19] E. Gadeski, H. Le Borgne, and A. Popescu, "Fast and robust duplicate image detection on the web," Multimed. Tools Appl., 2016.
[20] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," arXiv Prepr. arXiv 1409.4842, pp. 1–9, 2014.
[21] H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search – Extended version –," Eccv, vol. 5302, no. October, pp. 304–317, 2008.
[22] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," IEEE Trans. Inf. Forensics Secur., vol. 6, no. 3 PART 2, pp. 1099–1110, 2011.