CNN based near-duplicate image detection used for email spam detection

Vít LISTÍK[1]

[1]Dept. of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague, Technická 2, 166 27 Praha, Czech Republic

listivit@fel.cvut.cz

Abstract. *Near-duplicate image detection is used for searching similar images but may also be used for spam detection. Near-duplicate images may be blacklisted as image spam or number of occurrences of the nearly same image may be calculated. We propose using features extracted from the convolutional neural network for the detection task. We also propose how to convert the vector to a much smaller hash and we compare those methods to previously used methods. We did test our solution for 1000 publicly available images. We did 5 alternations on each image from the dataset (blur, noise, crop, rotate, brightness). Then we tested all the algorithms for searching if the original image is the closest one to the altered images. Our proposed solution performed significantly better for this task. Other algorithms were having a lot of false positives for cropped and rotated images.*

Keywords

Image, Near-duplicate detection, Spam, ResNet, Convolutional Neural Network (CNN)

1. Introduction

In this work, we want to tackle the problem of email spam. Image spam in particular [12]. Spam emails are defined as unsolicited messages which are usually sent in bulks. Image spam refers to images contained in spam messages. This technique was first used to hide from the anti-spam filter (which was not doing image analysis). Nowadays images are very common part of email communication, therefore, they are used for anti-spam with text and other information which may be extracted from emails.

Duplicate detection is an essential part of the anti-spam system because spam messages are sent in bulks. But the messages may contain slight alternations, making exact duplicate detection inefficient. That is why near-duplicate detection is used [16]. Near-duplicates may be detected based on previously gathered samples (blacklist) or they may be transformed to different representation (usually simpler) by which they are clustered and the number of their occurrences over some fixed time window is monitored.

2. State of the art

Near-duplicate image, detection is used for image search, copyright enforcement or spam detection for websites or emails [15]. Image duplicates may be detected by cryptographic hashing the same as for any other data. Near-duplicate detection is much more complicated because similar objects should have same hash or should be nearby some given metric, for which the cryptographic hash is totally unsuitable because by its definition a small variance in input should cause a huge difference in the output. Near-duplicate image detection is usually based on image analysis (like histograms or wavelengths analysis) or locality-sensitive hashing or scale-invariant feature transform (SIFT) [3, 6, 4]. Sift of often superior to the other techniques because it is using higher level features extracted from the image, which on the other hand may affect performance [11].

3. Methods

We are using several image hashing methods. Reference methods are implemented in the ImageHash library [1] and our proposed method is based on convolutional neural network (CNN) and is described in the following chapters.

3.1. Average hashing

Average hashing is the simplest algorithm from Imagehash library and it is also the quickest one [2]. This algorithm is similarly as others based on the idea of reducing high frequencies which carry detail and keeping only low frequencies. The algorithm is the following:

1. Reduce image size (Bilinear) to 8x8 (64) pixels
2. Covert to grayscale

[1] https://github.com/JohannesBuchner/imagehash

3. Compute average color of the whole image
4. Bits are computed based on pixel values which are thresholded by the average to 0 and 1 (64-bit integer)

This algorithm should be immune to scaling and brightness and contrast changes.

3.2. Perceptual hash

Perceptual hash or pHash is described in [2]. This algorithm is based on discrete cosine transform (DCT):

1. Optional size and color reduction (because of speed)
2. Compute the DCT which transform the image to wave spectrum (used in JPEG)
3. Reduce high frequencies from the image
4. Compute average value and do the thresholding (same as for average hashing)

3.3. Difference hash

Difference hash or dHash is described in [1]. This algorithm is based on gradient direction:

1. Size and color reduction (same as for average hashing)
2. Compute relative gradient direction (difference of adjacent pixels)
3. Bits are computed based on the brightness of the neighbor

3.4. Wavelet hash

Wavelet hash or whash is based on Haar wavelet which is based on Fourier analysis. This approach does convert the image to wave spectrum in which it is possible to reduce high frequencies directly.

3.5. Convolutional neural networks

Convolutional Neural networks (CNN) are a special case of artificial neural networks. Those networks are composed of neurons which are not in linear layers but they form filters. Those filters are trained for a specific task. This approach is commonly used for image processing tasks [9]. CNNs are used because of the ability to compress the information from multi-dimensional data like images and because of their high performance. Those networks may be used thanks to raising the computing power of GPUs. Thanks to them the state of the art results for image classification and object detection changed significantly [13]. CNNs trained for classification task on millions of images are great feature extractors also for other tasks. The image features are extracted from the previous to last network layer and are often called semantic feature vector.

3.6. Resnet

One of the examples of those deep CNNs is ResNet [5]. This network architecture won the ImageNet classification challenge in 2015 [14]. The architecture is using residual connections which made training of network this deep possible.

3.7. CNN hash

We propose using a semantic vector extracted from the CNN as a hash [7]. This method should be invariant to alternations because the network is trained to compress important visual information to the extracted feature vector.

We did also propose how to create a universal representation from this vector. The binarization is inspired by the average hash. We did the binarization based on fixed threshold 0.5 and also based on the mean. The extracted vector has length 2048, therefore it is space inefficient and keeps too much information. We also compress this binarized vector with a technique similar to max pooling. We took windows of size 2, 4, 8, 16, 32, 64 and searched for at least one 1 (max reduction) and looked if the average is above 0.5 (average reduction).

3.8. Distance metrics

We need some way to do define the distance of two image representations (hashes). There are two metrics used in this work.

Hamming distance - is a metric used for a distance of strings. This metric computes the number of changes needed for inputs of the same length to make them equal. This metric is used in ImageHash library for comparison of the bits of the 64-bit integer.

Cosine distance - Is used for real-valued high dimensional vectors. This metric ignores the magnitude of the vectors. The computation of cosine distance is described in Eq. 1.

$$\cos(\Theta) = \frac{A.B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

3.9. Implementation

Our implementation may be found at Github platform[2] as an open-source. Implementation is in Python language and is runnable via Docker compose.

4. Experimental results

We performed several experiments on publicly available images and their alternations which show the performance of described hashing methods.

4.1. Dataset

We used 1000 first images from OpenImages database [10, 8]. The dataset consists of photos which may be used for classification or object recognition. We used this dataset because it is very general and will be similar to the email traffic which would be much harder to obtain because of the email private nature.

4.2. Image alternations

We did alternate the images for testing the robustness of the hashing approach. The alterations were the following:

- *Blur* - Gaussian blur of the image pixels with radius 2
- *Crop* - Random crop of the image not smaller then 40% of the original
- *Brightness* - Making the image lighter or darker up to 80%
- *Rotate* - Image rotation up to 30 deg
- *Noise* - Adding Gaussian noise to the image

4.3. Similarity detection

We did test if the alternations of the image are the closest ones to the original image. The images were sorted based on the distance metric and if the alternation was found after some other images the score was lowered proportionally to the number of false positive images (0-1) shown in Tab. 1.

We also evaluated each alternation separately. The ordering for the CNN is unique, the exact ordering for other hashing methods may be uncertain, therefore the alternated images "compete" with each other for CNN that may share the same spot for the other methods. The results are shown in Tab. 2

[2] https://github.com/tivvit/image-duplicate-detection-eval

method	score
cnn	0.98
average hash	0.10
dHash	0.04
pHash	0.03
wHash	0.09

Tab. 1. Score in range 0-1 for searching image alternations

method	Blur	Crop	Bright	Rotate	Noise
cnn	2.39	4.44	1.65	2.97	4.03
avg hash	1.05	802.00	2.68	350.36	1.63
dHash	1.20	1843.44	3.00	559.65	3.52
pHash	1.11	1761.22	2.39	756.63	1.69
wHash	1.07	755.39	1.87	310.55	1.39

Tab. 2. Average positions of the alternations for the hashing methods.

4.4. Hash representation

We tested if the altered image hash representation is an exact match with the original image representation. The results are shown in Tab 3.

method	Match
dHash	33.2%
pHash	42.5%
average hash	44.5%
wHash	50.7%
ab8	0.4%
aba16	0.1%
ab16	2.5%
mb16	5.5%
mba16	5.9%
ab32	6.3%
aba32	7.0%
ab64	10.1%
mb32	12.3%
aba64	12.5%
mba32	13.0%
mb64	13.1%
mba64	13.1%

Tab. 3. Results for number of exactly matching hashes. Where ab8 means average reduced windows of size 8 for threshold binarization (*ba - average binarization, m* - max reduction).

5. Conclusions and future work

We show the performance of near duplicate image detection algorithms on 1000 images. Each of the images was altered with 5 operations (blur, noise, crop, rotate, brightness). We did search the closest images to the original image and shown that proposed CNN solution with score 0.98 (max

1) is significantly better for that task then simpler similarity detection algorithms which scored best 0.1. The feature vector extracted from CNN was performing almost the same for all alternations. Other hashing algorithms were affected by rotation and mostly by crop. We did also test the exact match of the computed hash from the altered image to the hash computed from the original image. Our simple method for binarization and reduction the feature vector achieved up to 13% match while hashing algorithms achieved up to 50%.

Our proposed solution of an image near-duplicate detection using CNN extracted feature vector is able to perform well for known samples. When we want to use the image hash as an identifier it is better to use wHash. Both methods are usable for anti-spam detection. The hashing solution may be used for counting the number of occurrences of the images in the traffic and the solution using CNN feature vector may be used for the exact match. The exact match may be used for blacklisting or a tree structure with image representations may be used for creating image buckets for counting occurrences.

Acknowledgements

The research described in the paper was supervised by Prof. V. Hlaváč and J. Šedivý CSc. CIIRC in Prague.

References

[1] Kind of like that - the hacker factor blog.

[2] Looks like it - the hacker factor blog.

[3] Ondrej Chum, James Philbin, Andrew Zisserman, et al. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, volume 810, pages 812–815, 2008.

[4] Wei Dong, Zhe Wang, Moses Charikar, and Kai Li. High-confidence near-duplicate image detection. In *Proceedings of the 2nd acm international conference on multimedia retrieval*, page 1. ACM, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Yan Ke, Rahul Sukthankar, Larry Huston, Yan Ke, and Rahul Sukthankar. Efficient near-duplicate detection and sub-image retrieval. In *ACM multimedia*, volume 4, page 5. Citeseer, 2004.

[7] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *International conference on multimedia modeling*, pages 251–263. Springer, 2017.

[8] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[11] David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pages 1150–1157, 1999.

[12] Bhaskar Mehta, Saurabh Nangia, Manish Gupta, and Wolfgang Nejdl. Detecting image spam using visual features and near duplicate detection. In *Proceedings of the 17th international conference on World Wide Web*, pages 497–506. ACM, 2008.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[15] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. Robust image hashing. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 664–666. IEEE, 2000.

[16] Zhe Wang, William K Josephson, Qin Lv, Moses Charikar, and Kai Li. Filtering image spam with near-duplicate detection. In *CEAS*, 2007.