IC14

CNN BASED NEAR-DUPLICATE IMAGE DETECTION USED FOR EMAIL SPAM DETECTION

Vít Listík (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)

TASK

Cluster images extracted from email traffic stream. Clustering should be supine to slight alternations, which are used by spammers.

METHODS

Because the task requires clustering to unknown number of clusters we are searching for image hash representation which will define a cluster.
We are using ImageHash library as a baseline. This library offers first 4 hashes in the following list.

- **Average hashing** - Reducing high frequencies. Bilinear downscale (8x8), then convert to grayscale and compute average intensity. Compute bits based on threshold (more or less then average).
- **Perceptual hash** - Based on discrete cosine transform. Reduce high frequencies. Compute average value and do the thresholding.
- **Difference hash** - Based on gradient direction (difference of adjacent pixels). Bits are computed based on the brightness of the neighbour.
- **Wavelet hash** - Based on Haar wavelet which is using Fourier analysis.
- **CNN hash** - Our proposed hash method. We are using ResNet for feature vector (hash) extraction. We are using vector of length 2048 extracted from the fully-connected layer after the convolution.

CNN hash vector is originally 2048 float numbers, which makes the hash space too big for our use case. We propose using binarization inspired by Average hashing. We used fixed threshold 0.5 and average threshold. We used method similar to max pooling with window size 2, 4, 8, 16, 32 and 64 and looked if 1 is present (max reduction) and if average is above 0.5 (average reduction)

We did several alternations on each image to simulate alternations done by spammers.

- **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

Distance metrics used for measuring the closest image.

- **Hamming** - 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.
- **Cosine** - Used for real-valued high dimensional vectors such as our proposed CNN based hash. This metric ignores the magnitude of the vectors.

Python implementation may be found at https://github.com/tivvit/image-duplicate-detection-eval

ACKNOWLEDGEMENTS

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

EXPERIMENTAL RESULTS

Dataset

We used 1000 first images from OpenImages database.

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 and individual alternations in Tab. 2.

Method	CNN	Average hash	pHash	pHash	wHash
Score	0.98	0.10	0.04	0.03	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 individual hashing methods.

Hash representation

We compared hash representations of the altered and original images and expected exact matches. The results may be seen in Tab. 3, for more detailed results please see our paper.

Method	dHash	pHash	average hash	wHash
Match	33.2%	42.5%	44.5%	50.7%

Method	ab8	ab16	mba16	aba32	mba32	mba64
Match	0.4%	2.5%	5.9%	7.0%	13.0%	13.1%

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).

Conclusion

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%. We propose using exact match for blacklisting or a tree structure with image representations which may be used for creating image buckets for counting occurrences.