Deep Residual Hashing Network for Image Retrieval

Edwin Jimenez-Lepe and Andres Mendez-Vazquez

CINVESTAV Guadalajara {eejimenez,amendez}@gdl.cinvestav.mx
http://www.gdl.cinvestav.mx

Abstract. Conventional methods in Content-Based Image Retrieval use hand-crafted visual features as input but sometimes such feature vectors do not preserve the similarity between images. Taking advantage of the improvements in the Convolutional Neural Networks (CNN) area we propose a deep learning method that generates binary hash codes based on the features learned. We explode a previous proposed idea in the field: in a supervised manner the binary codes can be learned adding an extra hidden layer to represent the main features that identifies the classes in a database. The experimental results outperforms the state-of-the-art hashing algorithms on the CIFAR-10 dataset. Its remarkable that the proposed neural network has 8 million parameters (DRHN-15) less than other CNN based methods e.g. AlexNet (a widely used model in CV tasks) have 60 million.

Keywords: Convolutional Neural Networks, Content-Based Image Retrieval, Computer Vision, Deep Learning

1 Introduction

The ever-growing information generated and shared on the web leads to an amazing field of oportunity when we search for different kinds of data. The study of image retrieval based on text began in 1970. Some problems emerged like the cost of manual annotation for database images [21] and limited capability of using a restricted number of words to describe the content of an image.

The study of Content-Based Image Retrieval (CBIR) began in 1990, with images indexed by their visual features, such as texture and color [25]. In this context a strategy was to use global descriptors to measure similarity between the query and images in the database, nevertheless such descriptors are susceptible to changes as illumination, occlusion, intersection, and truncation [25]. In 2003 the BoW [19] model began to gain popularity in the area. However, in 2012 [10] outperforms previous results in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using Convolutional Neural Networks (CNN), consequently there was an increased interest in deep learning methods. Therefore, this work presents a new approach to CBIR problem called Deep Residual Hashing Network (DRHN). The rest of this paper is organized as follows. Related work is presented in Section 2. The proposed model DRHN is explained in Section 3.

Section 4 presents the experimentation and results of the comparison to supervised and unsupervised methods in the CIFAR-10 dataset. Finally, Section 5 exposes the conclusions of the research.

2 Related Work

When we talk about the generation of hash codes for image retrieval we can classify the methods as [12]:

- Supervised, which uses extra information like class labels or calculating pairwise similarities. For example: Binary Reconstruction Embedding (BRE) [11], Minimal Loss Hashing (MLH) [16] and Supervised Hashing with Kernels (SHK) [15].
- Unsupervised, which uses just data itself. For example: Spectral Hashing (SH) [23], Locality Sensitive Hashing (LSH) [3] and Iterative Quantization (ITQ) [4].

Most of the supervised methods use hand-crafted visual features as input to represent images (e.g., GIST [17]). Nevertheless, with the introduction of CNN for image classification [10, 18, 20] and retrieval [14, 12, 24] the use of raw image pixels as input became usual. Therefore, the risk of discard useful information disappears.

3 Methodology

The most widely used architecture of CNN [10, 13] uses convolution, ReLU and max pooling as the basic sequence of layers and repeat it until they have a fully conected layer at the end of the model. But [7] proposed an architecture where pooling layers are substituted by little convolutions with filters of size 3×3 and *stride* of 2. This emulates subsampling and extracts features in the same process. Therefore, achieving state-of-the-art results in image classification with a deeper and simpler model than previous architectures [10, 20, 5].

3.1 Deep Residual Hashing Network

We propose a novel method called Deep Residual Hashing Network (DRHN). The base of our model is a residual block [7] which is formed by the following operations:

- Convolution as is presented in Eq. 1, where w is a square filter of size m with c channels and x is the input with a zero padding set to one 1, this is performed without kernel rotation.
- Rectified linear unit (ReLU), is shown in Eq. 2, where x is the input.
- We use the Batch normalization as defined in [8], and
- Element-wise addition (Add) is defined in Eq. 3. Is repeated for all channels c and positions ij in matrices of same dimensions x and z.

$$y_{ij} = \sum_{c=0}^{C-1} \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{abc} x_{(i+a)(j+b)c} \tag{1}$$

$$y_{ij} = max\{x_{ij}, 0\} \tag{2}$$

$$y_{cij} = x_{cij} + z_{cij} \tag{3}$$

$$y = Add(BN_{\gamma_2,\beta_2}(conv(BN_{\gamma_1,\beta_1}(Relu(conv(x, w1))), w2)), x) \tag{4}$$

Table 1. Definition of the layers of DRHN

Layer	Convolution dimensions	Output dimensions
Input		3x32x32
$BN_{\gamma,\beta}(Relu(conv(x, w1)))$	16x3x3x3	16x32x32
Residual Group (n)	16x16x3x3	16x32x32
Residual Group (n) with increase dimension	*32x16x3x3, 32x32x3x3	32x16x16
Residual Group (n) with increase dimension	*64x32x3x3, 64x64x3x3	64x8x8
Residual Group (n)	64x64x3x3	64x8x8
Residual Group (n)	64x64x3x3	64x8x8
Residual Group (n) with increase dimension	*128x64x3x3, 128x128x3x3	128x4x4
Average Pooling Layer		128
Hash Layer		h
Fully Connected Layer with Softmax		10

A Residual Group is the join of n Residual Blocks, which are defined in Eq. 4. The Average Pooling Layer calculates the average of all values in a channel. Hash Layer is defined in Subsection 3.2. The output of the model is a fully connected layer with Softmax function. The architecture of the proposed model is shown in Figure 1. Table 1 details the layers of the model indicating the name of the layer (or group of layers), the dimension of the related filters (number of filters $\times$ channels $\times$ height $\times$ width) and the dimensions of the output (channels $\times$ height $\times$ width). When the number of channels in the output is increased, the first convolution of the block is performed with *stride* of 2 and the dimensions indicated with * in Table 1. Otherwise convolution is performed with *stride* of 1. Element-wise addition at the end of a block with increase dimention is possible if an identity shortcut is performed [7].

3.2 Hash Layer

The Hash Layer (H) is a fully conected layer with a sigmoid activation function s:

$$H = s(Wx + b) \in \mathbb{R}^h \tag{5}$$

where $x \in \mathbb{R}^d$ is the input of the layer, $b \in \mathbb{R}^h$ is a bias and $W \in \mathbb{R}^{h \times d}$ represent the weights that connect the units of x with H.

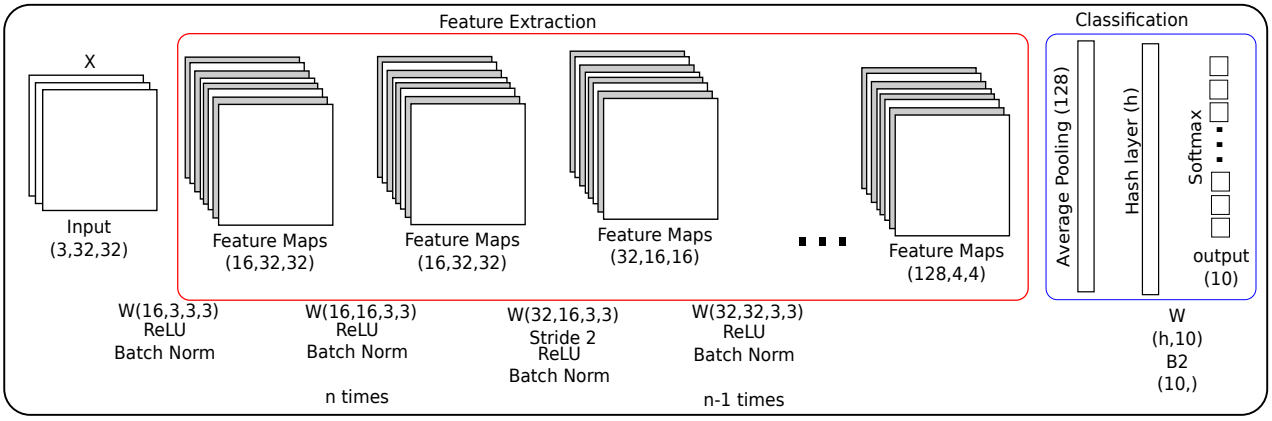

Fig. 1. The model for Deep Residual Hashing Network

Given an image $I \in \mathbb{R}^{c \times z \times w}$, where c represents the channels of the image, z the height and w the width, the layers of the model from Input layer to H form a hash function that performs the mapping from $\mathbb{R}^{c \times z \times w}$ to $\mathbb{R}^h$.

Consequently, to obtain the binary code related to I as described by [14], we extract the output of H, and binarize the activation by a threshold to obtain the correspondent code. For each element in H we apply the sign function:

$$sign(H^i) = \begin{cases} 1, \text{if } H^i > 0.5 \\ 0, \text{otherwise } . \end{cases} \tag{6}$$

Once we have the binary codes we can perform image retrieval as follows: Let $IM = \{I_1, I_2, \ldots, I_n\}$ and $IM_H = \{H_1, H_2, \ldots H_n\}$ denote the dataset with n images and the corresponding binary codes, respectively. Given a query image I_q and its binary code H_q we can return the elements of IM where the Hamming distance between H_q and $H_i \in IM_H$ is lower than a threshold, or we can return the top-k if we need a specific k number of images returned by the query.

A summary of how the model is used to generate the binary codes and retrieve similar images is shown in Figure 2

4 Experiments and Results

In this section, we show the evaluation of the proposed method on the next dataset:

– **CIFAR-10 Dataset** [9] consists of 60,000 color images divided in 10 classes, each one with 6,000 images of 32x32 pixels. The dataset is splitted into training and validation set, 50,000 and 10,000 images respectively. We perform augmentation of the dataset by mirroring the images.

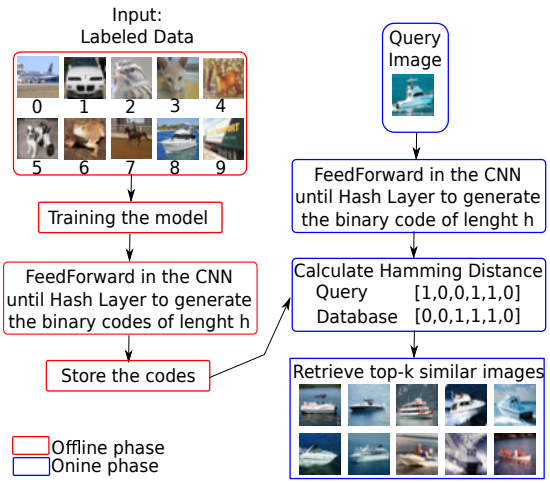

Fig. 2. Brief overview of the proposal.

We compare DRHN with n=15 against three unsupervised methods LSH [3], SH [23] and ITQ [4], and ten supervised methods CNNH[24], CNNH+ [24], KSH [15], MLH [16], BRE[11], ITQ-CCA [4], DNNH [12], DHN [26], CNNBH [6] and DLBHC [14].

After that, we compare the precision (Eq. 7) when our proposal have a variation in the depth using n=5,9,10,11 and 15.

We used the implementation of [7][1] in Lasagne [2] [22] as base to build and train the proposed model.

4.1 Evaluation Metrics

According to [1] and [14] we use a ranking based criteria as evaluation metric. Given a query (image in our case) q and a similarity measure (Hamming Distance), we can assign a rank for each dataset image. We calculate the precision of the top k ranked images with respect to a query q as $P@K$ using the indicator function $X(i)$:

$$P@K = \frac{\sum_{i=1}^{k} X(i)}{k} \tag{7}$$

$$X(i) = \begin{cases} 1, \text{if } L(q) = L(i) \\ 0, \text{otherwise} \end{cases} \tag{8}$$

where $L(q)$ denotes query image q label and $L(i)$ denotes ith ranked image label. Also we used mean average precision (mAP) which is a standard evaluation

[1] `https://github.com/Lasagne/Recipes/blob/master/papers/deep_residual_learning/Deep_Residual_Learning_CIFAR-10.py`

metric in CBIR to perform a fair comparison:

$$mAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{m_q}\sum_{k=1}^{m_q} P@K \text{ (if } k^{th} \text{ item was relevant)} \tag{9}$$

where $|Q|$ represent the number of queries and m_q is the number of result images for a given query q.

4.2 Results on CIFAR-10 dataset

Performance of Image Classification: We trained the model for the the image classification task. After the 43th epoch we obtained an accuracy of 93.70% (which is slightly better than the original implementation accuracy ≈93.25%). It means that the binary layer does not affect in a negative way the performance of the model.

Performance of Image Retrieval: The performance of image retrieval can be seen at Figure 3[2]. We plot the results of retrieving relevant images using 48 bits binary codes and Hamming distance between the query image q and the ith retrieved image. Our approach achieves better perfomance than other methods (supervised and unsupervised). It obtains a 92.91% precision retrieving 1000 images, which improves by almost 3% the performance compared to [14].

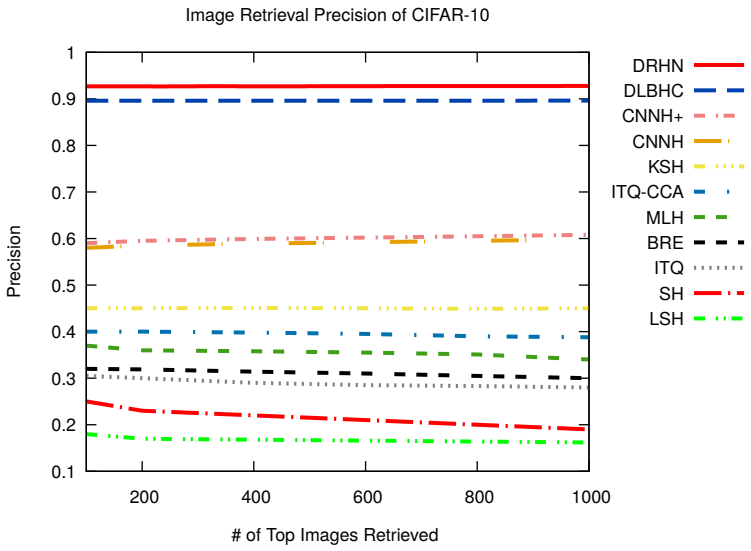

Fig. 3. Image retrieval precision with 48 bits on CIFAR-10 dataset.

[2] We thank to [14] for the repository with the information available for Figure 3 and Table 2.

In Table 2, we compare the mean average precision (mAP) of some hashing methods at different number of bits when we retrieve the top 1000 images. Figure 5 shows the retrieval results for a ship image query. The relevant images have the same label and similar appearance. Something remarkable is the absence of false positive results.

Table 2. mAP comparison of different hashing methods on CIFAR-10 dataset.

Method	12 bits	32 bits	48 bits
DRHN-15	**92.65**	**92.23**	**92.91**
DLBHC	89.3	89.72	89.73
CNNBH	53.2	61.0	61.7
DHN	55.5	60.3	62.1
DNNH	55.2	55.8	58.1
CNNH+	46.5	52.1	53.2
CNNH	43.9	50.9	52.2
KSH	30.3	34.6	35.6
ITQ-CCA	26.4	28.8	29.5
LSH	12.1	12.0	12.0

Table 3. mAP comparison of different depth in our method on CIFAR-10 dataset.

Method	12 bits	24 bits	32 bits	48 bits
DRHN-16	92.53	92.32	92.15	92.25
DRHN-15	92.65	92.02	92.23	**92.91**
DRHN-14	91.78	**92.74**	92.93	92.19
DRHN-13	92.52	92.58	92.18	92.30
DRHN-12	91.55	92.71	92.78	92.89
DRHN-11	**92.66**	92.35	92.57	92.57
DRHN-10	91.93	92.55	**92.94**	91.95
DRHN-9	91.97	91.92	92.35	92.58
DRHN-8	92.17	92.03	92.49	91.95
DRHN-7	91.76	91.74	92.30	92.50
DRHN-6	91.85	92.49	91.36	92.23
DRHN-5	91.75	91.21	91.21	91.89
DRHN-4	91.32	90.83	91.43	90.71
DRHN-3	90.81	91.04	90.99	90.78
DRHN-2	90.40	90.22	90.62	90.66
DRHN-1	87.50	88.01	88.59	88.43

Variation of the Depth in the Model We variate the depth of the model using n=[1, . . . , 16] and the number of bits in the binary code using h=[12,24,32,48]. In Table 3, the mAP is shown. Also in Figure 4, we present how the precision with respect to the number of retrieved images changes.

Results on CIFAR-100 We extended the experiments on CIFAR-100 which contains the same images that CIFAR-10 but with one hundred different labels, to see if the mehtod is scalable for datasets with more classes. The results are shown in Table 4 and Figure 6.

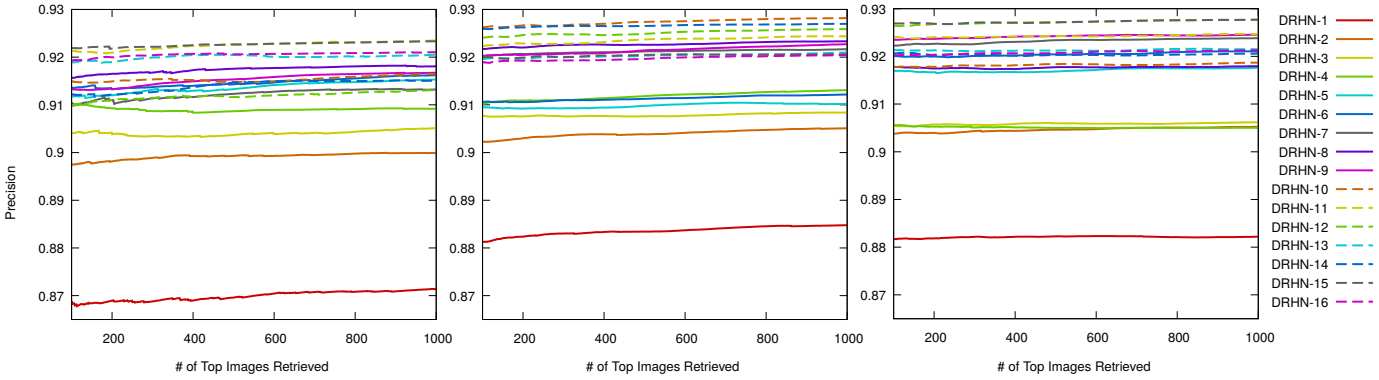

Fig. 4. Image retrieval precision with 12, 32 and 48 bits on CIFAR-10 dataset.

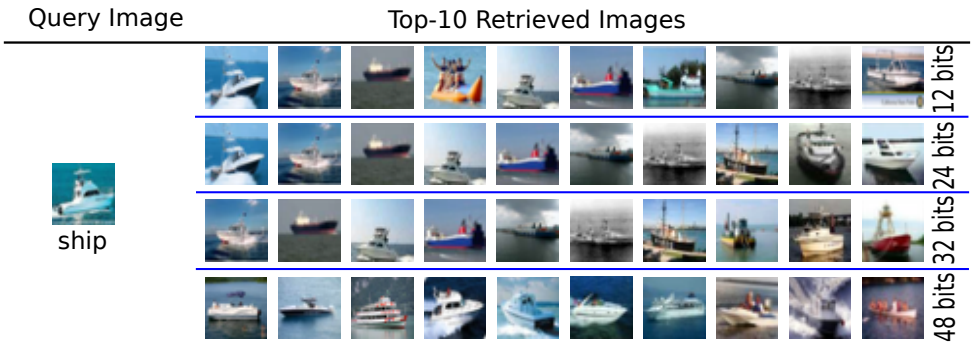

Fig. 5. Top 10 retrieved images from CIFAR-10 with different bit numbers.

Table 4. mAP comparison of different depth in our method on CIFAR-100 dataset.

Method	12 bits	24 bits	32 bits	48 bits
DRHN-16	54.51	**62.37**	61.17	62.08
DRHN-15	56.78	59.19	62.18	61.56
DRHN-14	56.89	60.13	61.84	61.55
DRHN-13	55.66	59.88	61.03	61.56
DRHN-12	**57.05**	60.35	**62.57**	61.96
DRHN-11	55.01	60.33	60.87	61.03
DRHN-10	54.65	61.17	60.84	59.71
DRHN-9	55.28	61.69	61.59	60.96
DRHN-8	56.16	59.35	60.64	**62.26**
DRHN-7	55.98	59.97	60.01	60.59
DRHN-6	55.30	61.02	60.08	60.22
DRHN-5	53.11	59.08	59.15	58.48
DRHN-4	53.69	57.86	59.06	58.51
DRHN-3	53.41	55.54	55.85	56.83
DRHN-2	50.73	54.06	53.01	53.73
DRHN-1	45.23	46.08	46.26	45.78

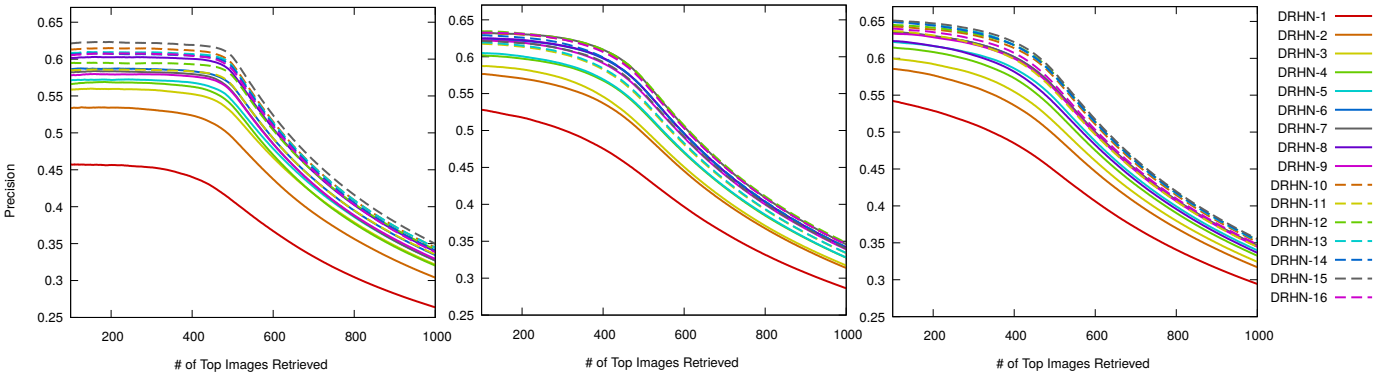

Fig. 6. Image retrieval precision with 12, 32 and 48 bits on CIFAR-100 dataset.

5 Conclusions

In this paper, we proposed a deep model of CNN based on the work of [7] to generate hashing codes in a supervised manner, allowing an efficient search of images based on their content. This method takes fully advantage of data labels (instead of requiring a triplet of images as other methods do) and raw data thanks to the properties of CNN. Experimental results show that we can outperform several state-of-the-art results on image retrieval on a popular dataset (CIFAR-10 and CIFAR-100).

Acknowledgments. We appreciate the financial support given by CONACYT.

References

1. Deng, J., Berg, A.C., Fei-Fei, L.: Hierarchical semantic indexing for large scale image retrieval. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 785–792. IEEE (2011)
2. Dieleman, S., et al.: Lasagne: First release. (Aug 2015), `http://dx.doi.org/10.5281/zenodo.27878`
3. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: VLDB. vol. 99, pp. 518–529 (1999)
4. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(12), 2916–2929 (2013)
5. Graham, B.: Spatially-sparse convolutional neural networks. arXiv preprint arXiv:1409.6070 (2014)
6. Guo, J., Li, J.: Cnn based hashing for image retrieval. arXiv preprint arXiv:1509.01354 (2015)

7. 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. pp. 770–778 (2016)
8. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
9. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097--1105 (2012)
11. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: Advances in neural information processing systems. pp. 1042–1050 (2009)
12. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3270–3278 (2015)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
14. Lin, K., Yang, H.F., Hsiao, J.H., Chen, C.S.: Deep learning of binary hash codes for fast image retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 27–35 (2015)
15. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2074–2081. IEEE (2012)
16. Norouzi, M., Blei, D.M.: Minimal loss hashing for compact binary codes. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 353–360 (2011)
17. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision 42(3), 145–175 (2001)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Sivic, J., Zisserman, A., et al.: Video google: A text retrieval approach to object matching in videos. In: iccv. vol. 2, pp. 1470–1477 (2003)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)
21. Tamura, H., Yokoya, N.: Image database systems: A survey. Pattern recognition 17(1), 29–43 (1984)
22. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688 (May 2016), `http://arxiv.org/abs/1605.02688`
23. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in neural information processing systems. pp. 1753–1760 (2009)
24. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: AAAI. vol. 1, p. 2 (2014)
25. Zheng, L., Yang, Y., Tian, Q.: Sift meets cnn: a decade survey of instance retrieval. arXiv preprint arXiv:1608.01807 (2016)
26. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: AAAI. pp. 2415–2421 (2016)