Supervised Hashing Binary Code with Deep CNN for Image Retrieval

Jun-yi Li,
Shanghai Jiaotong University Electrical and Electronic Engineering College
Shanghai, China

Jian-hua,Li
Shanghai Jiaotong University Electrical and Electronic Engineering College
Shanghai, China

Abstract—***Approximate nearest neighbor search is a good method for large-scale image retrieval. We put forward an effective deep learning framework to generate binary hash codes for fast image retrieval after knowing the recent benefits of convolutional neural networks (CNNs). Our concept is that we can learn binary codes by using a hidden layer to present the latent concepts dominating the class labels when the data labels are usable. CNN also can be used to learn image representations. Other supervised methods require pair-wised inputs for binary code learning. However, our method can be used to learn hash codes and image representations in a point-by-point manner so it is suitable for large-scale datasets. Experimental results show that our method is better than several most advanced hashing algorithms on the CIFAR-10 and MNIST datasets. We will further demonstrate its scalability and efficiency on a large-scale dataset with 1 million clothing images.***

Keywords- convolutional neural networks; nearest neighbor search; hidden layer; LSH; supervised learning;

I. INTRODUCTION

Image retrieval based on content aims at searching similar images through the analysis on image content so image representations and similarity measure become very important for such a task. In this research, one of the most challenging issues is about the pixel-level information to the semantics from human perception [25, 27]. Although several manual features have been proposed to represent the images [19, 2, 22], performance of these visual descriptors is still not very good before the recent breakthrough in deep learning. Recent studies [14, 7, 21, 23] have shown that deep CNN greatly makes the performance on various vision tasks such as object detection, image classification and segmentation better. These accomplishments are due to the ability of deep CNN to learn the rich mid-level image representations.

During learning of rich mid-level image descriptors, Krizhevsky et al. [14] used the feature vectors from the 7th layer in image retrieval and showed excellent performance on ImageNet. However, because the features of CNN are high-dimensional and it is not good enough to directly calculate the similarity between two 4096-dimensional vectors, Babenko et al. [1] suggest to use PCA and distinctive dimensionality reduction to make the features of CNN compact, and finally they obtain a good performance.

In CBIR, both image representations and computational cost are important. Due to the recent growth of visual contents, people need rapid search in a large database. Many studies aim at how to efficiently retrieve the relevant data from the large-scale database. Because of the high-computational cost, traditional linear search (or exhaustive search) is inappropriate for searching in a large corpus. Instead of linear search, using the technique of Approximate Nearest Neighbor (ANN) or hashing based method for speed becomes a practical way [6, 29, 18, 20, 15, 30]. These methods reflect the high-dimensional features to a lower dimensional space, which will generate the compact binary codes. Thanks to the produced binary codes, fast image search is workable via binary pattern matching or Hamming distance measurement. The result is the dramatic reduction of the computational cost and further optimization of the efficiency of the search. Some of these methods belong to the pair-wised method using similarity matrix (containing the pair-wised similarity of data) to describe the relationship between the image pairs or data pairs and this similarity information is used to learn hash functions. However, it is not so easy to construct the matrix and generate the codes while dealing with a large-scale dataset.

Based on the advancement of deep learning, we doubt whether we can take the advantage of deep CNN to achieve hashing. Can we generate the binary compact codes directly from the deep CNN rather than use the pair-wised learning method? To answer these questions, we propose a deep CNN model that can simultaneously learn image representations and binary codes. The premise is that the data are labeled, which means that our method is designed particularly for supervised learning. Furthermore, we argue that when a powerful learning model such as deep CNN is used and the data labels are available, the binary codes can be learned by employing some hidden layer for representing the latent concepts (with binary activation functions such as sigmoid) that dominate the class labels in the architecture, which is different from other supervised methods (such as [30]) that take the data labels into consideration but require pair-wised inputs to the prepared learning process. That is, our method

978-1-5090-0022-7/15/$31.00 ©2015 IEEE

learns binary hashing codes in a point-wised manner by taking advantage of the incremental learning nature (via stochastic gradient descent) of deep CNN. The employment of deep architecture also brings efficient-retrieval feature learning. Comparing with conventional methods, our method is suitable for large datasets.

The characteristics of the method are as follows:

- A simple yet effective supervised learning framework is introduced for rapid image retrieval.
- Our deep CNN simultaneously learns domain specific image representations and a set of hashing-like functions for rapid image retrieval with small modifications to the network model.
- The proposed method is better than all of the most advanced works on the public dataset MNIST and CIFAR-10. Our model improves the previous best retrieval performance on CIFAR10 dataset by 30%
- precision and on MNIST dataset by 1% precision.
- Comparing with conventional pair-wised methods, our method learns binary hashing codes in a point- wised manner and is easily scalable to the data size.

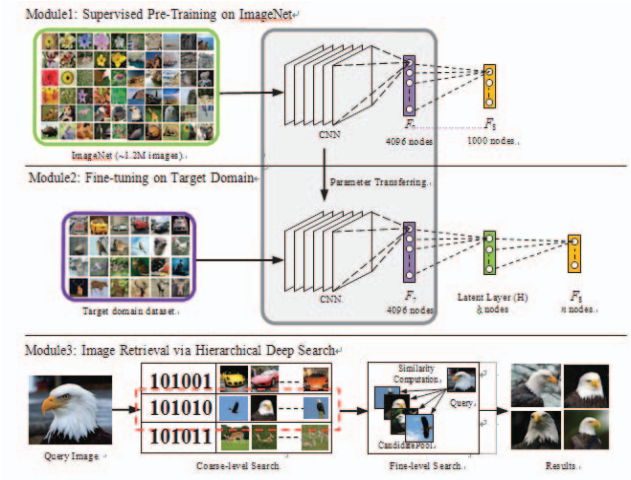

Figure 1: The image retrieval framework via classified deep search

Organization of this paper is as follows: We elaborate on our method in Section 2 Finally, experimental results are provided in Section 3, followed by conclusions in Section 4.

II. Related Work

Several hashing algorithms [6, 29, 18, 20, 28, 10] have been proposed to approximately identify data relevant to the query. These methods can be classified into two main categories - unsupervised and supervised methods.

Unsupervised hashing methods learn a set of hash functions through unlabeled data [6, 29, 8]. The most representative one is the Locality-Sensitive Hashing (LSH) [6] aiming at maximizing the probability that similar data are mapped to similar binary codes. LSH generates the bi- nary codes by projecting the data points to a random hyper- plane with random threshold. Spectral hashing (SH) [29] is another representative method, which produces the com- pact binary codes via threshold with non-linear functions along the PCA direction of the given data.

Recent studies have shown that the binary hash codes learning performance will be facilitated by using supervised information. Supervised methods [18, 20, 15] incorporate label information during learning. These supervised hashing methods usually use the pair-wised labels for generating effective hash functions. But generally these algorithms require a large sparse matrix to describe the similarity between data points in the training set.

Image representations are also of essential importance in CBIR except the research track of hashing. Recently, image retrieval has applied CNN-based visual descriptors in its task. Krizhevsky *et al*. [14] firstly retrieve images by using the features extracted from seventh layer and achieve impressive performance on ImageNet. Babenko *et al*. [1] focus on dimensional reduction of the CNN features and improve the retrieval performance with compressed CNN features. Though these recent works [14, 1] present good results on the task of image retrieval, the learned CNN features are employed for retrieval by directly performing pattern matching in the Euclidean space, which is inefficient.

Deep architectures have been used for hash learning. However, most of them are unsupervised and deep auto-encoders are used for learning the representations [24, 13]. Xia *et al*. [30] propose a supervised hashing method to learn binary hashing codes for fast image retrieval through deep learning and demonstrate most advanced retrieval performance on public datasets. However, in their pre-processing stage, a matrix-decomposition algorithm is used for learning the representation codes for data. Thus, the input of a pair-wised similarity matrix of the data is required and is not suitable for the case when the data size is large (*e.g*., 1M in our experiment) because it consumes both considerable storage and computational time.

In the contrary, we present a simple yet efficient deep learning method to learn a set of effective hash-like functions and it performs well on the publicly available datasets. We further apply our method to a large-scale dataset of 1 million clothing images to demonstrate the scalability of our method. We will describe the proposed method in the next section.

III. METHOD

Figure 1 shows the proposed framework. Our method includes three main components, among which the first component is the supervised pre-training on the large-scale ImageNet dataset [14]. The second component is adjusting the network with the latent layer for simultaneous learning of domain-specific feature representation and a set of hash-like functions. The third component retrieves images similar to the query one via the proposed classified deep search. We use the pre- trained CNN model proposed by Krizhevsky *et al*. [14] from the Caffe CNN library [11], which is trained on the large-scale ImageNet dataset containing more than 1.2 million images with 1000 object classes. Our method for learning binary codes is described in detail as follows.

A. Learning Hash-like Binary Codes

Recent studies [14, 7, 5, 1] said that the feature activations of layers F_{6-8} induced by the input image can be regarded as the visual signatures. The use of these mid-level image representations shows great improvement on the task of image classification, retrieval and others. However, these signatures are high-dimensional vectors that are inefficient for image retrieval in a large corpus. In order to promote efficient image retrieval, a practical way to reduce the computational cost is to convert the feature vectors into binary codes. The binary compact codes can be quickly compared using hashing or Hamming distance.

In this work, we propose to learn the domain specific image representations and a set of hash-like (or binary coded) functions at the same time. The assumption is that the final outputs of the classification layer F_8 rely on a set of h hidden attributes with each attribute *on* or *off*. In other points of view, images inducing similar binary activations would have the same label. We embed the latent layer H between F_7 and F_8 as shown in the middle row of Figure 1 to implement this idea. The latent layer H is a fully connected layer, and its neuron activities are regulated by the subsequent layer F_8 which is encoding semantics and achieves classification. The said latent layer H not only provides an abstraction of the rich features from F_7, but also links the mid-level features and the high-level semantics. In our design, the neurons in the latent layer H are activated by sigmoid functions so the activations almost equal to {0,1}.

We adjust the proposed network on the target-domain dataset via reverse transferring to realize domain adaptation. The initial weights of the deep CNN are set to be the same as the weights trained from ImageNet dataset. The weights of the latent layer H and the final classification layer F_8 are set at random. The initial random weights of latent layer H are regarded as LSH [6] by using random predictions to set up the hashing bits. The codes are then adapted from LSH to those suitable for the data better from supervised deep-network learning. With dramatic modifications to a deep CNN model, the target model learns domain specific visual descriptors and a set of hashing-like functions at the same time for efficient image searching.

B. Image Search with Hierachical Deep Search

Zeiler and Fergus [32] analyzed the deep CNN and showed that the shallow layers learn local visual descriptors while the deeper layers of CNN catch the semantic information suitable for identification. A coarse-to-fine search way is used for rapid and accurate image search. We firstly search a set of candidates with similar high-level semantics, i.e. with similar concealed binary activations from the latent layer. In order to further screen the images with similar appearance, similarity ranking is used based on the deepest mid-level image representations

Coarse-level Search. Given an image I, we first get the outputs of the latent layer as the image signature which is presented by Out(H). The binary codes are then obtained by binarizing the activations with a threshold. For each bit $k = 1 \ldots h$ (where h is the number of nodes in the latent layer), we output the binary codes of H by

$$H^{\mathrm{k}} = \begin{cases} 1 & LatOut^{\mathrm{k}}(H) \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

With Hi $\in$ {0, 1}, Given a query image Iq, and its binary codes Hq , we identify a pool of m candidates, $P=\{I_1^k, I_2^k, \ldots I_m^k\}$, if the Hamming distance between Hq and Hi is lower than a threshold.Let Γ = {I1, I2, . . . , In} mean the dataset containing n images for searching. The corresponding binary codes of each image are regarded as ΓH = {H1, H2, . . . , Hn}.

Smaller the Euclidean distance equals to higher level of similarity of the two images. Each candidate Ic is ranked in up order by the similarity; Finally, we identify top k ranked images.

Fine-level Search. Given the query image I_q and the candidate pool P, we use the features gotten from the layer F_7 to identify the top k ranked images, which will form the candidate pool P. Let V_q and V^P mean the feature vectors of the query image q and of the image I^c from the pool respectively. We call the similarity level between I_q and the i-th image of P as the Euclidean distance between their corresponding features vectors,

$$s_i = \left\| V_q - V_i^P \right\|. \quad (2)$$

IV. EXPERIMENTAL RESULTS

A. Datasets

MNIST Dataset [16] is composed of 10 categories of the handwritten digits form 0 to 9. There are 60,000 training

images and 10,000 test images. All the digits are normal-ized to gray-scale images with size 28×28.

CIFAR-10 Dataset [12] contains 10 object categories and each category consists of 6,000 images. Therefore, there are 60,000 images in total. The dataset is split into training and test sets, with 50,000 and 10,000 images respectively.

Yahoo-1M Dataset contains 1,124,087 shopping product images in total and 116 clothing-specific categories. The dataset is collected by catching the images from the Yahoo shopping sites. All the images are labeled with a category such as Top, Dress and Skirt etc. Figure 2 shows some examples of the dataset.

In the experiments of MNIST and CIFAR-10, we search the relevant images through the learned binary codes to compare with others. In the experiments of Yahoo-1M dataset, we search similar images from the entire dataset via the classified search.

B. *Evaluation Metrics*

A ranking-based standard [4] is used for evaluation. Given a query image q and a similarity measure, a rank can be distributed for each dataset image. We evaluate the ranking of top k images related to a query image q .

Table 1:Performance Comparison (Error,%)of Classification Error Rates on the MNIST dataset.

Methods	Test Error (%)
2-Layer CNN+2-Layer NN [31]	0.53
Stochastic Pooling [31]	0.47
NIN+Dropout [17]	0.47
Conv. maxout+Dropout [9]	0.45
Ours w/ 48 nodes latent layer	0.50

Precision:

$$Precision@k = \frac{\sum_{i=1}^{k} Rel(i)}{k}, \qquad (3)$$

Rel(i) means the ground truth relevance between a query q and the i-th ranked image. Here, we only consider the category label in measurement of the relevance so Rel(i) $\in \{0, 1\}$ with 1 for the query and the ith image with the same label and 0 otherwise.

Figure 2: Sample images from the Yahoo-1M Shopping Dataset. The heterogeneous product images demonstrate highly variation, and are challenging to image classification and retrieval

C. *Results on MNIST Dataset*

Performance of Image Classification. We modify the layer F_8 to 10-way to predict 10 digit classes in order to adapt our deep CNN on the new domain. We set the number of neurons h in the latent layer to 48 respectively to measure the effect of latent layer embedded in the deep CNN before we apply stochastic gradient descent (SGD) to train the CNN on the MNIST dataset. The network is trained for 10,000 repetitions with a learning rate of 0.001.

We compare our results with several most advanced methods [31, 17, 9] in Table 1. Our method with 48 latent nodes at trains 0.50% error rate and is better than most of the other methods. It is worth mentioning that our model is designed particularly for image search while others are optimized for a classification task through modification of a network. For example, the work of [31] proposed activation function improving the accuracy of dropout's approximate model averaging technique. Another famous work is Network in Network (NIN) [17], strengthening the identification of local patches via multilayer perception and avoids over fitting by using the international average pooling instead of the fully connected layers.

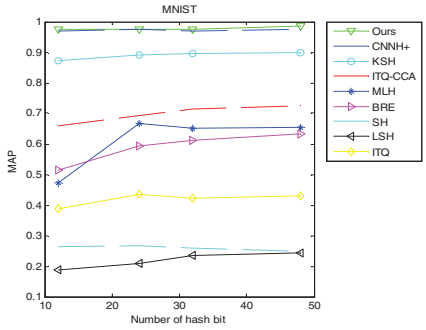

Figure 3: Performance of Image Classification on MNIST

Performance of Images Retrieval. In this experiment, we unify the retrieval evaluation that retrieves the relevant images by using 48 bits binary code and hamming distance

measure. The retrieval is performed by randomly selecting 1,000 query images from the testing set for the system to retrieve relevant ones from the training set.In order to assess the retrieval performance, we compare the said method with several most advanced hashing methods, including supervised (KSH [18], MLH [20], BRE [15], CNNH [30], and CNNH+ [30]) and unsupervised methods (LSH [6], SH [29], and ITQ [8]). The retrieval precision of different methods related to different number of searched images is shown in Figure 4. It is clear that our method works stably (98.5% retrieval precision) despite the number of images retrieved. In addition, our method improves the precision from 97.5% achieved by CNNH+ [30] to 98.5%, which learns the hashing functions via dissolution of the similarity information in pairs. This improvement indicates that our point-wised method only asking for class

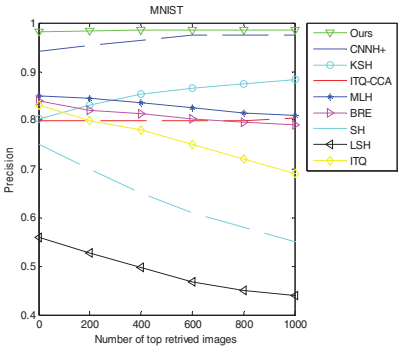

Figure 4: Performance of Image Retrieval on MINIST

labels is effective.

We further analyze the quality of the learned hash-like codes for $h = 48$ as shown in Figure 3. It is clear that both settings can learn informative binary codes for image retrieval.Figure 4 illustrates our searching results outperform the previous methods.

D. Results on CIFAR 10 Dataset

Performance of Image Classification. We modify F_8 to 10-way to predict 10 object categories to transfer the deep CNN to the domain of CIFAR-10, and h is also set as 48. We then adjust our network model on the CIFAR-10 dataset and finally achieve approximate 89.5% testing accuracy after 50, 000 repeated trainings. As we can see in Table 2, the said method is better than most methods [31, 26, 3, 14, 17], which indicates that buried the binary latent layer in the deep CNN does not affect the performance greatly.

Table 2: Performance Comparison (mAP, %) of Classification Accuracy on the CIFAR-10 dataset.

Methods	Accuracy (%)
Stochastic Pooling [31]	84.87
CNN + Spearmint [26]	85.02
MCDNN [3]	88.79
AlexNet + Fine-tuning [14]	89
NIN + Dropout [17]	89.59
NIN + Dropout + Augmentation [17]	91.2
Ours w/ 48 nodes latent layer	89.5

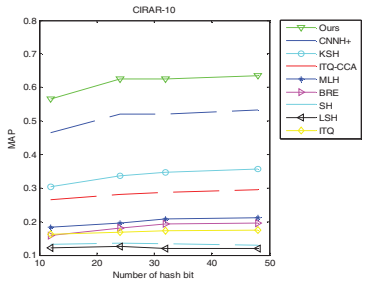

Figure 5: Performance of Image Classification on CIRAR-10

Performance of Image Retrieval. In order to compare with other hashing algorithms, we unify the evaluation method that searches the related images by 48 bits binary codes and Hamming distance. Figure 6 shows the precision curves related to different number of the top retrieved samples. Performance of our method is better than that of other unsupervised and supervised methods. What's more, it attains a precision of 89% while varying the number of retrieved images and improving the performance by more than 30% compared to CNNH+ [30]. These results suggest that it is a practical method to use a latent layer for representing the hidden concepts for learning of efficient binary codes.

Figure 6 shows our searching results. The proposed latent binary codes search images with relevant category, similar appearance, and/or both. We retrieves more appearance-relevant images by increasing the bit numbers from $h = 48$ according to our sight checking based on experience.

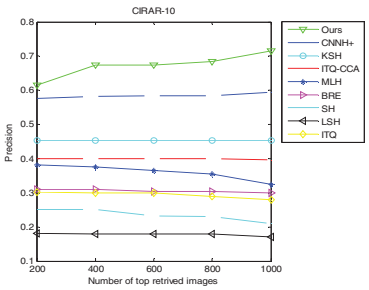

Figure 6: Performance of Image Retrieval on CIRAR-10

E. Results on Yahoo 1M Dataset.

Performance of Image Classification., We further test it on the large-scale Yahoo-1M dataset to show the scalability and efficacy of our method. This dataset is composed of efficient product images that are uneven and they are with different person poses with noisy backgrounds.

We set the number of neurons in the classification layer to 116, and h in the latent layer to 48. We then adjust our network with the whole Yahoo-1M dataset. After 750, 000

repeated trainings; our method provides 83.75% accuracy (obtained by the final layer) on the task of 116 categories clothing classification.

Performance of Images Retrieval. In this experiment, we prove that our method can learn efficient deep binary codes for the dataset of million data. This is not easy to achieve by using previous pair-wised-data methods because of the large time and storage complexity.

Figure 7 shows our searching results

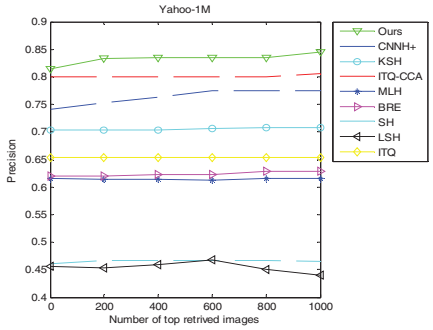

Figure 7: Performance of Image Retrieval on Yahoo-1M

V. Conclusions

We present a simple yet effective deep learning framework and create the hash-like binary codes for fast image retrieval. We add a latent feature layer in the deep CNN for learning of domain specific image representations and a set of hash-like functions. Our method does not rely on pair-wised similarities of data and is highly scalable to the dataset size. It is shown through experimental results that our method improves the previous best retrieval results with 1% and 30% retrieval precision on the MNIST and CIFAR-10 datasets respectively with only a simple modification of the deep CNN. We further prove the scalability and efficacy of the said method on the large-scale dataset of 1 million shopping images.

References

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Proc. ECCV*, pages 584–599. Springer, 2014. 1, 2, 3

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. ECCV*, pages 404–417. Springer, 2006. 1

[3] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proc. CVPR*, pages 3642–3649. IEEE, 2012. 6

[4] J. Deng, A. C. Berg, and F.-F. Li. Hierarchical semantic indexing for large scale image retrieval. In *Proc. CVPR*, 2011. 4

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proc. ICML*, 2014. 3

[6] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999. 1, 2, 3, 6

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 1, 3

[8] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean method to learning binary codes. In *Proc. CVPR*, pages 817–824, 2011. 2, 6

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013. 5

[10] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *Proc. CVPR*, pages 1–8, 2008. 2

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2

[12] A. Krizhevsky. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Report*, 2009. 4

[13] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011. 2

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 1, 2, 3, 6, 7

[15] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Proc. NIPS*, pages 1042–1050, 2009. 1, 2, 6

[16] Y. LeCun and C. Cortes. The mnist database of handwritten digits, 1998. 4

[17] M. Lin, Q. Chen, and S. Yan. Network in network. In *Proc. ICLR*, 2014. 5, 6

[18] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Proc. CVPR*, pages 2074–2081, 2012. 1, 2, 6

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[20] M. Norouzi and D. M. Blei. Minimal loss hashing for compact binary codes. In *Proc. ICML*, pages 353–360, 2011. 1, 2, 6

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. CVPR*, 2014. 1

[22] G. Qiu. Indexing chromatic and achromatic patterns for content-based colour image retrieval. *PR*, 35(8):1675–1686, 2002. 1

[23] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proc. CVPRW*, pages 512–519. IEEE, 2014. 1

[24] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 500(3):500, 2007. 2

[25] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI*, 22(12):1349–1380, 2000. 1

[26] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Proc. NIPS*, pages 2951–2959, 2012. 6

[27] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proc. ACM MM*, pages 157–166, 2014. 1

[28] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proc. CVPR*, pages 3424–3431, 2010. 2

[29] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. NIPS*, pages 1753–1760, 2009. 1, 2, 6

[30] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *Proc. AAAI*, 2014. 1, 2, 6, 7

[31] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013. 5, 6

[32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, pages 818–833. Springer, 2014. 4