Deep learning of binary hash codes for fast image retrieval

Conference Paper · June 2015

DOI: 10.1109/CVPRW.2015.7301269

CITATIONS
115

READS
264

4 authors, including:

Kevin Lin
University of Washington Seattle
14 PUBLICATIONS **231** CITATIONS
SEE PROFILE

Huei-Fang Yang
Howard Hughes Medical Institute
26 PUBLICATIONS **275** CITATIONS
SEE PROFILE

Chu-Song Chen
Academia Sinica
123 PUBLICATIONS **2,269** CITATIONS
SEE PROFILE

Deep Learning of Binary Hash Codes for Fast Image Retrieval

Kevin Lin[†], Huei-Fang Yang[†], Jen-Hao Hsiao[‡], Chu-Song Chen[†]
[†]Academia Sinica, Taiwan [‡]Yahoo! Taiwan
{kevinlin311.tw,song}@iis.sinica.edu.tw, hfyang@citi.sinica.edu.tw, jenhaoh@yahoo-inc.com

Abstract

Approximate nearest neighbor search is an efficient strategy for large-scale image retrieval. Encouraged by the recent advances in convolutional neural networks (CNNs), we propose an effective deep learning framework to generate binary hash codes for fast image retrieval. Our idea is that when the data labels are available, binary codes can be learned by employing a hidden layer for representing the latent concepts that dominate the class labels. The utilization of the CNN also allows for learning image representations. Unlike other supervised methods that require pair-wised inputs for binary code learning, our method learns hash codes and image representations in a point-wised manner, making it suitable for large-scale datasets. Experimental results show that our method outperforms several state-of-the-art hashing algorithms on the CIFAR-10 and MNIST datasets. We further demonstrate its scalability and efficacy on a large-scale dataset of 1 million clothing images.

1. Introduction

Content-based image retrieval aims at searching for similar images through the analysis of image content; hence image representations and similarity measure become critical to such a task. Along this research track, one of the most challenging issues is associating the pixel-level information to the semantics from human perception [25, 27]. Despite several hand-crafted features have been proposed to represent the images [19, 2, 22], the performance of these visual descriptors is still limited until the recent breakthrough of deep learning. Recent studies [14, 7, 21, 23] have shown that deep CNN significantly improves the performance on various vision tasks, such as object detection, image classification, and segmentation. These accomplishments are attributed to the ability of deep CNN to learn the rich mid-level image representations.

As deep CNNs learn rich mid-level image descriptors, Krizhevsky *et al*. [14] used the feature vectors from the 7th layer in image retrieval and demonstrated outstanding performance on ImageNet. However, because the CNN features are high-dimensional and directly computing the similarity between two 4096-dimensional vectors is inefficient, Babenko *et al*. [1] proposed to compress the CNN features using PCA and discriminative dimensionality reduction, and obtained a good performance.

In CBIR, both image representations and computational cost play an essential role. Due to the recent growth of visual contents, rapid search in a large database becomes an emerging need. Many studies aim at answering the question that how to efficiently retrieve the relevant data from the large-scale database. Due to the high-computational cost, traditional linear search (or exhaustive search) is not appropriate for searching in a large corpus. Instead of linear search, a practical strategy is to use the technique of Approximate Nearest Neighbor (ANN) or hashing based method [6, 29, 18, 20, 15, 30] for speedup. These methods project the high-dimensional features to a lower dimensional space, and then generate the compact binary codes. Benefiting from the produced binary codes, fast image search can be carried out via binary pattern matching or Hamming distance measurement, which dramatically reduces the computational cost and further optimizes the efficiency of the search. Some of these methods belong to the pair-wised method that use similarity matrix (containing the pair-wised similarity of data) to describe the relationship of the image pairs or data pairs, and employ this similarity information to learn hash functions. However, it is demanding to construct the matrix and generate the codes when dealing with a large-scale dataset.

Inspiring from the advancement of deep learning, we raise a question that can we take the advantage of deep CNN to achieve hashing? Instead of the use of the pair-wised learning method, can we generate the binary compact codes directly from the deep CNN? To address these questions, we propose a deep CNN model that can simultaneously learn image representations and binary codes, under the assumption that the data are labeled. That is, 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. This is different from other supervised methods (such as [30]) that take into consideration the data labels but require pair-wised inputs to the prepared learning process. In other words, our approach learns binary hashing codes in a point-wised manner, taking advantage of the incremental learning nature (via stochastic gradient descent) of deep CNN. The employment of deep architecture also allows for efficient-retrieval feature learning. Our method is suitable for large datasets in comparison of conventional approaches.

Our method is with the following characteristics:

- We introduce a simple yet effective supervised learning framework for rapid image retrieval.
- With small modifications to the network model, our deep CNN simultaneously learns domain specific image representations and a set of hashing-like functions for rapid image retrieval.
- The proposed method outperforms all of the state-of-the-art 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.
- Our approach learns binary hashing codes in a point-wised manner and is easily scalable to the data size in comparison of conventional pair-wised approaches.

This paper is organized as follows: We briefly review the related work of hashing algorithms and image retrieval with deep learning in Section 2. We elaborate on the details of our method in Section 3. Finally, experimental results are provided in Section 4, followed by conclusions in Section 5.

2. Related Work

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

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

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

Beside the research track of hashing, image representations also play an essential role in CBIR. CNN-based visual descriptors have been applied on the task of image retrieval recently. Krizhevsky *et al*. [14] firstly use the features extracted from seventh layer to retrieve images, 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, where deep auto-encoders are used for learning the representations [24, 13]. Xia *et al*. [30] propose a supervised hashing approach to learn binary hashing codes for fast image retrieval through deep learning and demonstrate state-of-the-art retrieval performance on public datasets. However, in their pre-processing stage, a matrix-decomposition algorithm is used for learning the representation codes for data. It thus requires the input of a pair-wised similarity matrix of the data and is unfavorable 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 contrast, we present a simple but efficient deep learning approach to learn a set of effective hash-like functions, and it achieves more favorable results 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 approach. We will describe the proposed method in next section.

3. Method

Figure 1 shows the proposed framework. Our method includes three main components. The first component is the supervised pre-training on the large-scale ImageNet dataset [14]. The second component is fine-tuning the network with the latent layer to simultaneously learn domain-specific feature representation and a set of hash-like function. The third retrieves images similar to the query one via the proposed hierarchical 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 which contains more than 1.2

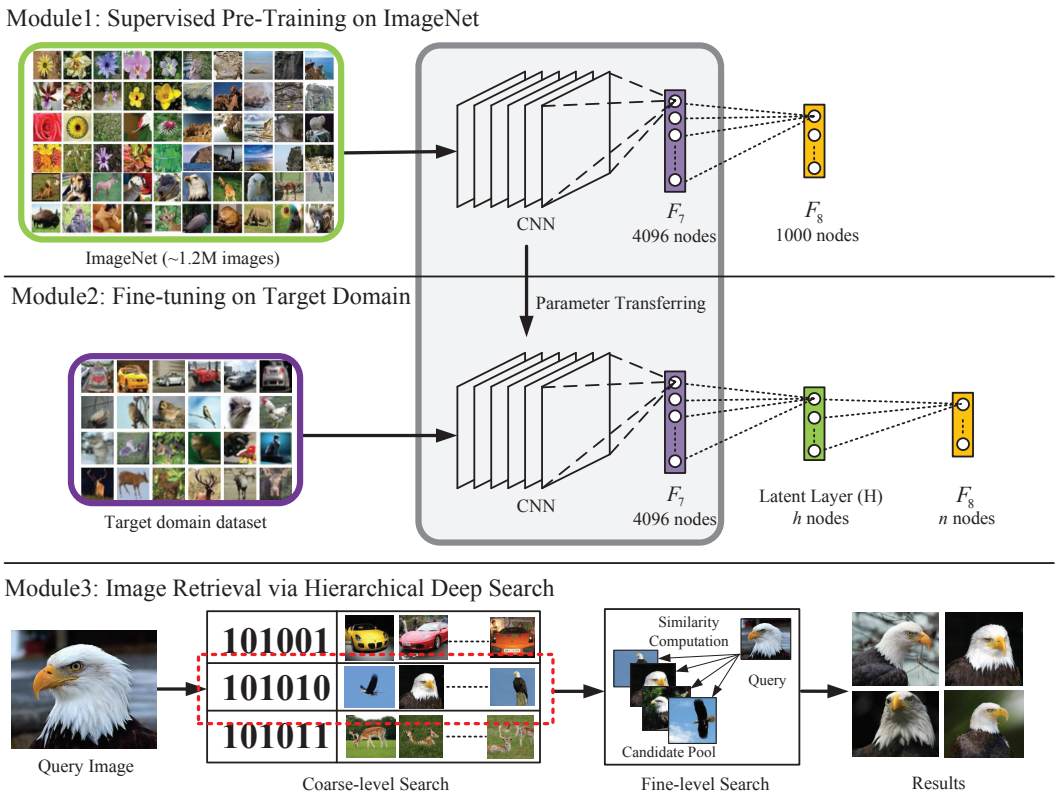

Figure 1: The proposed image retrieval framework via hierarchical deep search. Our method consists of three main components. The first is the supervised pre-training of a convolutional neural network on the ImageNet to learn rich mid-level image representations. In the second component, we add a latent layer to the network and have neurons in this layer learn hashes-like representations while fine-tuning it on the target domain dataset. The final stage is to retrieve similar images using a coarse-to-fine strategy that utilizes the learn hashes-like binary codes and F_7 features.

million images categorized into 1000 object classes. Our method for learning binary codes is described in detail as follows.

3.1. Learning Hash-like Binary Codes

Recent studies [14, 7, 5, 1] have shown that the feature activations of layers F_{6-8} induced by the input image can serve as the visual signatures. The use of these mid-level image representations demonstrates impressive 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. To facilitate efficient image retrieval, a practical way to reduce the computational cost is to convert the feature vectors to binary codes. Such 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 simultaneously. We assume 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. To fulfill this idea, we embed the latent layer H between F_7 and F_8 as shown in the middle row of Figure 1. The latent layer H is a fully connected layer, and its neuron activities are regulated by the succeeding layer F_8 that encodes semantics and achieves classification. The proposed latent layer H not only provides an abstraction of the rich features from F_7, but also bridges 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 are approximated to $\{0, 1\}$.

To achieve domain adaptation, we fine-tune the proposed network on the target-domain dataset via back propagation. The initial weights of the deep CNN are set as the weights trained from ImageNet dataset. The weights of the latent layer H and the final classification layer F_8 are randomly initialized. The initial random weights of latent layer H acts like LSH [6] which uses random projections for constructing the hashing bits. The codes are then adapted from LSH to those that suit the data better from supervised deep-network learning. Without dramatic modifications to a deep

CNN model, the propose model learns domain specific visual descriptors and a set of hashing-like functions simultaneously for efficient image retrieval.

3.2. Image Retrieval via Hierarchical 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 capture the semantic information suitable for recognition. We adopt a coarse-to-fine search strategy for rapid and accurate image retrieval. We firstly retrieve a set of candidates with similar high-level semantics, that is, with similar hidden binary activations from the latent layer. Then, to further filter the images with similar appearance, similarity ranking is performed based on the deepest mid-level image representations.

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

$$H^j = \begin{cases} 1 & Out^j(H) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let $\mathbf{\Gamma} = \{I_1, I_2, \ldots, I_n\}$ denote the dataset consisting of n images for retrieval. The corresponding binary codes of each images are denoted as $\mathbf{\Gamma}_H = \{H_1, H_2, \ldots, H_n\}$ with $H_i \in \{0,1\}^h$. Given a query image I_q and its binary codes H_q, we identify a pool of m candidates, $P = \{I_1^c, I_2^c, \ldots, I_m^c\}$, if the Hamming distance between H_q and $H_i \in \mathbf{\Gamma}_H$ is lower than a threshold.

Fine-level Search. Given the query image I_q and the candidate pool P, we use the features extracted from the layer F_7 to identify the top k ranked images to form the candidate pool P. Let V_q and V_i^P denote the feature vectors of the query image q and of the image I_i^c from the pool, respectively. We define the similarity level between I_q and the i-th image of P as the Euclidean distance between their corresponding features vectors,

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

The smaller the Euclidean distance is, the higher level the similarity of the two images is. Each candidate I_i^c is ranked in ascending order by the similarity; hence, top k ranked images are identified.

4. Experimental Results

In this section, we demonstrate the benefits of our approach. We start with introducing the datasets and then present our experimental results with performance comparison to several state-of-the-arts on the public datasets, MNIST and CIFAR-10 datasets. Finally, we verify the scalability and the efficacy of our approach on the large-scale Yahoo-1M dataset.

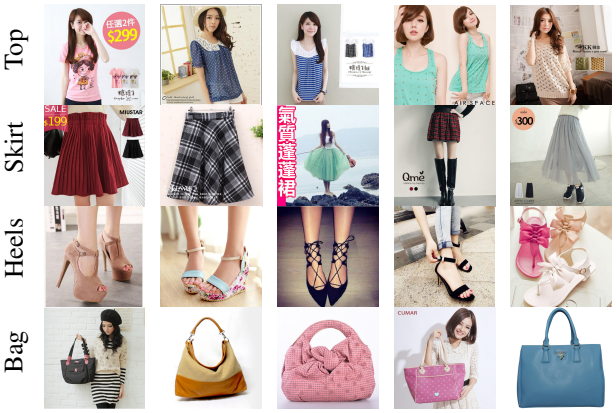

Figure 2: Sample images from the Yahoo-1M Shopping Dataset. The heterogeneous product images demonstrate highly variation, and are challenging to image classification and retrieval.

4.1. Datasets

MNIST Dataset [16] consists 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 normalized to gray-scale images with size 28×28.

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

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

In the experiments of MNIST and CIFAR-10, we retrieve the relevant images using the learned binary codes in order to fairly compare with other hashing algorithms. In the experiments of Yahoo-1M dataset, we retrieve similar images from the entire dataset via the hierarchical search.

4.2. Evaluation Metrics

We use a ranking based criterion [4] for evaluation. Given a query image q and a similarity measure, a rank can be assigned for each dataset image. We evaluate the ranking of top k images with respect to a query image q by a

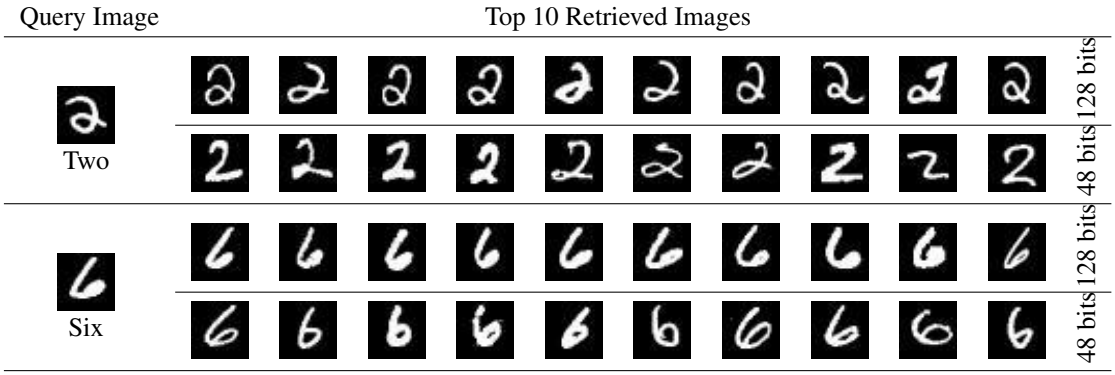

Figure 3: Top 10 retrieved images from MNIST dataset by vary bit numbers of the latent binary codes. Relevant images with similar appearance are retrieved when the bit numbers increased.

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	0.47
Ours w/ 128 nodes latent layer	0.50

precision:

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

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

4.3. Results on MNIST Dataset

Performance of Image Classification. To adapt our deep CNN on the new domain, we modify the layer F_8 to 10-way softmax to predict 10 digit classes. In order to measure the effect of latent layer embedded in the deep CNN, we set the number of neurons h in the latent layer to 48 and 128, respectively. Then, we apply stochastic gradient descent (SGD) to train the CNN on the MNIST dataset. The network is trained for 50,000 iterations with a learning rate of 0.001.

We compare our results with several state-of-the-arts [31, 17, 9] in Table 1. Our approach with 48 latent nodes attains 0.47% error rate and performs favorably against most of the alternatives. It it worth noting that our model is designed particularly for image retrieval whereas others are optimizing for a classification task through modification of a network. For example, the work of [31] proposed the maxout activation function which improves the accuracy of dropout's approximate model averaging technique. Another representative work is Network in Network (NIN) [17], which enhances the discriminability of local patches via multilayer perception, and avoids overfitting using the global average pooling instead of the fully connected layers. Also note that our method with 48 latent nodes yields an error rate lower than the model with 128 nodes does. This may be due to that few latent nodes are capable of representing latent concepts for classification and adding more neurons can cause overfitting.

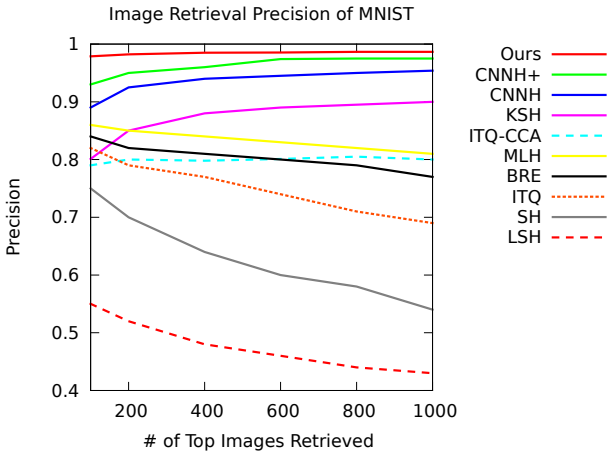

Figure 4: Image retrieval precision with 48 bits of MNIST dataset.

Figure 5: Top 10 retrieved images from CIFAR-10 by vary bit numbers of the latent binary codes. Relevant images with similar appearance are retrieved when the bit numbers increased.

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	89.4
Ours w/ 128 nodes latent layer	89.6

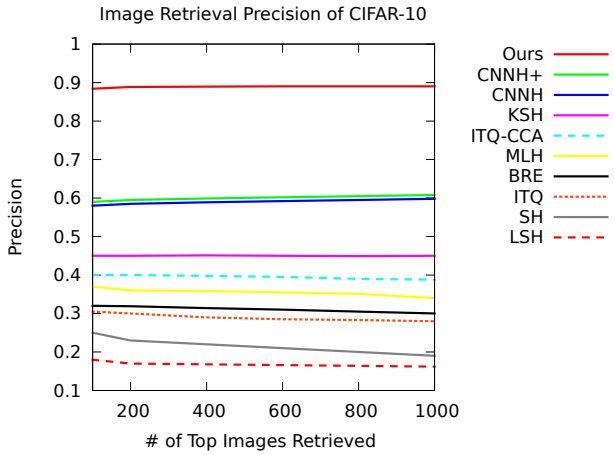

Figure 6: Image retrieval precision with 48 bits of CIFAR-10 dataset.

Performance of Images Retrieval. In this experiment, we unify the retrieval evaluation that retrieve the relevant images 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.

To evaluate the retrieval performance, we compare the proposed method with several state-of-the-art hashing approaches, including supervised (KSH [18], MLH [20], BRE [15], CNNH [30], and CNNH+ [30]) and unsupervised methods (LSH [6], SH [29], and ITQ [8]). Figure 4 shows the retrieval precision of different methods with respect to different number of retrieved images. As can be seen, our method demonstrates stable performance ($98.2^{+}_{-}0.3\%$ retrieval precision) regardless of the number of images retrieved. Furthermore, our approach improves the precision to 98.5% from 97.5% achieved by CNNH+ [30], which learns the hashing functions via decomposition of the pair-wised similarity information. This improvement indicates that our point-wised method that requires only class labels is effective.

We further analyze the quality of the learned hash-like codes for $h = 48$ and $h = 128$, respectively, as shown in Figure 3. As can be seen, both settings can learn informative binary codes for image retrieval.

4.4. Results on CIFAR-10 Dataset

Performance of Image Classification. To transfer the deep CNN to the domain of CIFAR-10, we modify F_8 to 10-way softmax to predict 10 object categories, and h is also set as 48 and 128. We then fine-tune our network model on the CIFAR-10 dataset, and finally achieves around 89% testing accuracy after $50,000$ training iterations. As shown in Table 2, the proposed method is more favorable against most approaches [31, 26, 3, 14, 17], which indicates that embedding the binary latent layer in the deep CNN does not severely alter the performance.

Figure 7: Image classification results on Yahoo-1M dataset. The first row indicates ground truth label. The bars below depict the prediction scores sorted in ascending order. Red and blue bar represent the correct and incorrect predictions, respectively.

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

Figure 5 shows our retrieval results. The proposed latent binary codes successfully retrieve images with relevant category, similar appearance, and/or both. Increasing the bit numbers from $h = 48$ to $h = 128$ retrieves more appearance-relevant images according to our empirical eyeball checking. For example, in Figure 5, using $h = 128$ bits binary code tends to retrieve more relevant horse-head images (instead of entire horses) than that of the $h = 48$ bits.

4.5. Results on Yahoo-1M Dataset.

Performance of Image Classification. To show the scalability and efficacy of our method, we further test it on the large-scale Yahoo-1M dataset. This dataset consists of plentiful product images that are heterogeneous and they are variant in person poses with noisy backgrounds.

We set the number of neurons in the classification layer to 116, and h in the latent layer to 128. We then finetune our network with the entire Yahoo-1M dataset. After $750,000$ training iterations, our proposed approach achieves 83.75% accuracy (obtained by the final layer) on the task of 116 categories clothing classification. As shown in Fig 7, though the clothing images are backgroundless or of noisy backgrounds, with or without human, the proposed method demonstrates a good classification performance. Note that some of the images are miss-predicted

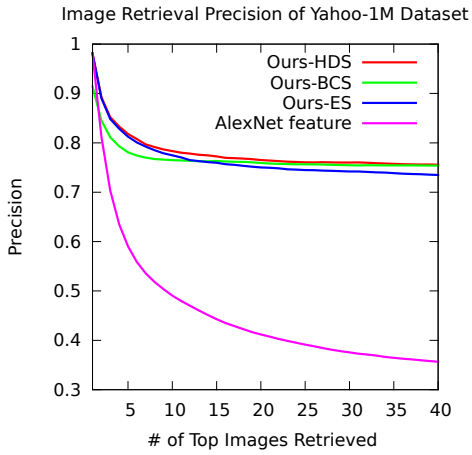

Figure 8: Image retrieval precision of Yahoo-1M dataset.

because the products might be ambiguous between some specific categories. For example, it might be difficult to distinguish between the Mary Janes and the Flats as shown in Fig 7. However, our method can still retrieve the images similar to the query image.

Performance of Images Retrieval. In this experiment, we demonstrate that our method can learn efficient deep binary codes for the dataset of million data. This is demanding to achieve by using previous pairwised-data approaches due to the large time and storage complexity.

Because image representations are critical to image retrieval, we compare the retrieval results obtained by features from different network modes: (1) **AlexNet:** F_7 feature from the pre-trained CNN [14]; (2) **Ours-ES:** F_7 features from our network; (3) **Ours-BCS:** Latent binary codes from our network; and (4) **Ours-HDS:** F_7 features and latent binary codes from our network.

We conduct the exhaustive search (or linear search) based on L_2-norm distance when the F_7 features are used in

Figure 9: Top 5 retrieved images from Yahoo-1M dataset by different features. The blue check marks indicate the query and retrieved images share the same label; the black crosses indicate otherwise.

retrieval; the hashing is performed based on Hamming distance when the binary codes of the latent layer are used; the coarse-to-fine hierarchical search is performed to retrieve relevant images by using both the laytent layer codes and F_7. We randomly select 1000 images from the Yahoo-1M dataset, and retrieve the relevant images from the same dataset.

Figure 8 shows the precision regarding to various number of the top images retrieved using different CNN features. The proposed methods perform more favorably against the original AlexNet feature. Apparently, the procedure of fine-tuning successfully transfers deep CNN to the new domain (clothing images). Among the fine-tuned models, Our-ES and Our-HDS show good retrieval precision at first. However, Ours-BCS outperforms Ours-ES with higher and more stable retrieval precision when more than 12 images are retrieved. This indicates the learned binary codes are informative and with high discriminative power. Ours-HDS complements both Ours-BCS and Ours-ES and achieves the best retrieval precision in overall.

Figure 9 shows the top 5 images retrieved by different features. As can be seen, AlexNet retrieves the images with great diversity. The fine-tuned models retrieve more images with the same label as the query than AlexNet. Ours-HDS, Ours-BCS, and Ours-ES demonstrate good performance, and successfully retrieve similar products. Nevertheless, benefiting from the binary codes, Ours-BCS achieves the fastest search among the approaches compared. Extracting CNN features takes around 60 milliseconds (ms) on the machine with Geforce GTX 780 GPU and 3 GB memory. The search is carried out on the CPU mode with C/C++ implementation. Performing an Euclidean distance measure between two 4096-dimensional vectors takes 109.767 ms. In contrast, computing the hamming distance between two 128 bits binary codes takes 0.113 ms. Thus, Ours-BCS is 971.3x faster than traditional exhaustive search with 4096-dimensional features.

5. Conclusions

We present a simple yet effective deep learning framework to create the hash-like binary codes for fast image retrieval. We add a latent-attribute layer in the deep CNN to simultaneously learn domain specific image representations and a set of hash-like functions. Our method does not rely on pairwised similarities of data and is highly scalable to the dataset size. Experimental results show that, with only a simple modification of the deep CNN, our method improves the previous best retrieval results with 1% and 30% retrieval precision on the MNIST and CIFAR-10 datasets, respectively. We further demonstrate the scalability and efficacy of the proposed approach on the large-scale dataset of 1 million shopping images.

Acknowledgement: This work was supported in part by the Ministry of Science and Technology of Taiwan under Contract MOST 103-2221-E-001-010.

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 approach 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