DEEP HASHING WITH MULTI-TASK LEARNING FOR LARGE-SCALE INSTANCE-LEVEL VEHICLE SEARCH

Dawei Liang[1,2], *Ke Yan*[1,2], *Yaowei Wang*[3], *Wei Zeng*[1,2], *Qingsheng Yuan*[4,5,6], *Xiuguo Bao*[5,6], *Yonghong Tian*[1,2*]

[1]National Engineering Laboratory for Video Technology,
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
[2]Cooperative Medianet Innovation Center, China
[3]School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[4]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[5]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[6]National Computer Network Emergency Response Technical Team/Coordination Center of China
{dwliang, keyan, weizeng, yhtian}@pku.edu.cn, yaoweiwang@bit.edu.cn,
yuanqingsheng@iie.ac.cn, baoxiuguo@139.com

ABSTRACT

Hashing is a hot research topic in large-scale image search, due to its low memory cost and fast search speed. Recently, deep hashing, which adapts deep convolutional neural networks into hashing, has attracted much attention. In this paper, we propose a new supervised deep hashing method to deal with large-scale instance-level vehicle search, and make the following contributions. Firstly, multi-task learning is employed to learn the hash code, which exploits the available multiple labels of each vehicle, i.e., ID, model, and color. Secondly, differing from several deep hashing methods, which utilize sigmoid or tanh as the activation function of the hash layer, rectified linear unit is adopted in this paper and shows better performance. Thirdly, taking GoogLeNet as the base network, we show that search performance can be promoted significantly, by learning the network's parameters from scratch on our vehicle data. Finally, we perform extensive experiments on a large-scale dataset with up to one million vehicles. The experimental results demonstrate the effectiveness of the proposed method, which outperforms single task deep hashing methods with classification and triplet ranking losses, respectively.

Index Terms— Hashing, Deep Learning, Multi-Task Learning, Vehicle Search, Large Scale

1. INTRODUCTION

Recent years we have witnessed the tremendously increase of data volume in multimedia, especially in the form of image and video. How to effectively and efficiently store, manage and search image is a big challenge in multimedia community. In the past decades, hashing has emerged as a promising approach to deal with this challenge, due to its low memory cost and fast search speed. Since its introduction in the seminal work of Locality Sensitive Hashing (LSH) [1], lots of hashing methods have been proposed in the literature. Most LSH methods employ random projections to generate the hash code; however, these data-independent methods may be sub-optimal, since they ignore the underlying data distributions and the available semantic information. Recently, data-dependent hashing methods, i.e., learning to hash, have attracted much attention, because they can better exploit the data distributions or semantic labels to generate the hash code. These methods can be roughly categorized into supervised (e.g., [2], [3]), semi-supervised (e.g., [4]), and unsupervised (e.g., [5], [6]). Because more semantic information can be exploited, supervised hashing methods generally perform better than other methods.

Fig. 1. Samples of our dataset with one million vehicles.

[*] Corresponding author: Yonghong Tian (yhtian@pku.edu.cn).

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

Traditional hashing methods mainly consist of two parts, i.e., handcrafted feature extraction and hash functions learning. Since the two parts are designed separately, the generated hash code may be sub-optimal. In the past five years, deep convolutional neural networks (DCNN) have achieved dramatic performance boost on most computer vision tasks, such as image classification [7], object detection [8], semantic segmentation [9], to name just a few. DCNN puts feature extraction and classifier design in a single framework, and learns them in an end-to-end fashion. Due to the success of DCNN in most computer vision tasks, some researchers have adapted DCNN into supervised hashing. According to the form of supervision, these methods can be roughly divided into point-wise (e.g., [10]), pair-wise (e.g., [11], [12]), triplet-wise (e.g., [13], [14], [15]), etc.

The aforementioned methods achieve some success in large-scale image search; however, the main drawback is that several methods adopt sigmoid or tanh or other saturated functions as the activation function of the hash layer, which makes the network hard to be trained. Another drawback is that these methods mainly focus on category-level hashing, while do not make fully use of the instance-level information. For example, in this paper we deal with large-scale instance-level vehicle search. by instance-level search, we mean that given a query vehicle image, all vehicle images with the same license plate number should be returned, and ranked in the top positions. Note that we only use license plate number to assign an ID to each vehicle image, and do not use it in any other form. Besides ID, each vehicle also has other labels, i.e., model and color. To make fully use of multiple labels in instance-level vehicle search, we propose a new supervised deep hashing method and make the following contributions.

Firstly, multi-task learning is employed to learn the hash code, which exploits the available multiple labels of each vehicle, i.e., ID, model, and color. Secondly, differing from several deep hashing methods which use sigmoid or tanh or other saturated functions as the activation function of the hash layer, rectified linear unit is employed and shows better performance. Thirdly, taking GoogLeNet as the base network, we show that search performance can be promoted significantly, by learning the network's parameters from scratch on our vehicle data. Finally, we perform extensive experiments on a large-scale dataset with up to one million vehicles, the samples of which are shown in Fig. 1. The experimental results demonstrate the effectiveness of the proposed method, which outperforms single task deep hashing methods using classification and triplet ranking losses, respectively.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 provides the details of our proposed method. Experiments are performed in section 4. Finally, we conclude the paper with discussions of future work in section 5.

2. RELATED WORKS

2.1. Deep Hashing

Due to the space limit, we refer the interested readers to recent surveys on hashing methods [18], [19], [20], and only focus on recent works on deep hashing. Xia et al. [21] propose a two-step approach to supervised deep hashing. They first decompose the image similarity matrix into binary hash code, and then learn a DCNN given image and hash code pairs. Lai et al. [15] extend Xia et al.'s work [21] into a single framework, where feature learning and hash coding are combined in a single DCNN framework trained by triplet ranking loss. Zhao et al. [14] present a deep hashing approach for multi-label image search, where the authors train a DCNN via listwise ranking loss. However, the loss cannot be directly optimized, and they relax it to triplet ranking loss. Zhang et al. [13] propose a bit-scalable deep hashing method trained with triplet ranking loss, and they also obtain the weight of each hash bit. Lin et al. [10] modify the structure of AlexNet [7], and use image classification loss to learn the hash code. Liu et al. [22] present a supervised deep hashing method, where pair-wise loss is employed. Other deep hashing methods using pair-wise loss include [11], [12]. All the aforementioned methods focus on category-level hashing, and mainly deal with single task.

2.2. Instance Search

In multimedia community, instance search mainly focuses on scene (e.g., INRIA Holidays [23]), buildings (e.g., Oxford5K [24], Paris6K [25]), object (e.g., UKBench [26]), etc. To make large-scale evaluation, usually an enormous amount of distractors downloaded from the internet are added to the database. The distractors are diverse in appearance and may be very different from the query. However, in large-scale instance-level vehicle search, every distractor is also an image of a vehicle. To make the problem even more difficult, two different vehicle instances may share the same model and color (e.g., White Audi Q5). Liu et al. [27] propose a deep relative distance learning approach to vehicle re-identification and search, and test their method on a dataset with 200K vehicle images. Only ID and model labels are available in their dataset; however, our dataset has more complete labels including ID, model and color. Besides, our proposed approach is tested on a dataset with up to one million vehicles, which is significantly larger than theirs.

3. PROPOSED APPROACH

Our proposed approach is shown in Fig. 2. GoogLeNet [16] is chosen as the base network because of its good performance on ImageNet [28] classification. The classifier layers and loss layers are discarded, and a fully-connected layer called hash

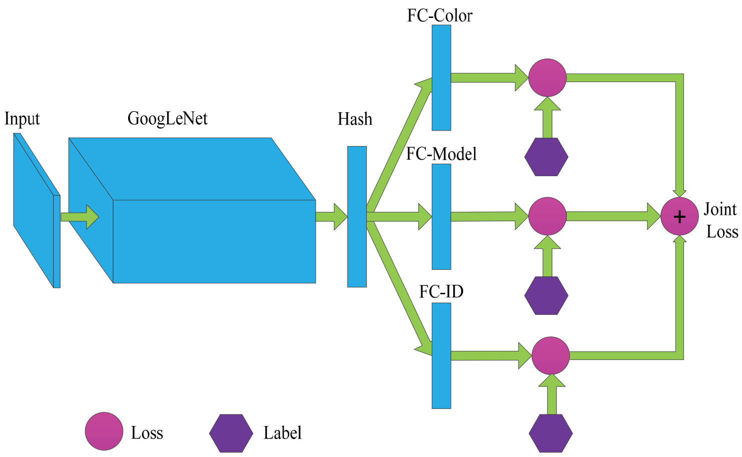

Fig. 2. A schematic diagram of the proposed approach. GoogLeNet [16] is chosen as the base network. The classifier layers and loss layers are discarded, and a fully-connected layer called hash layer is added after the last pooling layer. Then, a rectified linear unit (ReLU) [17] activation function is applied element-wise on the hash layer, which is connected to three fully-connected layers, corresponding to ID, model and color, respectively. Softmax loss is chosen for each fully-connected layer, and the final loss is the sum of these softmax losses.

layer is added after the last pooling layer. Then, a rectified linear unit (ReLU) [17] activation function is applied element-wise on the hash layer. After that three fully-connected layers are directly connected to the hash layer. These fully-connected layers correspond to ID, model and color, respectively. Softmax loss is chosen for each fully-connected layer, and the final loss is the sum of these softmax losses. Denote x_{nk} as the output of the k-th unit in a fully-connected layer of the n-th sample. Then, the softmax function is defined as

$$\sigma(x_{nk}) = \frac{e^{x_{nk}}}{\sum_{k'=1}^{K} e^{x_{nk'}}}, \tag{1}$$

where K is the number of classes. Then, softmax loss for a batch of N samples is defined as

$$loss = -\frac{1}{N}\sum_{n=1}^{N} log(\sigma(x_{nl_n})), \tag{2}$$

where $l_n \in \{0,1,...,K\text{-}1\}$ is the label of the n-th sample. The training of the network is detailed in the experiments section.

After the training process finishes, a test image is propagated through the network in a forward pass, and the hash layer with ReLU activation is binarized to generate the hash code as follows

$$b_i = \begin{cases} 1 & \text{if } h_i>0, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where i is the index of the units in the hash layer.

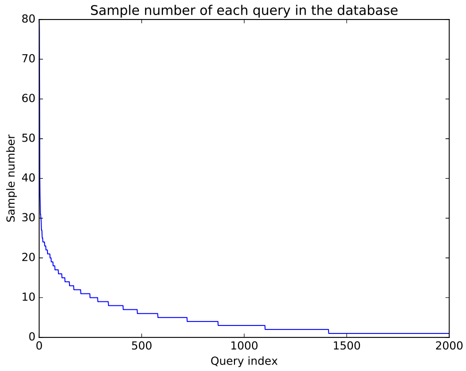

Fig. 3. Sample number of each query in the database.

4. EXPERIMENTS

4.1. Dataset and Experimental Settings

Our vehicle data is collected from real surveillance cameras distributed in a small city. The vehicle data is preprocessed to obtain each vehicle's bounding box, license plate number, model, and color. We use license plate number to assign an ID to each vehicle image. That is to say, two vehicle images sharing the same ID have identical license plate number. We cover each vehicle's license plate number by a mask, to discard its influence on vehicle search. In other words, we evaluate search performance only using each vehicle's appearance, without resorting to it's license plate number.

After some postprocessing, there are 1,097,649 images left. From these images 422,326 images are split as the train-

Table 1. mAP of different methods on small scale dataset (106,887)

Methods	GoogLeNet-Vehicle-ReLU	GoogLeNet-ImageNet-ReLU	GoogLeNet-Vehicle-Sigmoid	GoogLeNet-ImageNet-Sigmoid
ID+model+color	0.821	0.704	0.363	0.323
ID+model	0.810	0.751	0.284	0.245
ID+color	0.806	0.716	0.557	0.153
ID	0.753	0.726	0.465	0.242
model+color	0.522	0.371	0.380	0.328
model	0.344	0.311	0.270	0.253
triplet	0.532	0.280	0.350	0.169

Table 2. mAP of different methods on medium scale dataset (604,032)

Methods	GoogLeNet-Vehicle-ReLU	GoogLeNet-ImageNet-ReLU	GoogLeNet-Vehicle-Sigmoid	GoogLeNet-ImageNet-Sigmoid
ID+model+color	0.613	0.482	0.182	0.150
ID+model	0.606	0.543	0.135	0.109
ID+color	0.611	0.518	0.334	0.068
ID	0.552	0.539	0.256	0.108
model+color	0.314	0.186	0.193	0.158
model	0.184	0.157	0.132	0.121
triplet	0.381	0.170	0.228	0.099

ing set, which has 70,591 ID labels, 1,232 model labels, and 11 color labels. Note that the remaining images have no ID labels in common with the training set. In the remaining images, we randomly select 2000 images with different ID labels as the query set, and construct small, medium, and large scale databases to evaluate search performance. The small, medium, large scale databases contain 106,887, 604,032, and 1,097,649 images, respectively. Each query image has at least one sample in the databases with ID label in common, and the statistics of the sample numbers is shown in Fig. 3. We use Hamming distance to rank images and mean Average Precision (mAP) as the performance measure. mAP is computed using the top 1000 returned images.

We implement our network using the open source deep learning framework Caffe [29]. As the layer immediately before the hash layer in GoogLeNet has 1024 units, we set the number of units in the hash layer also to be 1024, without loss of information. We train the network with a batch size of 60, base learning rate of 0.001, momentum of 0.9, and weight decay of 0.0005. The number of maximum iteration is 100,000, and the base learning rate is decreased with a factor of ten after 50,000 iterations. The weights of new layers are initialized by xavier-type weight filler, and the learning rate of each fully-connected layer after the hash layer is set to be 10 times of the base learning rate. Each image is resized to 256x256, and a random patch of size 224x224 is cropped as the input to the network. We also subtract mean value [104, 117, 123] from each pixel's RGB values. All the following experiments use the same settings described here.

4.2. Comparative Experiments

We perform a series of comparative experiments regarding supervisory signals, activation functions of the hash layer, and the base networks. As for supervisory signals, besides ID+model+color we also consider other possible combinations, such as ID+model, ID+color, ID, model, model+color. We also implement a network with triplet ranking loss [30]. The network structure is the same as Fig. 2 including and before the hash layer. After the hash layer, a L2 normalization layer is inserted between the hash layer and the triplet ranking loss layer. Each triplet is selected according to the ID label. Query and the positive sample have same ID labels, while query and the negative sample have different ID labels but same model and color labels if possible. As for activation functions of the hash layer, We compare the commonly used sigmoid in the deep hashing literature and the ReLU adopted in this paper. As for base networks, We also train a network from scratch with the same structure as GoogLeNet on the same training set with 422,326 images. Since GoogLeNet has three levels of losses, we use model supervision on the first two levels and Id+model+color supervision on the last

Table 3. mAP of different methods on large scale dataset (1,097,649)

Methods	GoogLeNet-Vehicle-ReLU	GoogLeNet-ImageNet-ReLU	GoogLeNet-Vehicle-Sigmoid	GoogLeNet-ImageNet-Sigmoid
ID+model+color	0.532	0.412	0.148	0.118
ID+model	0.524	0.472	0.106	0.087
ID+color	0.532	0.451	0.265	0.053
ID	0.479	0.470	0.210	0.085
model+color	0.267	0.150	0.158	0.127
model	0.151	0.128	0.099	0.098
triplet	0.331	0.146	0.193	0.084

level. We denote this model as GoogLeNet-Vehicle, and the model pre-trained on ImageNet as GoogLeNet-ImageNet. All the experimental results are shown in Table 1, 2, 3 for small, medium, and large scale databases, respectively.

4.3. Discussions

From Table 1, 2, 3, We can observe that methods with ReLU as the activation function of the hash layer consistently outperform their counterparts with sigmoid. The reason is perhaps that networks with sigmoid as the activation function of the hash layer is difficult to be trained, especially when dealing with tens of thousands of classes and hundreds of thousands of training samples. We also see that methods with GoogLeNet-Vehicle as the base network consistently outperform their counterparts with GoogLeNet-ImageNet. This result may imply that the features learned on ImageNet are different from those learned on our vehicle dataset, and we cannot transfer these features directly to deal with instance-level vehicle search. Therefore, we can conclude that the best choices of the network components are ReLU activation function for the hash layer, and GoogLeNet pre-trained on our vehicle dataset as the base network. As far as these components are selected, it can be observed that the best performance is achieved by the multi-task network trained by ID+model+color on all three scale databases, respectively. It outperforms the networks trained by ID and triplet ranking losses with a large margin. We also see that other multi-task networks trained by ID+model and ID+color also outperform the networks trained by ID and triplet ranking losses. And this validates the merits of using multi-task learning to learn the hash code.

Since over one fourth of the queries only have one sample with ID in common in the databases according to Fig. 3, we also evaluate rank one precision with the configuration of GoogLeNet-Vehicle and ReLU activation function. The results are shown in Fig. 4, in which we can see that the results of our multi-task networks outperform the ones with ID and triplet ranking losses on all three scale databases, respectively. It can be observed that the results of ID is significantly better than the ones of model+color. This shows that we can learn instance-specific features, even though two vehicles may have identical model and color.

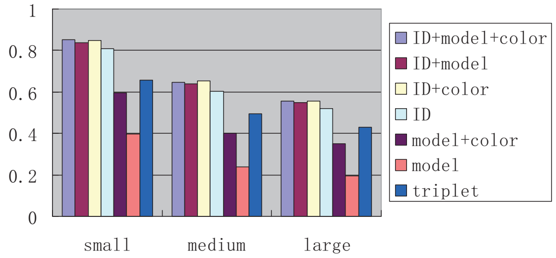

Fig. 4. Rank one precision of networks with GoogLeNet-Vehicle and ReLU on small, medium, and large scale databases, respectively.

5. CONCLUSIONS AND FUTURE WORK

In this paper, We propose a new supervised deep hashing method with multi-task learning, to deal with instance-level vehicle search, and perform extensive experiments on a dataset with up to one million vehicles. We show that the networks with ReLU as the activation function of the hash layer outperform their counterparts with sigmoid, and the networks with base network GoogLeNet pre-trained on our vehicle data outperform their counterparts pre-trained on ImageNet. We also show that multi-task network trained by ID+model+color outperforms other methods in terms of mAP. In future work, we plan to replace GoogLeNet with ResNet [31], since it outperforms GoogLeNet on many vision tasks. We also plan to investigate the influence of different numbers of hash units on the search performance.

6. ACKNOWLEDGEMENTS

This work is partially supported by grants from the National Basic Research Program of China under grant 2015CB351806, the National Natural Science Foundation of

China under contract No. U1611461, No. 61390515, and No. 61425025, and Shenzhen Peacock Plan.

7. REFERENCES

[1] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *STOC*, 1998, pp. 604–613.

[2] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *CVPR*, 2008.

[3] W. Liu, J. Wang, R. Ji, and et al., "Supervised hashing with kernels," in *CVPR*, June 2012, pp. 2074–2081.

[4] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for scalable image retrieval," in *CVPR*, 2010.

[5] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[6] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS*, pp. 1753–1760. 2009.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105. 2012.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, June 2015, pp. 3431–3440.

[10] K. Lin, H. F. Yang, J. H. Hsiao, and C. S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *CVPRW*, June 2015, pp. 27–35.

[11] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *IJCAI*, 2016.

[12] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *AAAI*, 2016.

[13] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE TIP*, vol. 24, no. 12, pp. 4766–4779, Dec 2015.

[14] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *CVPR*, June 2015, pp. 1556–1564.

[15] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *CVPR*, 2015, pp. 3270–3278.

[16] C. Szegedy, W. Liu, Y. Jia, and et al., "Going deeper with convolutions," in *CVPR*, June 2015, pp. 1–9.

[17] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[18] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for Similarity Search: A Survey," *ArXiv e-prints*, Aug. 2014.

[19] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data - A survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2016.

[20] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A Survey on Learning to Hash," *ArXiv*, June 2016.

[21] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, 2014.

[22] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *CVPR*, 2016.

[23] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008, pp. 304–317.

[24] J. Philbin, O. Chum, M. Isard, and et al., "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, June 2007, pp. 1–8.

[25] J. Philbin, O. Chum, M. Isard, and et al., "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, June 2008, pp. 1–8.

[26] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.

[27] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *CVPR*, 2016, pp. 2167–2175.

[28] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, June 2009, pp. 248–255.

[29] Y. Jia, E. Shelhamer, J. Donahue, and et al., "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, 2014, pp. 675–678.

[30] J. Wang, Y. Song, T. Leung, and et al., "Learning fine-grained image similarity with deep ranking," in *CVPR*, June 2014, pp. 1386–1393.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.