# Focusing on Detail: Deep Hashing Based on Multiple Region Details (Student Abstract)

**Quan Zhou,**[1] **Xiushan Nie,**[2,*] **Yang Shi,**[3] **Xingbo Liu,**[3] **Yilong Yin**[1,*]
[1]School of Software, Shandong University, Jinan, P.R. China
[2]School of Computer Science and Technology, Shandong Jianzhu University, Jinan, P.R. China
[3]School of Computer Science and Technology, Shandong University, Jinan, P.R. China
woodschou@outlook.com, {niexiushan, shiy228}@163.com, sclxb@mail.sdu.edu.cn, ylyin@sdu.edu.cn

## Abstract

Fast retrieval efficiency and high performance hashing, which aims to convert multimedia data into a set of short binary codes while preserving the similarity of the original data, has been widely studied in recent years. Majority of the existing deep supervised hashing methods only utilize the semantics of a whole image in learning hash codes, but ignore the local image details, which are important in hash learning. To fully utilize the detailed information, we propose a novel deep multi-region hashing (DMRH), which learns hash codes from local regions, and in which the final hash codes of the image are obtained by fusing the local hash codes corresponding to local regions. In addition, we propose a self-similarity loss term to address the imbalance problem (i.e., the number of dissimilar pairs is significantly more than that of the similar ones) of methods based on pairwise similarity.

## Formulation

Suppose we have a training set $\boldsymbol{X}$(i.e., $\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^{n}$). The pairwise similarity matrix is denoted as $\boldsymbol{S} = \{s_{ij}\}^{n\times n}$, where $s_{ij} = 1$ implies that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, whereas $s_{ij} = 0$ implies dissimilar. For each image $\boldsymbol{x}_i$, the goal is to learn a hash code $\boldsymbol{b}_i \in \{-1, 1\}^c$, where $c$ is the code length.

To successfully utilize the region details, we propose a ConvNet architecture that learns hash codes based on preserving pairwise similarity. The architecture of our method includes a feature learning part and a hashing learning part.

### Feature Learning Part

The backbone of the feature learning part is CNN-F (Chatfield et al. 2014), which is widely used in deep supervised hashing methods. In the CNN-F network, the size of the input image is $224 \times 224$. Therefore, if we wish to learn hash codes for multiple local regions of an image, we need to manually chop up an image into multiple overlapping regions of size $224 \times 224$; this tends to be tedious. Inspired by the discriminator used in PatchGAN (Isola et al. 2017), we modify the structure of CNN-F to obtain multiple outputs that correspond to multiple local regions. Specifically, we replace the fully connected layers of CNN-F with convolution layers. Through this modification, the proposed network can allow for input images of different sizes according to the number of local regions. It can also produce feature maps with sizes greater than $1 \times 1$ in the last layer. As it easy to calculate, the size of the input image in the proposed network is $(192 + 32 * N) \times (192 + 32 * N)$ for $N \times N$ local regions. It is possible to obtain $N^2$ outputs $\{\boldsymbol{h}_{ik}\}_{k=1}^{N^2}$ corresponding to $N^2$ local regions of a given image $\boldsymbol{x}_i$, where $\boldsymbol{h}_{ik}$ is the output of the $k_{th}$ region in the image $\boldsymbol{x}_i$.

### Hashing Learning Part

To preserve the data similarity, we use pairwise similarity as supervision. Let $\boldsymbol{h}_i = \frac{1}{N^2}\sum_{k=1}^{N^2}\boldsymbol{h}_{ik}$ be the output of the feature learning part for the image $\boldsymbol{x}_i$. The notation $\boldsymbol{b}_i = sign(\boldsymbol{h}_i)$ is the hash code of image $\boldsymbol{x}_i$, where $sign()$ is the sign function. There are two loss terms in the initial objective function $J(\theta)$ with the network parameter $\theta$, which is denoted as:

$$J(\theta) = -\frac{1}{n^2}\sum_{s_{ij}\in\boldsymbol{S}}(s_{ij}\Theta_{ij} - log(1 + e^{\Theta_{ij}})) + \frac{\eta}{n \times c}\sum_{i=1}^{n}\|\boldsymbol{b}_i - \boldsymbol{h}_i\|_2^2. \tag{1}$$

Here, the first term represents the similarity loss, which causes $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ to exhibit similarity when $s_{ij} = 1$. The second term is the quantization loss, which causes the output $\boldsymbol{h}_i$ to be close to the corresponding hash code $\boldsymbol{b}_i$. The $\Theta_{ij}$ is set to $\frac{1}{2}(\boldsymbol{h}_i^T\boldsymbol{h}_j)$. The $\eta$ is a hyper-parameter.

However, hashing methods based on pairwise similarity often encounter the class imbalance problem. Some of the existing methods address this issue by modifying the value of similarity, which has adverse effects on the preservation of similarity. In this study, we can increase the number of similar pairs by exploiting the relationship between multiple overlapping regions of the same image. Therefore, we add a self-similarity loss term $M$ to the objective function:

$$M = \frac{1}{n \times N^4}\sum_{i=1}^{n}\sum_{m=1}^{N^2}\sum_{k=1}^{N^2}log(1 + e^{-\Lambda_{imk}}), \tag{2}$$

[*]Xiushan Nie and Yilong Yin are the corresponding authors.

Table 1: MAP@5K on CIFAR-10

| Method | 24bits | 48bits | 64bits | 128bits |
|---|---|---|---|---|
| DSH | 0.7864 | 0.7830 | 0.7834 | 0.7835 |
| DPSH | 0.8821 | 0.8853 | 0.8858 | 0.8876 |
| DSDH | 0.8985 | 0.9004 | 0.9002 | 0.8970 |
| DCH | 0.8753 | 0.8752 | 0.8749 | 0.8273 |
| DDSH | 0.8681 | 0.8875 | 0.8922 | 0.8995 |
| ADSH | 0.9043 | 0.9073 | 0.9073 | 0.9072 |
| DMRH | **0.9252** | **0.9251** | **0.9288** | **0.9243** |

Table 2: MAP@5K on NUS-WIDE

| Method | 24bits | 48bits | 64bits | 128bits |
|---|---|---|---|---|
| DSH | 0.6598 | 0.6653 | 0.6587 | 0.6598 |
| DPSH | 0.8390 | 0.8429 | 0.8423 | 0.8468 |
| DSDH | 0.8225 | 0.8328 | 0.8347 | 0.8415 |
| DCH | 0.7552 | 0.7632 | 0.7647 | 0.7602 |
| DDSH | 0.7672 | 0.8171 | 0.8161 | 0.8008 |
| ADSH | **0.8962** | **0.9030** | **0.9035** | **0.8927** |
| DMRH | 0.8831 | 0.8845 | 0.8857 | 0.8879 |

where $\Lambda_{imk} = \frac{1}{2}(\boldsymbol{h}_{im}^T \boldsymbol{h}_{ik})$.

Therefore, the optimized problem of DMRH is

$$\begin{aligned} \underset{\boldsymbol{\theta}}{min} & - \frac{1}{n^2} \sum_{s_{ij} \in \boldsymbol{S}} (s_{ij}\Theta_{ij} - log(1 + e^{\Theta_{ij}})) \\ & + \frac{\gamma}{n \times N^4} \sum_{i=1}^{n} \sum_{m=1}^{N^2} \sum_{k=1}^{N^2} log(1 + e^{-\Lambda_{imk}}) \qquad (3) \\ & + \frac{\eta}{n \times c} \sum_{i=1}^{n} \|\boldsymbol{b}_i - \boldsymbol{h}_i\|_2^2, \end{aligned}$$

where $\gamma$ and $\eta$ are hyper-parameters.

## Experiments

We evaluate the DMRH model using three datasets: CIFAR-10, NUS-WIDE, and MS-COCO. The mean average precision top-5K (MAP@5K) is used as the evaluation metric.

In the experiment, six deep supervised hashing methods are chosen for comparison with DMRH; these are DSH (Liu et al. 2016), DPSH (Li, Wang, and Kang 2016), DSDH (Li et al. 2017), DCH (Cao et al. 2018), DDSH (Jiang, Cui, and Li 2018) and ADSH (Jiang and Li 2018).

We initialize our network with the pre-trained CNN-F model on ImageNet. We set the batch size to 32, and the weight decay to $1e-5$. The initial learning rate is set to $0.1$ and decreases by $90\%$ after every 50 epochs (150 epochs in total). $\gamma$ and $\eta$ are $5e-2$ and $2e-2$, respectively.

The results for the three datasets are listed in Tables 1-3, where $N = 6$ for the DMRH model. DMRH performs better than the other models in most cases, especially with MS-COCO. The performance of DMRH with different $N$ in CIFAR-10 is illustrated in Fig. 1 (limited by the space), from which we can observe that a larger $N$ leads to a better performance. However, the performance of the proposed method drops when the value of $N$ is greater than a fixed value(e.g. $N = 6$ on CIFAR-10). This is because a larger number of local regions will result in the size of each local region becoming smaller. In addition, this adds noise, which degrades the semantic representation of images.

Table 3: MAP@5K on MS-COCO

| Method | 24bits | 48bits | 64bits | 128bits |
|---|---|---|---|---|
| DSH | 0.5135 | 0.5069 | 0.5147 | 0.5072 |
| DPSH | 0.6623 | 0.6871 | 0.6965 | 0.7073 |
| DSDH | 0.6988 | 0.7191 | 0.7220 | 0.7227 |
| DCH | 0.5858 | 0.5954 | 0.5948 | 0.5953 |
| DDSH | 0.5807 | 0.6004 | 0.6127 | 0.6292 |
| ADSH | 0.6605 | 0.6596 | 0.6648 | 0.6762 |
| DMRH | **0.7680** | **0.7972** | **0.7991** | **0.8120** |

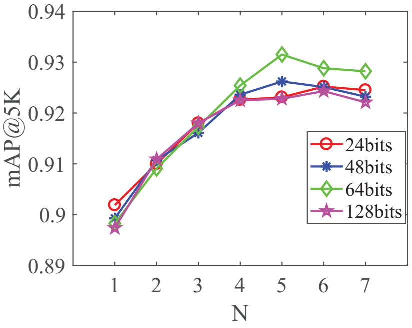

Figure 1: mAP@5K of different N on CIFAR-10

## References

Cao, Y.; Long, M.; Liu, B.; and Wang, J. 2018. Deep cauchy hashing for hamming space retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1229–1237.

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Jiang, Q.-Y., and Li, W.-J. 2018. Asymmetric deep supervised hashing. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jiang, Q.-Y.; Cui, X.; and Li, W.-J. 2018. Deep discrete supervised hashing. *IEEE Transactions on Image Processing* 27(12):5996–6009.

Li, Q.; Sun, Z.; He, R.; and Tan, T. 2017. Deep supervised discrete hashing. In *Advances in Neural Information Processing Systems*, 2479–2488.

Li, W. J.; Wang, S.; and Kang, W. C. 2016. Feature learning based deep supervised hashing with pairwise labels. In *International Joint Conference on Artificial Intelligence*.

Liu, H.; Wang, R.; Shan, S.; and Chen, X. 2016. Deep supervised hashing for fast image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2064–2072.