



Deep Disentangled Hashing with Momentum Triplets for Neuroimage Search

Erkun Yang¹, Dongren Yao^{1,2}, Bing Cao¹, Hao Guan¹, Pew-Thian Yap¹,
Dinggang Shen¹, and Mingxia Liu¹✉

¹ Department of Radiology and BRIC, University of North Carolina at Chapel Hill,
Chapel Hill, NC 27599, USA

mxliu@med.unc.edu

² Brainnetome Center and National Laboratory of Pattern Recognition, Institute
of Automation, University of Chinese Academy of Sciences, Beijing 100190, China

Abstract. Neuroimaging has been widely used in computer-aided clinical diagnosis and treatment, and the rapid increase of neuroimage repositories introduces great challenges for efficient neuroimage search. Existing image search methods often use triplet loss to capture high-order relationships between samples. However, we find that the traditional triplet loss is difficult to pull positive and negative sample pairs to make their Hamming distance discrepancies larger than a small fixed value. This may reduce the discriminative ability of learned hash code and degrade the performance of image search. To address this issue, in this work, we propose a *deep disentangled momentum hashing* (DDMH) framework for neuroimage search. Specifically, we first investigate the original triplet loss and find that this loss function can be determined by the inner product of hash code pairs. Accordingly, we disentangle hash code norms and hash code directions and analyze the role of each part. By decoupling the loss function from the hash code norm, we propose a unique *disentangled triplet loss*, which can effectively push positive and negative sample pairs by desired Hamming distance discrepancies for hash codes with different lengths. We further develop a *momentum triplet strategy* to address the problem of insufficient triplet samples caused by small batch-size for 3D neuroimages. With the proposed disentangled triplet loss and the momentum triplet strategy, we design an end-to-end trainable deep hashing framework for neuroimage search. Comprehensive empirical evidence on three neuroimage datasets shows that DDMH has better performance in neuroimage search compared to several state-of-the-art methods.

1 Introduction

Neuroimage analysis plays a vital role in modern clinical analysis [1], image-guided surgery [2], and automated disease diagnosis [3]. Nowadays, tremendous amounts of neuroimaging data have been captured and recorded in digital formats. However, interpreting neuroimaging data usually requires extensive practical experience and professional knowledge, and may be affected by inter-observer

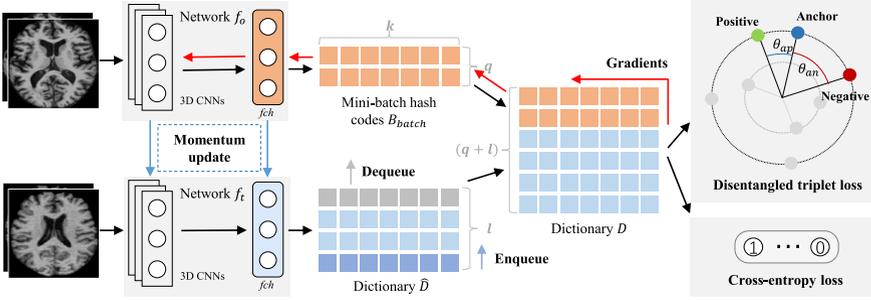


Fig. 1. Proposed deep disentangled momentum hashing (DDMH) framework for neuroimage search, including three key components: (1) a disentangled triplet loss, which improves the traditional triplet loss by decoupling it from the hash code length; (2) a cross-entropy loss, which is used to optimize the hash codes by jointly learning a linear classifier; (3) a momentum triplet strategy, which builds a large and consistent dictionary on-the-fly and enables efficient triplet-based learning even with a small batch-size.

variance [4, 5]. To facilitate clinical decision-making, it is important to provide physicians with previous similar cases and corresponding treatment records to form case-based reasoning and evidence-based medicine [6]. In practice, such neuroimage search systems are required to be both scalable and accurate. Due to a sensible balance between search quality and computational cost, hashing-based search methods [7, 8], which encode neuroimages as binary embeddings in Hamming space, have recently attracted increasing attention in the field.

According to whether supervisory signals are involved in the learning phase, existing hashing methods can be roughly divided into two groups: 1) *unsupervised hashing* [9–11] and 2) *supervised hashing* [12–18]. Unsupervised methods usually learn hash functions from original feature space to Hamming space by exploiting topological information and data distributions. In contrast, supervised hashing incorporates semantic labels to improve the quality of hash function learning. In this paper, we focus on supervised hashing, as it can take advantage of the supervisory signals and usually outperforms unsupervised methods.

Existing supervised hashing methods usually rely on ranking-based loss functions [15–17, 19–21] to preserve similarity between samples, such as contrastive loss and triplet loss. Recent studies [22, 23] have shown that the triplet loss encourages samples from the same category to reside on a manifold and generally outperforms the contrastive loss. Even though many triplet-based hashing methods have been proposed [17, 19], we find a crucial misspecification problem in the traditional triplet loss, that is, traditional triplet loss is difficult to pull positive and negative pairs to make their Hamming distance discrepancies larger than a small fixed value. This may reduce the discriminative ability of the learned hash code, thereby degrading the performance of triplet loss-based methods.

To this end, we propose a supervised hash model, called *deep disentangled momentum hashing* (DDMH), for neuroimage search. As illustrated in Fig. 1, the proposed DDMH consists of three key components: (1) a disentangled triplet loss, which decouples the traditional triplet loss with hash code norms and can effectively pull positive and negative pairs by a desired Hamming distance discrepancies for hash codes with different lengths; (2) a cross-entropy loss, which is used to optimize the hash codes with a jointly learned linear classifier; (3) a momentum triplet strategy to enable efficient triplet-based learning even with a small batch-size. These three components are seamlessly incorporated into an end-to-end trainable deep hashing framework for neuroimage search. Extensive experiments on three neuroimage datasets demonstrate that the proposed DDMH outperforms several state-of-the-art methods in neuroimage search.

2 Method

2.1 Problem Formulation

Boldface uppercase letters (e.g., \mathbf{A}) denote matrices, and \mathbf{A}^T represents the transpose of a matrix \mathbf{A} . Also, $\text{sign}(\cdot)$ represents the element-wise signum function, defined as

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases} \quad (1)$$

Neuroimage hashing aims to learn hash functions that can map neuroimages into hash codes in Hamming space. Here, hash codes represent binary vectors, and the Hamming space contains a set of binary vectors. Assume that we have N neuroimages in a given dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, with each image \mathbf{x}_i associated with a label vector \mathbf{l}_i . We denote $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^N$ as the hash codes for \mathbf{X} , where $\mathbf{b}_i \in \{-1, +1\}^k$ represents the binary hash code for the sample \mathbf{x}_i , k is the hash code length, and b_{ix} indicates the x -th element of \mathbf{b}_i .

2.2 Deep Disentangled Hash Code Learning

For effective neuroimage search, we aim to learn discriminative binary codes to well preserve the original semantic information, that is, semantically similar neuroimages are expected to have similar hash codes and vice versa. To achieve this goal, we employ a triplet-based ranking loss, which is defined as

$$\mathcal{L}_o(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n) = \log(1 + e^{d_h(\mathbf{b}_a, \mathbf{b}_p) - d_h(\mathbf{b}_a, \mathbf{b}_n)}) = \log(1 + e^{-\alpha}), \quad (2)$$

where \mathbf{b}_a is the learned hash code for an anchor neuroimage \mathbf{x}_a , \mathbf{b}_p is the binary code for a positive neuroimage \mathbf{x}_p that shares the same semantic label with \mathbf{x}_a , and \mathbf{b}_n is the binary code for a negative neuroimage \mathbf{x}_n that has a different label from \mathbf{x}_a . Also, $(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n)$ is a randomly selected triplet sample. $(\mathbf{b}_a, \mathbf{b}_p)$ represents a positive pair and $(\mathbf{b}_a, \mathbf{b}_n)$ denotes a negative pair. $d_h(\cdot, \cdot)$ is the Hamming distance function for a pair of neuroimages. $\alpha = d_h(\mathbf{b}_a, \mathbf{b}_n) - d_h(\mathbf{b}_a, \mathbf{b}_p)$

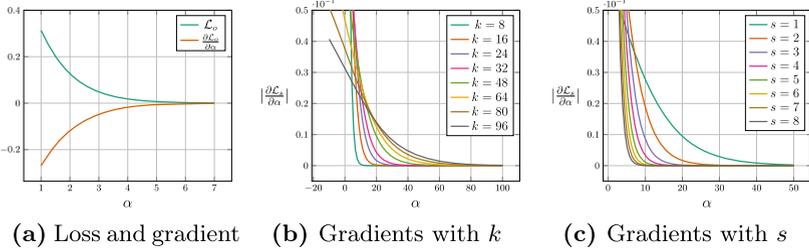


Fig. 2. The loss and gradients of conventional triplet loss \mathcal{L}_o w.r.t. α is close to 0 when α is larger than 6, which is ill-specified for hash code learning. By replacing α with $s\beta$, the gradients of our disentangled triplet loss \mathcal{L}_s are more meaningful for different k . We set $s = 3$ in (b) and $k = 16$ in (c).

represents the discrepancy between the corresponding Hamming distances. If there is no ambiguity, we will use \mathcal{L}_o to represent $\mathcal{L}_o(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n)$ in the following.

Since the Hamming distance function $d_h(\mathbf{b}_i, \mathbf{b}_j) = |b_{ix} \neq b_{jx}, 1 \leq x \leq k|$ requires discrete operations and is non-differentiable, directly optimizing Eq. (2) with back propagation is not feasible. As indicated in [12], there exists a nice relationship between the inner product and the Hamming distance as

$$d_h(\mathbf{b}_i, \mathbf{b}_j) = \frac{k - \langle \mathbf{b}_i, \mathbf{b}_j \rangle}{2}. \tag{3}$$

Thus, we can obtain the following

$$\alpha = \frac{k - \langle \mathbf{b}_a, \mathbf{b}_n \rangle}{2} - \frac{k - \langle \mathbf{b}_a, \mathbf{b}_p \rangle}{2} = \frac{\langle \mathbf{b}_a, \mathbf{b}_p \rangle - \langle \mathbf{b}_a, \mathbf{b}_n \rangle}{2}. \tag{4}$$

Substituting Eq. (4) into Eq. (2), the loss function \mathcal{L}_o can be readily optimized.

Although the loss in Eq. (2) can help preserve the original semantic information, we observe a key misspecification problem when directly applying Eq. (2) to hash code learning. As illustrated in Fig. 2 (a), when the Hamming distance discrepancy α is large than 6, the loss function value of \mathcal{L}_o and its gradients (w.r.t. α) are close to 0. This implies that, no matter how large the hash code length k is, it is difficult to pull the Hamming distance between positive and negative pairs to be greater than 6 by optimizing the objective function \mathcal{L}_o . For effective hash code learning, we generally expect that longer hash codes can be more discriminative and the Hamming distance discrepancy α can be increased with the increase of the hash code length. Unfortunately, the triplet loss function in Eq. (2) is misspecified for this goal.

To tackle the above misspecification problem, we investigate the relationship between the loss function \mathcal{L}_o and the hash code length k carefully. From Eq. (2) and Eq. (4), we can see that the loss function \mathcal{L}_o can be determined by the inner product of hash code pairs. Considering that both the norm and the direction of a vector can affect the inner product value, we disentangle these two items as

$$\alpha = \frac{k}{2} (\cos \theta_{ap} - \cos \theta_{an}) = \alpha \beta, \tag{5}$$

where θ_{ap} is the angle between \mathbf{b}_a and \mathbf{b}_p , θ_{an} is the angle between \mathbf{b}_a and \mathbf{b}_n , $o = \frac{k}{2}$, and $\beta = \cos \theta_{ap} - \cos \theta_{an}$. It should be noted that this work performs the disentanglement to deal with the misspecification problem in Eq. (2), which is different from previous studies [24, 25] in face verification. Actually, for a given fixed hash code length k , the cosine similarity discrepancy β is proportional to the Hamming distance discrepancy α , while for different hash code length k , α up-weights β with o . By combining Eq. (5) and Eq. (2), one can observe that when the hash code length k is relatively large, the weight o may greatly hinder the optimization. To eliminate this influence, we replace the weight o with a new scale factor s and design a unique *disentangled triplet loss* as follows

$$\mathcal{L}_s(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n) = \log(1 + e^{s(\cos \theta_{an} - \cos \theta_{ap})}) = \log(1 + e^{-\frac{2s}{k}\alpha}). \quad (6)$$

The gradients of $\mathcal{L}_s(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n)$ w.r.t. the Hamming distance discrepancy α with different k and s are illustrated in Fig. 2 (b) and (c), respectively. From these figures we can see that the saturation region can be tuned by selecting different s . That is, by using a proper s , we can decouple the hash code length from the loss function and enable desired Hamming distance discrepancies for hash codes with different lengths, which helps generate discriminative hash codes.

2.3 Overall Objective Function

Following previous studies [26, 27], we assume that optimal hash codes can be jointly learned with a classification task, and use a linear classifier $f(\cdot)$ to model the relationship between learned binary codes and semantic labels, defined as

$$f(\mathbf{b}_i) = \sigma(\mathbf{W}^\top \mathbf{b}_i), \quad (7)$$

where σ is the sigmoid function and \mathbf{W} is the parameter. A cross-entropy loss is then adopted as

$$\mathcal{L}_c(\mathbf{b}_i) = -l_i^\top \ln f(\mathbf{b}_i). \quad (8)$$

By combining Eq. (8) and Eq. (6), we can obtain the overall loss function as

$$\mathcal{L} = \sum_{(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n) \in \mathcal{T}} \mathcal{L}_s(\mathbf{b}_a, \mathbf{b}_p, \mathbf{b}_n) + \lambda \sum_{i=1}^N \mathcal{L}_c(\mathbf{b}_i), \quad (9)$$

where \mathcal{T} is the set for all triplets, and λ is a hyper-parameter to balance the contributions of the disentangled triplet loss and the cross-entropy loss in Eq. (9).

2.4 Optimization with Momentum Triplets

A critical challenge for optimizing our triplet-based 3D deep model is the insufficient number of triplets. Actually, to construct a sufficient amount of triplets, triplet-based deep models often require a relatively large batch size to include more samples in a mini-batch. However, the memory requirement and computation time will increase as the batch size increases, which is particularly significant

for 3D convolution neural networks (CNNs). Therefore, it is desirable to design a feasible solution to optimize our model in Eq. (9) using 3D neuroimages.

Motivated by [28], we introduce a momentum strategy to supervised hashing and propose *momentum triplets* to address the above insufficient triplet problem. As shown in Fig. 1, we design two 3D CNNs, i.e., f_o and f_t , which share the same architecture. Denote q as the mini-batch size. For each mini-batch entity $\mathbf{X}_{batch} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$, we calculate their corresponding hash codes $\mathbf{B}_{batch} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_q\}$ by forward propagating them from f_o . We also maintain a l -length dynamic dictionary $\hat{D} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \dots, \hat{\mathbf{b}}_l\}$, with each element obtained from the network f_t . In each iteration, the dictionary keys of \hat{D} are updated as a queue, with the current mini-batch enqueued and the oldest mini-batch dequeued. To facilitate the construction of more triplets, for $\mathbf{X}_{batch} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$, we concatenate its corresponding codes \mathbf{B}_{batch} with the dynamic dictionary \hat{D} , yielding a new dictionary $D = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_q, \hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \dots, \hat{\mathbf{b}}_l\}$ with the length of $q + l$. The network f_o are then updated with the loss obtained with all triplets from the dictionary D by standard back-propagation. And the network f_t is updated with a momentum strategy via

$$\theta_t = m\theta_t + (1 - m)\theta_o, \quad (10)$$

where θ_o and θ_t are parameters of f_o and f_t , respectively, and $m \in (0, 1]$ is a momentum coefficient. The momentum update strategy for f_t is used to reduce the discrepancy between f_o and f_t and improve the hash code consistency.

2.5 Implementation

The 3D networks for f_o and f_t share the same architecture, which has 12 layers. The first 10 layers are 3D convolutional (Conv) layers (kernel size: $3 \times 3 \times 3$; stride: 1; zero padding). The last 2 layers are fully-connected (FC) layers with 256 and k hidden units, respectively. Batch normalization and ReLU activation are applied to all Conv layers. A 3D max-pooling with stride 2 is also applied to every two Conv layers. For the last FC layer, we use $\text{Tanh}(\cdot)$ to squeeze activation within $[-1, +1]$ to reduce quantization error and also denote it as fully-connected hash (*fch*) layer. We train the network from scratch with Adam [29] by setting the batch size as 2 and the dictionary size l as 10. The learning rate is set to 10^{-3} , λ in Eq. (9) is set to 1, the scale factor s is set to 3, and m is set to 0.999. Assuming the computation cost for one similarity calculation is d , the computation cost to rank all points will be $O(dN)$ and can be further reduced to $O(1)$ with predefined hash look-up tables.

3 Experiments

3.1 Experimental Setup

Materials. We evaluate the proposed DDMH method on three popular benchmark datasets: (1) Alzheimer’s Disease Neuroimaging Initiative (ADNI1) [30],

Table 1. The MAP results of eight different methods on three datasets.

Method	ADNI1				ADNI2				AIBL			
	8	16	24	32	8	16	24	32	8	16	24	32
SH	0.403	0.389	0.386	0.384	0.422	0.377	0.375	0.372	0.613	0.611	0.605	0.604
SpH	0.398	0.398	0.400	0.391	0.375	0.384	0.385	0.375	0.599	0.587	0.584	0.578
ITQ	0.391	0.396	0.396	0.393	0.398	0.389	0.384	0.390	0.596	0.582	0.579	0.577
DSH	0.390	0.402	0.398	0.396	0.408	0.387	0.393	0.394	0.596	0.577	0.575	0.576
KSH	0.372	0.384	0.386	0.391	0.385	0.369	0.376	0.374	0.600	0.591	0.588	0.583
DPSH	0.438	0.460	0.447	0.467	0.403	0.413	0.415	0.391	0.702	0.701	0.700	0.684
SSDH	0.421	0.427	0.435	0.422	0.397	0.408	0.406	0.411	0.712	0.710	0.715	0.698
Ours	0.478	0.480	0.498	0.511	0.602	0.612	0.618	0.632	0.758	0.763	0.761	0.764

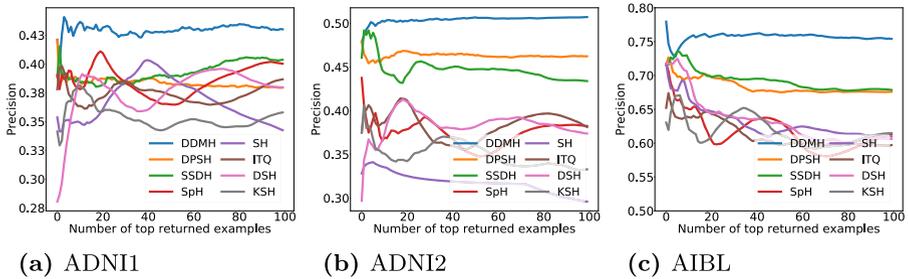


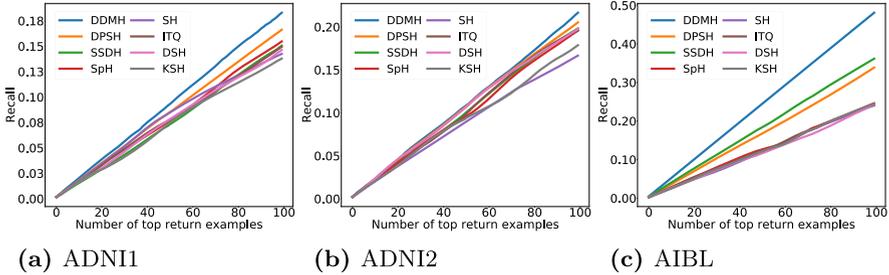
Fig. 3. The topN-precision results of different methods on three datasets.

(2) ANDI2, and (3) Australian Imaging, Biomarkers and Lifestyle dataset (AIBL) [31]. ADNI1 contains 821 subjects with 1.5T T1-weighted structural MRI scans. ADNI2 contains 636 subjects with 3T T1-weighted MRI scans. Besides, AIBL contains 614 subjects with 3T T1-weighted MRI data. Each subject from these three datasets was annotated by a category-level label, *i.e.*, Alzheimer’s disease (AD), normal control (NC), or mild cognitive impairment (MCI). For each dataset, we randomly select 10% samples as a query set and others as a retrieval set. All MRIs were pre-processed via skull-stripping, intensity correction, and spatial normalization to Automated Anatomical Labeling (AAL) template.

Evaluation Setting. Our DDMH is compared with five state-of-the-art shallow hashing methods, *i.e.*, SH [32], SpH [33], ITQ [34], DSH [35], and KSH [12], as well as two recent deep hashing methods, *i.e.*, DPSH [36] and SSDH [37]. Volumes of gray matter tissue inside 90 regions-of-interest (defined in AAL) are used as features for shallow hashing methods, while two deep models and our DDMH employ MRI scans as input. mean of average precision (MAP), topN-precision, and recall@K are used as evaluation metrics.

Table 2. The MAP values of DDMH and its two variants on ADNI1.

Method	$k = 8$	$k = 16$	$k = 24$	$k = 32$
DDMH-C	0.441	0.453	0.456	0.457
DDMH-D	0.462	0.460	0.459	0.466
DDMH	0.478	0.480	0.498	0.511

**Fig. 4.** The recall@k results of different methods on three datasets.

3.2 Results and Discussion

We first report MAP values of all methods with different lengths of hash code (*i.e.*, k) to provide a global evaluation. Then we report the topN-precision and recall@K curves with $k = 16$ for comprehensive contrastive study.

MAP. The MAP results achieved by all methods on the three datasets are reported in Table 1. The results suggest that, compared with shallow hashing methods, deep learning approaches usually yield better performance. This could be attributed to the fact that deep networks enable joint learning of image representations and hash codes directly from the original image. Besides, our DDMH usually outperforms other competing methods by large margins.

TopN-Precision. The topN-precision curves are shown in Fig. 3. This figure suggests that DDMH generally achieves higher precision, which is consistent with the MAP evaluation. It is worth noting that, in the task of neuroimage retrieval, users usually focus on top returned instances. Therefore, it is essential to provide users with top returned instances that are highly relevant to the query. From the results, one can see that DDMH outperforms other methods by a large margin when the number of returned instances is small. This verifies that DDMH can effectively return relevant examples and is suitable for neuroimage search.

Recall@K. The recall@K curves of all methods are reported in Fig. 4. From Fig. 4, one can observe that our DDMH always yields higher recall values for different numbers of returned examples on three datasets, which further proves that DDMH can effectively learn hash codes with high quality.

Ablation Study. We investigate two variants of our DDMH, including (1) DDMH-C without using the cross-entropy loss, *i.e.*, $\lambda = 0$ in Eq. (9); and (2)

DDMH-D with the traditional triplet loss *i.e.*, $s = \frac{k}{2}$ in Eq. (6). MAP results of DDMH and its two variants on ADNI1 are reported in Table 2. From the results, we can see that DDMH is consistently superior to its two variants, implying that both the classification loss and the disentangled triplet loss contribute to the performance improvement. In particular, DDMH achieves higher MAP values using longer hash codes, but DDMH-D benefits little from longer hash codes. These results show that compared with the traditional triplet loss (used in DDMH-D), the proposed disentangled loss (used in DDMH) can effectively promote the learning of longer hash codes. On the other hand, from Tables 1 and 2, one can observe that DDMH-D generally outperforms existing state-of-the-art hashing methods on ADNI1. This implies that, with the proposed momentum triplet strategy, DDMH-D enables efficient hashing learning even with the traditional triplet loss function.

4 Conclusion

This paper presents a deep disentangled momentum hashing (DDMH) method for neuroimage search. Specifically, by disentangling the hash code norm and directions, we propose a unique disentangled triplet loss, which can pull positive and negative pairs to have a desired Hamming distance discrepancy with different hash code lengths. We also design a momentum triplet strategy to provide a sufficient number of triplets for model training, even with a small batch-size. Using a 3D CNN as hash functions, DDMH can generate discriminative hash codes by jointly optimizing the disentangled triplet loss and a cross-entropy loss. Extensive experiments on three neuroimage datasets show that our DDMH achieves better performance than several state-of-the-art methods.

Acknowledgments. This work was partly supported by NIH grants (Nos. AG041721, AG053867).

References

1. Graham, R.N., Perriss, R., Scarsbrook, A.F.: DICOM demystified: a review of digital file formats and their use in radiological practice. *Clin. Radiol.* **60**(11), 1133–1140 (2005)
2. Grimson, W.E.L., Kikinis, R., Jolesz, F.A., Black, P.: Image-guided surgery. *Sci. Am.* **280**(6), 54–61 (1999)
3. Owais, M., Arsalan, M., Choi, J., Park, K.R.: Effective diagnosis and treatment through content-based medical image retrieval (CBMIR) by using artificial intelligence. *J. Clin. Med.* **8**(4), 462 (2019)
4. Cheng, B., Liu, M., Shen, D., Li, Z., Zhang, D.: Multi-domain transfer learning for early diagnosis of Alzheimer’s disease. *Neuroinformatics* **15**(2), 115–132 (2017)
5. Liu, M., Zhang, J., Adeli, E., Shen, D.: Landmark-based deep multi-instance learning for brain disease diagnosis. *Med. Image Anal.* **43**, 157–168 (2018)
6. Holt, A., Bichindaritz, I., Schmidt, R., Perner, P.: Medical applications in case-based reasoning. *Knowl. Eng. Rev.* **20**(3), 289–292 (2005)

7. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: ICCV, pp. 2130–2137 (2009)
8. Yang, E., Deng, C., Liu, W., Liu, X., Tao, D., Gao, X.: Pairwise relationship guided deep hashing for cross-modal retrieval. In: AAAI, pp. 1618–1625 (2017)
9. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(6), 1092–1104 (2012)
10. Dai, B., Guo, R., Kumar, S., He, N., Song, L.: Stochastic generative hashing. arXiv preprint [arXiv:1701.02815](https://arxiv.org/abs/1701.02815) (2017)
11. Yang, E., Deng, C., Liu, T., Liu, W., Tao, D.: Semantic structure-based unsupervised deep hashing. *IJCA I*, 1064–1070 (2018)
12. Liu, W., Wang, J., Ji, R., Jiang, Y., Chang, S.F.: Supervised hashing with kernels. In: CVPR, pp. 2074–2081 (2012)
13. Gui, J., Liu, T., Sun, Z., Tao, D., Tan, T.: Fast supervised discrete hashing. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 490–496 (2017)
14. Yang, E., Deng, C., Li, C., Liu, W., Li, J., Tao, D.: Shared predictive cross-modal deep quantization. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5292–5303 (2018)
15. Cao, Y., Long, M., Liu, B., Wang, J., Kliss, M.: Deep Cauchy hashing for hamming space retrieval. In: CVPR, pp. 1229–1237 (2018)
16. Cao, Y., Liu, B., Long, M., Wang, J., Kliss, M.: HashGAN: deep learning to hash with pair conditional Wasserstein GAN. In: CVPR, pp. 1287–1296 (2018)
17. Zhang, R., Lin, L., Zhang, R., Zuo, W., Zhang, L.: Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Trans. Image Process.* **24**(12), 4766–4779 (2015)
18. Deng, C., Yang, E., Liu, T., Li, J., Liu, W., Tao, D.: Unsupervised semantic-preserving adversarial hashing for image search. *IEEE Trans. Image Process.* **28**(8), 4032–4044 (2019)
19. Deng, C., Chen, Z., Liu, X., Gao, X., Tao, D.: Triplet-based deep hashing network for cross-modal retrieval. *IEEE Trans. Image Process.* **27**(8), 3893–3903 (2018)
20. Chen, L., Honeine, P., Qu, H., Zhao, J., Sun, X.: Correntropy-based robust multi-layer extreme learning machines. *Pattern Recogn.* **84**, 357–370 (2018)
21. Chen, L., Qu, H., Zhao, J., Chen, B., Principe, J.C.: Efficient and robust deep learning with correntropy-induced loss function. *Neural Comput. Appl.* **27**(4), 1019–1031 (2016)
22. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: CVPR, pp. 815–823 (2015)
23. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint [arXiv:1703.07737](https://arxiv.org/abs/1703.07737) (2017)
24. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: CVPR, pp. 4690–4699 (2019)
25. Wang, H., et al.: CosFace: large margin cosine loss for deep face recognition. In: CVPR, pp. 5265–5274 (2018)
26. Chen, Z., Cai, R., Lu, J., Feng, J., Zhou, J.: Order-sensitive deep hashing for multimorbidity medical image retrieval. In: Frangi, A., Schnabel, J., Davatzikos, C., Alberola-Lopez, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11070. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00928-1_70
27. Li, Q., Sun, Z., He, R., Tan, T.: Deep supervised discrete hashing. In: NeurIPS, pp. 2482–2491 (2017)
28. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint [arXiv:1911.05722](https://arxiv.org/abs/1911.05722) (2019)

29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR (2014)
30. Jack Jr, C.R., et al.: The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging Offic. J. Int. Soc. Magn. Reson. Med.* **27**(4), 685–691 (2008)
31. Ellis, K.A., et al.: The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer’s disease. *Int. Psychogeriatr.* **21**(4), 672–687 (2009)
32. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: NeurIPS, pp. 1753–1760 (2009)
33. Heo, J.P., Lee, Y., He, J., Chang, S.F., Yoon, S.E.: Spherical hashing. In: CVPR, pp. 2957–2964 (2012)
34. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(12), 2916–2929 (2013)
35. Jin, Z., Li, C., Lin, Y., Cai, D.: Density sensitive hashing. *IEEE Trans. Cybern.* **44**(8), 1362–1371 (2014)
36. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. *IJCA I*, 1711–1717 (2016)
37. Yang, H.F., Lin, K., Chen, C.S.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 437–451 (2017)