2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897379

ADAPTIVE PROXY ANCHOR LOSS FOR DEEP METRIC LEARNING

Nguyen Phan[⋆†] *Sen Tran*[⋆†] *Ta Duc Huy*[†] *Soan T. M. Duong*[†§]
Chanh D. Tr. Nguyen[†‡] *Trung Bui* *Steven Q.H. Truong*[†]

[†] VinBrain JSC., Vietnam; [‡] VinUniversity, Vietnam; [§] Le Quy Don Technical University, Vietnam

ABSTRACT

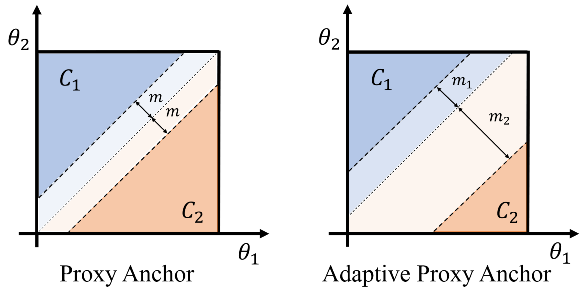
Deep metric learning (or simply called metric learning) uses the deep neural network to learn the representation of images, leading to widely used in many applications, e.g. image retrieval and face recognition. In the metric learning approaches, proxy anchor takes advantage of proxy-based and pair-based approaches to enable fast convergence time and robustness to noisy labels. However, in training the proxy anchor, selecting the hyperparameter margin is important to achieve a good performance. This selection requires expertise and is time-consuming. This paper proposes a novel method to learn the margin while training the proxy anchor approach adaptively. The proposed adaptive proxy anchor simplifies the hyperparameter tuning process while advancing the proxy anchor. We achieve state of the art on three public datasets with a noticeably faster convergence time. Our code is available at `https://github.com/tks1998/Adaptive-Proxy-Anchor`

***Index Terms*—** Deep metric learning, image retrieval, proxy-based loss, proxy anchor, adaptive margin

1. INTRODUCTION

Recently, deep metric learning (DML) has been of interest due to its visual recognition applications, e.g. face recognition and image retrieval [1–6]. DML aims to learn a representation function, mapping the images of objects to an embedded space in which embeddings of the same-class data are closed and ones of the different classes are far apart [7]. Several DML methods have been proposed, they are varied by the loss functions, being split into two categories: *pair-based* and *proxy-based*.

Pair-based metric learning is trained with the loss derived from the embedding-based distances between pairs of data [8]. The first pair-based loss, named contrastive loss [9], aims to minimize the embedding distance of identical-class input pairs and maximize the distance otherwise. Alternatively, the triplet loss uses three data points (two of the same class and one of the other) and constraints the embedding distance of the samples, i.e. minimizing the distance between the anchor and positive sample and maximizing the distance between the anchor and the negative sample [10]. The model trained with contrastive or triplet loss mostly depends on effective sampling strategies; precisely, the easy pairs (i.e. inter-class with distinctive content or intra-class with similar content) do not help to improve the convergence rates [5]. To overcome the issue, N-pair loss [6] and lifted-structure loss [7] are proposed to consider the hard pairs into the training process. N-pair loss picks out one positive sample from N-1 negative samples, while lifted-structure loss picks out one positive sample with all negative samples in a training batch. Although the n-pair and lifted structure losses involve the hardness of data, they do not reveal the entire data-to-data relationship [11]. Pair-based methods generally have high training complexity as the input always includes multiple data pairs, subsequently resulting in slow convergence [12].



Proxy Anchor Adaptive Proxy Anchor

Fig. 1. The comparison of decision margin between PA [12], and the proposed APA for two classes. C_1 is a hard class and C_2 is an easy class. The dotted line represents the actual decision boundary between the two classes. The areas between the dashed lines and dotted lines are decision margins.

Proxy-based metric learning introduces the proxies representing groups of same-class data from the training set. The proxy-based loss is derived from the proxy-data pairs instead of data-data pairs, significantly reducing the number of input pairs during the training [5]. In other words, the proxy-based approach addresses the training complexity issue of the pair-based approach. The first proxy-based loss, named ProxyNCA, builds the proxies using neighborhood component analysis [13]. ProxyNCA pulls input samples with their respective class proxies together and pushes them apart otherwise. SoftTriple loss uses multiple proxies for a class instead of only one in ProxyNCA; providing more flexibility for modeling intra-class variance in real-world datasets [14].

[⋆] These authors contributed equally.

An extension of ProxyNCA is ProxyNCA++, it renovates the components of ProxyNCA [15]. ProxyNCA is insensitive to noisy data and is potential to enable faster training convergence. However, it does not exploit data-to-data relations since it associates each data point only with proxies. Proxy anchor (PA) loss is proposed to handle entire data in the batch and associate them with each proxy by their relative hardness in data-to-data relations [12]. PA achieves state of the art on several datasets [12].

The performance of PA depends on the selection of hyperparameters, such as the margin and scaling parameters. In practice, the hyperparameter selection is often done via grid search or optimization algorithm, e.g. Tree of Parzen Estimator algorithm [16, 17]. Both methods are very time-consuming and expertise-required. Furthermore, the effects of the margin value are not thoroughly mentioned, implying a certain number of trials and several training tricks to be conducted for the best performance. Besides, whether equal or different margin among classes is good for proxy-anchor metric learning has not been solved. There is a need to investigate the margin setting of classes in metric learning.

In this paper, we propose to learn the margin in the proxy anchor loss instead of fixing it. The proposed method is called adaptive proxy anchor (APA). The contributions of our study are highlighted as follows:

1. The proposed APA treats the margin as the learnable parameter (Fig. 1). Thus, APA does not require many trials or expertise to select an optimal margin value while achieving state-of-the-art results on three public datasets with a faster convergence rate.

2. We conduct extensive ablation experiments to provide insights into several configurations of APA: (i) in proxy anchor learning, there is no need to set the individual margin for every class; (ii) APA is insensitive to the additional hyperparameter, demonstrating the hyperparameter-free of the proposed method.

The organization of the paper is as follows. Section 2 describes APA and its advancements compared to Proxy Anchor. Section 3 reports the results of APA compared with other state-of-the-art (SOTA) methods on four public datasets and analyzes the effects of different settings. Section 4 concludes our paper.

2. PROPOSED METHOD

The proxy anchor metric learning assigns a proxy as an anchor to represent a class and associates the proxy with all data points in a batch [12]. This mechanism allows the sample to interact with each other via proxies during training. And the fine-grained data-to-data relation is actively considered, which is combined with margin leading to intra-class compactness and inter-class separability.

Let $\mathbf{X}$ be a batch embedding vectors. Let $\mathbf{P}^+$ denote the proxies of existing classes in the batch (known as positive proxies), and $\mathbf{P}$ denote all proxies in the training set. The PA loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{proxy}}(\mathbf{X}) =&\frac{1}{|\mathbf{P}^+|}\sum_{\mathbf{p}\in\mathbf{P}^+}\log(1+\sum_{\mathbf{x}\in\mathbf{X}_\mathbf{p}^+} e^{-\alpha(s(\mathbf{x},\mathbf{p})-m)})\\ &+\frac{1}{|\mathbf{P}|}\sum_{\mathbf{p}\in\mathbf{P}}\log(1+\sum_{\mathbf{x}\in\mathbf{X}_\mathbf{p}^-} e^{\alpha(s(\mathbf{x},\mathbf{p})+m)}),\end{aligned} \tag{1}$$

where m is the margin and α is the scaling factor. The operator $|.|$ denotes the cardinality of the set and $s(,)$ denotes the cosine similarity (distance) between the two input vectors. $\mathbf{X}_\mathbf{p}^+$ represents the subset of $\mathbf{X}$ which has the same class as proxy $\mathbf{p}$. Similarly, $\mathbf{X}_\mathbf{p}^-$ represents the subset of $\mathbf{X}$ which has the different class as proxy $\mathbf{p}$. Minimizing the loss in Eq. (1) means pulling the representation of the proxy and data points of the same class close together, and pushing the representation of the proxy and data points from different classes far away. Experiments in [12] show that the performance of the model is sensitive to m and the performance is high and stable with any α greater than 16. The choice of an optimal margin could be different for each dataset, making it time-consuming for hyperparameter tuning. Fig. 2 shows the change of margin during training APA. Motivated by margin-selection free, we explore the margin effect and propose a method to automatically adjust the margin during the training. More precisely, for better generalization, we assign a learnable margin for each class. We then introduce an adaptive loss with two components: $\mathcal{L}_{\text{proxy}}$ to guarantee the compactness of intra-class and the separability of inter-class, and $\mathcal{L}_{\text{margin}}$ to control the variance of margins. Let $m_\mathbf{x}$ be the learnable margin of the same class as embedding $\mathbf{x}$. The new proxy anchor loss is now written as:

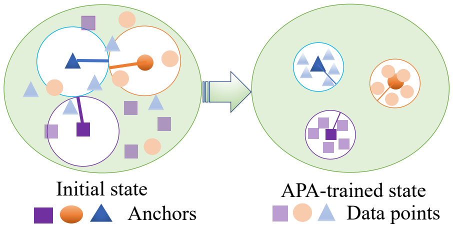


Fig. 2. Visualization of the flowchart of the margin state during training APA.

$$\begin{aligned}\mathcal{L}_{\text{proxy}}(\mathbf{X}) =&\frac{1}{|\mathbf{P}^+|}\sum_{\mathbf{p}\in\mathbf{P}^+}\log(1+\sum_{\mathbf{x}\in\mathbf{X}_\mathbf{p}^+} e^{-\alpha(s(\mathbf{x},\mathbf{p})-m_\mathbf{x})})\\ &+\frac{1}{|\mathbf{P}|}\sum_{\mathbf{p}\in\mathbf{P}}\log(1+\sum_{\mathbf{x}\in\mathbf{X}_\mathbf{p}^-} e^{\alpha(s(\mathbf{x},\mathbf{p})+m_\mathbf{x})}).\end{aligned} \tag{2}$$

This loss has the properties of PA, i.e. leveraging the advantages of fine-grained data relation. In training proxy anchor loss, a large value of margin easily leads to overfitting. To constraint the value of margins, we introduce the margin loss as follows:

$$\mathcal{L}_{\text{margin}} = \frac{1}{C}\sum_{i=1}^{C} m_i, \tag{3}$$

where m_i is the learnable margin value of the i^{th} class and C is the number of classes in the training set. To summarize, we introduce an adaptive anchor loss as:

$$\mathcal{L}_{\text{adaptive}}(\mathbf{X}) = \mathcal{L}_{\text{proxy}}(\mathbf{X}) + \lambda\frac{1}{\mathcal{L}_{\text{margin}}}, \tag{4}$$

where λ is a positive regularization parameter. Therefore, the proposed APA loss enables actively considering the relative data hardness in training and flexibly adjusting the margin to adapt to data distribution.

3. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we first describe the datasets and experimental settings. We then describe the experimental results of the proposed method compared to the state-of-the-art metric learning methods. Finally, we discuss the configurations of the proposed APA method.

3.1. Datasets and Settings

We evaluated the proposed method on four public datasets: Stanford Online Products (SOP) [7], CUB-200-2011 [18], Car196 [19], In-Shop clothes Retrieval (In-shop) [20], in which Table 1 shows the statistics in detail. For all experiments, the images were resized to 256 × 256 and cropped to 224 × 224, α was set to 32 for all experiments as suggested in [12]. Only cropping and flipping were used as augmentations.

Datasets	Train		Test	
	Classes	Images	Classes	Images
CUB-200-2011	100	5,864	100	5,924
CAR196	98	8,054	98	8,131
SOP	11,318	59,551	11,316	60,502
In-Shop	3,997	28,882	3,985	28,760

Table 1. Information of four dataset including the figures of train, test classes and images of each dataset.

For a fair comparison, we reproduced the PA results and conducted the proposed APA experiments using the same training pipeline. The proxies were initialized with normal distribution as suggested in [12]. ResNet-50 [25] was used as the backbone to extract the features. The last fully-connected layer was changed to obtain the dimensionality of embedding vectors and L2-normalized before returning the final output. To evaluate the methods' performance, we used Recall@K in which a higher value indicates a better model. All timing results were collected on a docker container with a single A100 GPU of 40GB RAM.

Datasets	CUB-200-2011	Cars-196	SOP	In-shop
Multi-margin	70.1	88.9	80.0	90.9
Single-margin	70.5	90.3	81.4	91.5

Table 2. Comparison the effect of single-margin and multi-margin to the performance on four public datasets. All the results are reported with R@1 (%) for the value $\lambda = 1$.

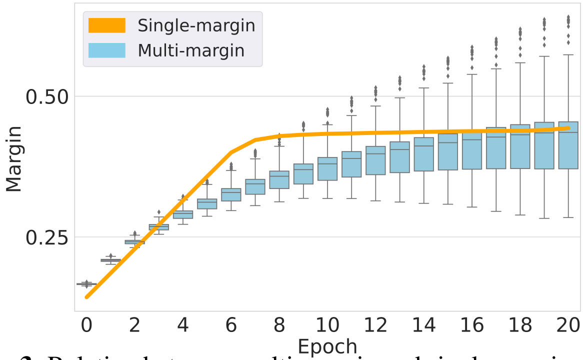


Fig. 3. Relation between multi-margin and single margin during training on In-Shop dataset.

3.2. Ablation Study

From multi-margin to single margin. APA can be configured with single-margin (the same margin value for all classes) or multi-margin (an individual margin for each class). We recorded the class margin values during training on the In-shop dataset and plotted the range of values over training time in Fig. 3. In the multi-margin setting, the margin value range (boxplots) broadens as the training progresses. This phenomenon implies the necessity of choosing a suitable class margin value for optimal performance in each dataset. In the single-margin setting, the margin (orange-solid line) converges to the mean margin value of the multi-margin setting, indicating a correlation between the two settings.

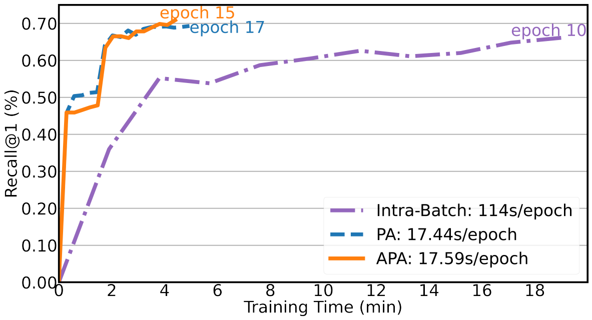


Fig. 4. Convergence speed of Intra-batch, PA, and APA on the CUB-200-2011 dataset.

We benchmarked the two settings on four datasets (see Table 2). The results show that the single-margin setting is marginally better than the multi-margin. Having a specific margin for each class in the training distribution leads to an overfit. Because the classes in the test set are unseen during training, multi-margin settings hamper the generalization of the algorithm on the test dataset. We then recommend using single margin settings in the proxy-anchor metric learning to avoid this effect and attain better performance.

		In-shop		
R@K(%)		1	10	20
Multi-Similarity [11]	BN	89.7	97.9	98.5
NormSoftmax [21]	R50	89.4	97.8	98.7
Cross-Entropy [22]	R50	90.6	98.0	98.6
EPSHN [23]	R50	87.8	95.7	96.8
Proxy Anchor [12]	R50	**91.5**	97.5	98.2
ProxyNCA++ [15]	R50	90.4	**98.1**	**98.8**
Intra-Batch [24]	R50	**92.8**	**98.5**	**99.1**
Proposed APA	R50	**91.5**	97.5	98.3

Table 4. Comparison to other methods on In-Shop clothes Retrieval dataset. The top-2 performances are highlighted in **blue** and **red**. All methods are compared with the size of 512-embedding, BN: Inception with batch normalization, R50: ResNet-50. Best viewed in color.

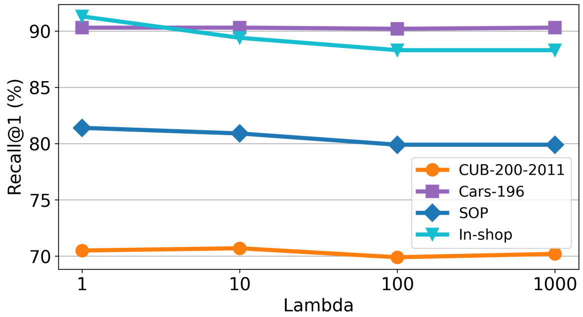


Fig. 5. Recall@1 versus λ values on the four datasets.

Effect of scaling λ factor: We examined the effect of the scaling parameter λ by conducting experiments with different values of $\lambda \in \{1, 10, 100, 1000\}$ on the four datasets. The experimental results in Fig. 5 shows that there is a negligible difference in the range of the scaling parameter λ from 1 to 1000, making it unnecessary to tune this hyperparameter. We recommend to set λ to 1.

3.3. Comparison to Other Methods

We compared the proposed APA to seven existing metric learning methods on four datasets: CUB-200-2011, Cars-196, SOP, and In-shop. Tables 3 and 4 summarize the results of the methods on the four datasets. Note that we conducted the experiments of PA and APA with ResNet-50, and with an embedding dimension of 512 for a fair comparison. The results of the other methods are based on published numbers. As shown in Tables 3 and 4, the proposed adaptive method improves the performance of PA without the need of choosing an optimal margin in many trials. Note that the best results for PA over several margins are reported. Furthermore, the proposed APA method outperforms state of the arts approaches in almost all settings, with the highest R@1 score on CUB-200-2011 at 70.7%, Cars-196 at 90.3%, and SOP at 81.4%. APA achieves a slightly lower recall than Intra-batch [24], 91.5% compared to 92.8%. The experimental results demonstrate the convenience of using the adaptive margins and the superiority of the proposed APA method.

In addition, we evaluated the training complexity, which plays a crucial role in deep learning, by recording the convergence speed of the proposed APA, AP [12], and Intra-Batch [24], shown in Fig. 4. The results show that the proxy-anchor approach proves effective and efficient during training, and our loss steadily maintains this training convergence property.

4. CONCLUSION

In this paper, we extend the proxy anchor metric learning method to reduce the requirement of expertise and time for selecting the best margin value. The proposed method, called adaptive proxy anchor (APA), adaptively adjusts the margin for corresponding domains during training. APA achieves state of the art on three public datasets, i.e. CUB-200-2011, Cars-196, and SOP, while maintaining the fast convergence rate compared to the proxy anchor method. We also study the margin behavior to highlight the effect of the optimal value. We have not considered the optimal sampling method for our approach and leave it for future scope of work.

		CUB-200-2011				***Cars-196***				***SOP***			
R@K(%)		1	2	4	8	1	2	4	8	1	10	100	1000
Multi-Similarity [11]	BN	65.7	77.0	86.6	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
NormSoftmax [21]	R50	65.3	76.7	85.4	91.8	**89.3**	**94.1**	96.4	98	79.5	91.5	**96.7**	-
Cross-Entropy [22]	R50	69.2	79.2	86.9	91.6	**89.3**	93.9	**96.6**	**98.4**	**81.1**	91.7	96.3	**98.8**
EPSHN [23]	R50	64.9	75.3	83.5	-	82.7	89.3	93.0	-	78.3	90.7	96.3	-
Proxy Anchor [12]	R50	70.2	79.7	87.0	**92.0**	89.2	93.8	96.0	97.8	80.5	91.4	96.4	98.7
ProxyNCA++ [15]	R50	69.0	**79.8**	**87.3**	**92.7**	86.5	92.5	95.7	97.7	80.7	**92.0**	**96.7**	**98.9**
Intra-Batch [24]	R50	**70.3**	**80.3**	**87.6**	**92.7**	88.1	93.3	96.2	**98.2**	**81.4**	91.3	95.9	-
Proposed APA	R50	**70.5**	79.6	87.2	**92.2**	**90.3**	**94.4**	**96.8**	98.0	**81.4**	**92.1**	**96.8**	**98.8**

Table 3. Comparison to other methods on CUB-200-2011, Cars-196, Stanford Online Products datasets. The top-2 performances are highlighted in **red** and **blue**. All methods are compared with the size of 512-embedding, except ProxyNCA with 64-embedding. BN: Inception with batch normalization, R50: ResNet-50. Best viewed in color.

References

[1] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: deep hypersphere embedding for face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 212–220.

[2] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, p. 926–930, 2018.

[3] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: large margin cosine loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[5] Y. Movshovitz-Attias, A. Toshev, T. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proc. IEEE International Conference on Computer Vision*, 2017, pp. 360–368.

[6] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016, pp. 1857–1865.

[7] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4004–4012.

[8] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 669–688, 1993.

[9] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 539–546.

[10] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-based Pattern Recognition*, 2015, pp. 84–92.

[11] X. Wang, X. Han, W. Huang, D. Dong, and M. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5022–5030.

[12] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3238–3247.

[13] J. Goldberger, G. Hinton, S. Roweis, and R. Salakhutdinov, "Neighbourhood components analysis," *Advances in Neural Information Processing Systems*, 2004.

[14] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "SoftTriple loss: deep metric learning without triplet sampling," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6450–6458.

[15] E. Teh, T. DeVries, and G. Taylor, "ProxyNCA++: revisiting and revitalizing proxy neighborhood component analysis," in *Proc. Euroupean Conference Computer Vision*, 2020, pp. 448–464.

[16] J. Bergstra, R. Bardenet, Y. Bengio, and Kegl B., "Algorithms for hyperparameter optimization," in *Proc. International Conference on Neural Information Processing Systems*, 2011, p. 2546–2554.

[17] J. Bergstra, D. Yamins, and D.D. Cox, "Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. International Conference on Machine Learning*, 2013, p. I–115–I–123.

[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[20] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1096–1104.

[21] A. Zhai and H. Wu, "Classification is a strong baseline for deep metric learning," *arXiv preprint arXiv:1811.12649*, 2018.

[22] M. Boudiaf, J. Rony, I. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. Ayed, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *European Conference on Computer Vision*, 2020, pp. 548–564.

[23] H. Xuan, A. Stylianou, and R. Pless, "Improved embeddings with easy positive triplet mining," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2474–2482.

[24] J. Seidenschwarz, I. Elezi, and L. Leal-Taixé, "Learning intra-batch connections for deep metric learning," *arXiv preprint arXiv:2102.07753*, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.