Second-Order Estimation Based Attention Network for Metric Learning

Zeyu Sun
Graduate School of Information
Production and Systems
Waseda University
Telephone: +81-80-3993-1551
Email: ikawajun@akane.waseda.jp

Sei-ichiro Kamata
Graduate School of Information
Production and Systems
Waseda University
Telephone: +81-93-692-5219
Email: kam@waseda.jp

***Abstract*—Mapping image data into the embedding space where objects of the same class or label have a short distance in-between and objects of different classes have long margins, has been an essential task for many computer vision applications. However, current approaches struggle to map image data into a proper embedding space due to the difficulty of constructing discriminative features from among numerous features from the original data. Existing approaches include finding effective loss and new sampling methods, which do not consider improving the embedding space by selecting fine features extracted by the network.**

In this work, we proposed a new attention approach by exploiting the variance of features. The method can improve the performance of the current metric learning. Our approach consists of a variance estimation module(VEM) and fusion stage for applying channel-wise attention on extracted features. It is easy to implement and fast for training. Unlike other traditional second-order based methods, the variance estimation module does not embed second-order calculation in the network itself, and cost no large extra computation time in the evaluation stage. The experiment shows promising performance while compared with current SOTA approaches on multiple metric learning benchmark datasets such as CUB200-2011, CARS196, In-shop Clothes.

Contribution—We design a new attention module by using estimation of variance in the features and achieve SOTA results in several benchmarks with almost no extra time cost in the test stage.

***Index Terms*—Attention, Second-order Statistics**

I. Introduction

Metric learning is aimed to represent similarity between data points of different cluster by distance. Recently deep learning method is applied to metric learning and achieved good performances. Deep metric learning methods is fundamental to many different applications including image retrieval [1–5], person re-identification [6–8], fine-grained retrieval [9], clustering [10], zero-shot-learning [11–13], geo-localization [14–16], and near duplicate detection [17].

Deep metric learning is trained with data points of different groups, and using optimization method on several loss functions based on pairs of data points [18] of different classes, triplets of images [19]. The key target is tp map points with the same class to a local area and separate different classes. The mapping space can show the semantic relationship of the data points.

Current deep metric learning approaches aim to construct embedding space only by features extracted by CNN(convolutional neurual network). However, recent works on CNN also show that second-order statistics [20,21] play an essential part in features. The second-order statistics in features can serve directly as new features, or as weighting for features. Also, the number of dimension of embedding space is usually a hyperparameter which has a large influence on the result. If the dimension number is too low, the metric learning might be difficult to train. On the other hand, setting a big dimension number could give a good result but prone to an overfitting problem. Finding a proper dimension number is an exhausting problem in deep metric learning. The features extracted by CNN can be treated as original high dimensions embedding space. The common FC layer which follows CNN is then a compression mapping for a low dimensions separation. Such problems appeal to an attention method for finding discriminative dimensions in CNN features.

For the above problems, we proposed a second-order estimation based attention network as a solution. We design a self-supervised variance estimation module(VEM) for predicting variance channel-wisely for each feature. The predicted value of the variance estimation module is compared with the actual variance of the features and compute a loss. This loss is further added to the metric learning loss as a final loss. Then the parameters of the module are updated every step by decreasing the total loss. The predicted variance is used as a weight for attention in the extracted features. These weighted features further serve as useful features for deep metric learning mapping. The proposed variance estimation module is different from other second-order based methods that directly embed second-order calculation in the network and achieves the goal by define a new and concise loss, and thus introduces very little extra computation cost to the whole network.

This approach has a concise structure and can be placed after any FC layer in the network structure. It also adds a few parameters into the network and has little time cost in the whole module. In the experiment, we use current SOTA loss Margin loss [22] and distance sampling, and show that an obvious boost in the result of several benchmark datasets

978-1-7281-9331-1/20/$31.00

CUB200-2011 [23], CARS196 [24], and In-shop Clothes [25]. In the paper, we will explain our work by introducing the related works, giving definition of the proposed approach, showing the experimental results and conclusion.

Fig. 1. visualization of embedding space by t-SNE [26] on dataset CUB200-2011 [23] by our work. We magnify some part of the image and show that the pictures of same classes of birds are clustered together in the new space.

II. Related Work

A. Metric Learning

Recently, metric learning has been applied to multiple areas such as zero or one-shot learning [11–13], image retrieval [1–5], and clustering [10]. Deep metric learning includes a deep neural network for feature extraction, mapping methods such as fully connection layer for transforming features into embedding area, corresponding loss, sampling strategy, evaluation metric, and optimization method to adapt parameters in the network.

Recently large effort has been devoted to constructing new loss functions. Several proposed losses are Contrastive loss [27,28], Triplet loss [19], N-pair loss [18], Margin based loss [22] , Quadruplet loss [29]. Contrastive loss [27,28] is the most simple one and mapping the data points that belong to different groups into several fixed points with a fixed margin. The Triplet loss [19] aims to make the distance between negative pairs bigger than positive pairs, while N-pair loss [18] further considering multiple negative pairs for one positive pair. The quadruplet loss [29] uses part of pairs as triplet loss [19] and aims at decreasing in-class variances and increase intra-class variance.

Others have developed new sampling methods. For instance, semi-hard sampling [30–32] constructs N positive pairs and randomly samples one negative pair for each positive pair. Distance weighted sampling [22] sets different sampling possibility for negative pairs considering the distance of them from the anchor. Nevertheless our proposed work focuses on different parts in metric learning compared with the above strategies. We design an attention module for the selection of useful features extracted by the network and achieve better performance on the final embedding space mapping.

B. Second-order CNN

Using higher-order statistics for exploiting the discriminative image features has been a hot topic recently. The global second-order pooling [20] is used in recent CV tasks. For example, image classification and object detection. Another example is the bilinear CNN [21] which uses second-order information of features and achieves SOTA performance in the fine-grained visual tasks. In the work of single image super-resolution [33], Tao et al. proposed to embed a second-order pooling module into the network and extract the relationship between features and then apply channel-wise attention mechanism. These works all add covariance calculation or matrix multiplication of features into the original first-order network, and have large extra computation cost. However, our proposed method does not introduce any time-consuming second-order calculation like other works in the evaluation process, but only extra time cost in the training process. The proposed model itself does not contain any second-order module like other works mentioned above but obtain variance information by an auxiliary loss.

C. Attention Network

In the convolution network, soft attention strategy is usually applied for end-to-end training. The widely-used works include spatial transformer module [34] for house number recognition, scale attention [35] for automatic scale selection and squeeze-and-excitation networks [36] for selection of useful feature channels. In our work, we use the predicted variance for attention to find discriminative features extracted by the convolution network.

III. PROPOSED APPROACH

The brief pipeline is: First the CNN module extracts features from images. Then the VEM calculates the difference between its predicted variance and the given features and gives the variance prediction loss. Next, the predicted variance is fuzed with extracted features by a sigmoid function. These weighted features are finally fed to the fully connection layer to map to the target embedding space and get corresponding metric learning loss. The addition of variance prediction loss and metric learning loss is the total loss for optimization.

A. Preliminaries

The total training dataset is denoted as $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$, where $\mathcal{X}$ denotes the image space. The image classes are denoted as $Y = \{y_1, \ldots, x_m\}$, the total label representation space is denoted as $\mathcal{Y}$. The network learns a maping from the dataset into a m-dimension space. The mapping is denoted by $f : \mathcal{X} \to \mathcal{Y}$.

The distance metric is defined as:

$$D(x_i, x_j) = d(\theta; y_i, y_j) = d(\theta; f(x_i), f(x_j)), \qquad (1)$$

where $d(\cdot)$ represents calculating the Euclidean distance and θ is the parameter of the network. To learn the distance metric,

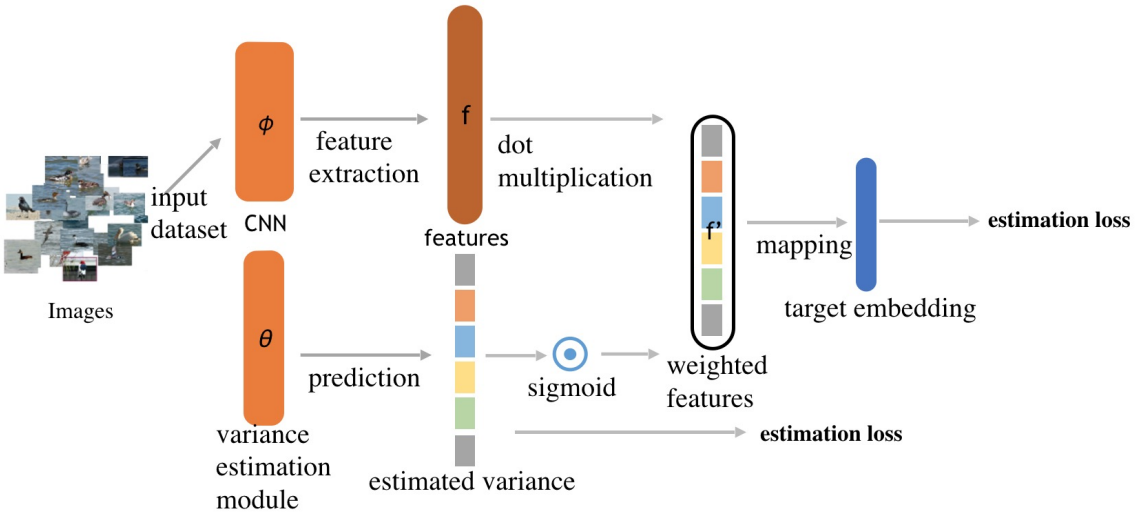

Fig. 2. **The pipeline of our approach.** We design a variance estimation module and corresponding loss for a second order attention mechanism in features. The method involves an estimation loss to update parameters in VEM. In the training process, the model adjusts its prediction on variances for each channel. Then, in the test process, the model uses the prediction for attention on each feature. The method, unlike other models embedded second-order modules, does not have extra time cost in the test(or application) process for second-order calculations.

we use margin based loss [22] as metric learning loss for our experiment. The margin based loss [22] is defined as:

$$l_{margin} = (\alpha + y_{ij}(D_{ij} - \beta))_+, \tag{2}$$

where $(\cdot)_+$ represents the positive part, α determines the margin of separation, β is the parameter to define the boundary between positive and negative pairs, $D_{ij} = D(x_i, x_j)$ and $y_{ij} \in \{-1, 1\}$ [22] . In the rest part we denote the metric learning loss as $loss_{metric}$.

B. Variance estimation

To involve second-order statistics into the network, we embed a VEM in the model. Variance estimation is the key part of our proposed method. Suppose that features extracted by CNN is a $N * W * H$ matrix. The N denotes the batch size, W is width and H represents height of origin feature map. The feature matrix is further flattened into N vector $\vec{v}_i$ of length m ($m = W * H$). This vector can be denoted as $\vec{v}_i = \langle v_1, v_2, \ldots, v_m \rangle$, the feature vector on the whole batch is represented by $\phi = \langle \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N \rangle$.

We denote that variance of one batch is $A = \langle a_1, a_2, \ldots, a_m \rangle$. The variance that VEM predicts is $\bar{A} = \langle \bar{a}_1, \bar{a}_2, \ldots, \bar{a}_m \rangle$. Then the predicting loss is calculated as mean-square error:

$$loss_{var} = \frac{1}{m} \sum_{i=1}^{m} (a_i - \bar{a}_i)^2, \tag{3}$$

The total loss $\mathcal{L}$ is defined as:

$$\mathcal{L}^{total} = \alpha * loss_{var} + \beta * loss_{metric}, \tag{4}$$

where α and β are the weights for the two losses. In the experiment, we use $\alpha = 1.0$ and $\beta = 1.0$.

The structure of the VEM is very simple, just a vector of length m. This is the value for the estimated variance that the module gives on every batch. We suppose that every batch is randomly sampled from the original dataset and that feature variance of it is close to the whole dataset. During the training process, the features extracted by CNN are gradually converging to a certain value, and thus the variance of features on the batch is also converging. Then in the optimization process, the predicted value $\bar{A}$ is closing to the real value A.

C. Channel attention

To make use of the information predicted by the VEM, we utilize attention strategy by multiplication with a channel-wise feature weighting. In order to use traditional end-to-end training, we apply simple gating mechanism by a sigmoid function after the variance prediction. Then the output is used as a weight for the features output by CNN by dot production. In this process, the higher variance of one feature is, the more weight is gained on this feature. The variances imply the discrimination ability of the features. After the weighting, the discriminating features are selected from the old ones. Thus the method increases the discrimination ability of features without increases the dimensions of features. Although it is difficult find exactly best dimension numbers for metric learning, this method can automatically select proper number of features while discarding others by decreasing the weight.

For the output $\bar{A}$ of the VEM, the corresponding weight generated from $\bar{A}$ is $W = \langle w_1, w_2, \ldots, w_m \rangle$. It is a vector of length m and can defined as: $w_i = \sigma(\bar{a}_i)$, where σ represents the sigmoid function: $S(t) = \frac{1}{1+e^{-t}}$.

Therefore, the attention process can be represented as:

$$\vec{s}_i = sigmoid(\bar{A}) \cdot \vec{v}_i, \tag{5}$$

where $\vec{s}_i$ means the weighted feature vector of length m: $\vec{s}_i = \langle s_1, s_2, \ldots, s_m \rangle$. The $sigmoid(\cdot)$ denotes an element-wise sigmoid function on one vector. "·" represents the dot production of the vectors.

Then the weighted feature vectors $\widehat{\phi} = \langle \vec{s}_1, \vec{s}_2, \ldots, \vec{s}_N \rangle$ is the input for the last fully connection layer. And the FC (fully-connected) layer maps it into the target embedding space $\mathcal{Y}$. The main target for training process can be represented as:

$$\underset{\theta_c, \theta_v, \theta_{fc}}{\operatorname{argmin}} \mathcal{L}^{total}, \tag{6}$$

where $\theta_c, \theta_v, \theta_{fc}$ are the parameters of convolution network, VEM and FC layer.

IV. EXPERIMENTS

Firstly, we introduce the datasets for the evaluation. The next part is the hyperparameters setting for the experiment and the loss and optimization method used in the experiment. The last part is the experiment result of our approach and comparison with the current SOTA methods. For the evaluation, we adopt the normalized mutual information score [37] $NMI(\Omega, \mathbb{C}) = \frac{2 \cdot I(\Omega, \mathbb{C})}{H(\Omega) + H(\mathbb{C})}$. The Ω means the true labels for clustering and $\mathbb{C}$ is the set of clusters of predicted points obtained by K-means. I is the mutual information, and H denotes the entropy. The retrieval result is evaluated by the Recall@k metric [38].

A. Datasets

The following datasets are benchmarks for image retrieval and used for evaluation of our approach.

1) CARS196 [24]: This dataset contains 196 classes of cars with total 16183 images. The first 98 classes(8052 images) are used for training the network and then the rest(8131 images) are for evaluating.

2) CUB200-2011 [23]: This dataset consists of 200 different classes of bird species. The first 100 classes(5864 images) of birds are used for training and the other 100 classes(5924 images) for evaluating.

3) In-shop Clothes Retrieval [25]: This dataset has 11,735 classes of clothing items with total 54,642 images. We used 3997 classes of clothing items(25882 images) for the training process and the rest of 3985 classes(28760 images) for evaluating

B. Implementation Details

In the experiment, we test the method by following the implementation details in other metric learning works. We use 128 for embedding space, and batch size 64 for all experiments. The input image is resized to 256 * 256 and then cropped to a size of 224 * 224.

The experiment is based on ResNet-50 [39]. We use a pretrained model on dataset ImageNet. The FC layer and VEM are randomly initialized. We adopt Adam optimizer [40] for all datasets. The epoch is set to 70. The hyperparameter for margin based loss [22] is $\beta = 1.2$, which is similar to other works.

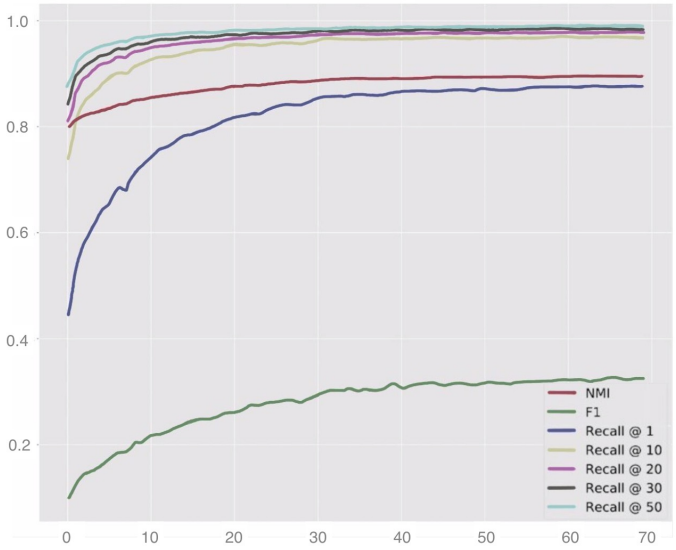

Fig. 3. Evalution by Recall@k and NMI, F1 for 70 epochs in the experiment on In-shop Clothes Retrieval [25].

Fig. 4. Retrieval results on CARS196 [24]. We use 5 car images for query, and retrieve 4 nearest neighbors for each. The results show that images for cars of same type are retrieved.

C. Results

We compare the method second-order attention network with recent SOTA methods. These method including several different loss methods: N-pairs [18], Angular [42], Triplet [19], ProxyNCA [43], Histogram [41]; and other training strategies: Boosting Independent Embedding Robustly(BIER [47]), A-BIER [48] Hard-Aware Deeply Cascaded Embedding(HDC) [49], smartMining [46], HTG [50], and HTL [51]; and also the FashionNet [25] benchmark, deep randomized ensembles(DREML [44]), and divide and conquer [45]. From tables I II III we can see that the proposed method outperforms the existing SOTA methods on 3 datasets with consistent boost in both Recall@k and NMI. For Fig.4 ,Fig.5, Fig.6 we show the query result for nearest neighborhood in image retrieval.

We use only 128 dimensions of embedding space rather than

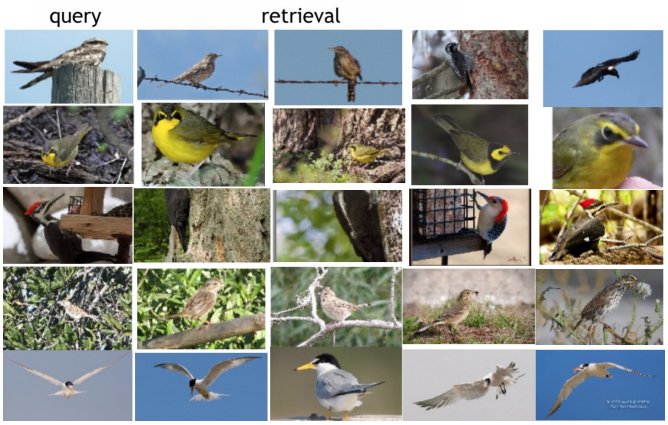

Fig. 5. Retrieval results on CUB200-2011 [23]. We use 5 bird images for query, and retrieve 4 nearest neighbors for each. The results show that images for birds of same species are retrieved.

TABLE I
RECALL@K WITH NMI ON DATASET CUB200-2011 [23]

Method	R@1	R@2	R@4	R@8	NMI
N-pairs [18]	51.0	63.3	74.3	83.2	60.4
Histogram [41]	50.3	61.9	72.6	82.4	-
Angular [42]	54.7	66.3	76.0	83.9	61.1
Triplet [19]	42.6	55.0	66.4	77.2	55.4
ProxyNCA [43]	49.2	61.9	67.9	72.4	64.9
DREML [44]	63.9	75.0	83.1	89.7	67.8
Divide and Conquer [45]	63.9	76.6	84.4	90.6	69.6
proposed(margin)	**66.0**	**77.3**	**85.2**	**91.1**	**69.8**

512 in works like A-BIER [48], BIER [47], HTL [51]. Also, different with works like DREML [44] which adopts a big model that is an ensemble of 48 ResNet18 [39], we merely add parameters into the origin network and a have little extra computation cost for the whole process. The method converges quickly within 70 epochs. For simplicity, we use 70 epochs for all 3 datasets. But as showed in Fig.3, the in-shop clothes retrieval can converges around 50 epochs. And our method has no additional time-consuming process like k-means in divide and conquer the embedding space [45] and other various hard examples mining which consist of calculating difficult data points for separation in the dataset. As for the robustness of the model, we are using a rather small number of dimensions for feature representations. Also, the parameters added to the original CNN are very limited. Therefore the whole model is less prone to the overfitting problem.

V. CONCLUSION

In the paper, we proposed an effective and easy-to-implement approach for deep metric learning. The approach involves a specially designed module for estimation of variance in the features and corresponding attention mechanism. The whole approach is end-to-end trainable and adds very few additional parameters. The experimental results on CARS196 [24], CUB200-2011 [23], and In-shop Clothes [25] outperform the current SOTA methods and demonstrate the effectiveness.

TABLE II
RECALL@K WITH NMI ON DATASET CARS196 [24]

Method	R@1	R@2	R@4	R@8	NMI
N-pairs [18]	71.1	79.7	86.5	91.6	64.0
SmartMining [46]	64.7	76.2	84.2	90.2	-
Angular [42]	71.4	81.4	87.5	92.1	63.2
Triplet [19]	51.5	63.8	73.5	82.4	53.4
ProxyNCA [43]	73.2	82.4	86.4	88.7	64.9
DREML [44]	86.0	91.7	95.0	97.2	76.4
Divide and Conquer [45]	84.6	90.7	94.1	96.5	70.3
proposed(margin)	**86.0**	**89.6**	**95.2**	**96.6**	**71.3**

TABLE III
RECALL@K WITH NMI ON DATASET IN-SHOP CLOTHES RETRIEVAL [25]

method	R@k					NMI
	1	10	20	30	50	
FashionNet [25]	53.0	73.0	76.0	77.0	80.0	-
HDC [49]	62.1	84.9	89.0	91.2	93.1	-
BIER [47]	76.9	92.8	95.2	96.2	97.1	-
HTG [50]	80.9	93.9	95.8	96.6	97.1	-
HTL [51]	80.9	94.3	95.8	97.2	97.8	-
A-BIER [48]	83.1	95.1	96.9	97.5	98.0	-
DREML [44]	78.4	93.7	95.8	96.7	-	-
Divide and Conquer [45]	85.7	95.5	96.9	97.5	98.0	88.6
proposed(margin)	**87.8**	**97.0**	**98.0**	**98.4**	**98.8**	**89.7**

As for the limitation, the method will take more time and memory in the training process for the variance estimation than the traditional CNN networks. However, it costs little extra time in the test or the application, which is different from current more complex models that involves more and more calculation. In the future works, we will further exploit the potential of using higher order statistics of features. Also, rather than just using simple sigmoid gate for attention mechanism, it is possible to adopt network to catch correlation between each variance of feature like SENet.

REFERENCES

[1] K. Sohn, "Improved deep metric learning with multiclass n-pair loss objective," *the Conference on Advances in Neural Information Processing Systems (NIPS)*.
[2] T. K. L. S. I. Yair Movshovitz-Attias, Alexander Toshev and S. Singh, "No fuss distance metric learning using proxies," *IEEE International Conference on Computer Vision (ICCV)*, 2017.
[3] A. J. S. Chao-Yuan Wu, R. Manmatha and P. Krahenbuhl, "Sampling matters in deep embedding learning," *IEEE International Conference on Computer Vision (ICCV)*, 2017.
[4] G. C. I. R. Ben Harwood, Vijay Kumar B G and T. Drummond, "Smart mining for deep metric learning," *IEEE International Conference on Computer Vision (ICCV)*, 2017.
[5] V. R. Hyun Oh Song, Stefanie Jegelka and K. Murphy, "Deep metric learning via facility location," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[6] R. H. S. Chopra and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 539–546, 2005.
[7] L. B. Alexander Hermans and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv:1703.07737*, 2017.
[8] Y. Z. Liming Zhao, Xi Li and J. Wang, "Deeply-learned part-aligned representations for person reidentification," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

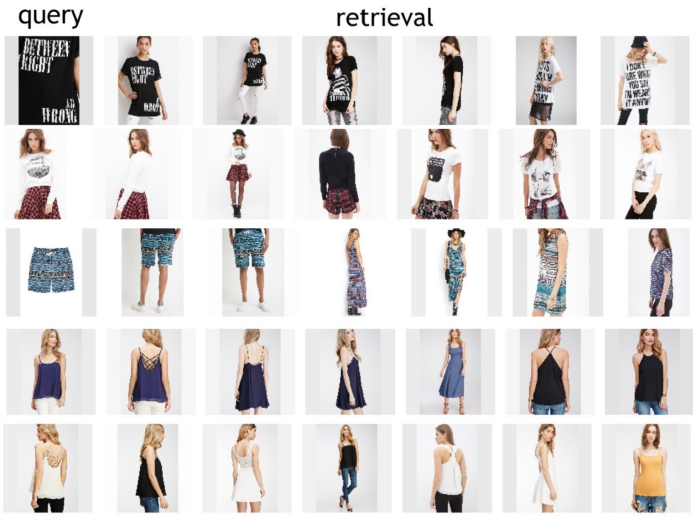

Fig. 6. Retrieval results on In-shop clothes retrieval [25]. We use 5 clothes images for query, and retrieve 6 nearest neighbors for each. The results show that images for clothes of same type are retrieved.

[9] T. K. L. S. I. Y. Movshovitz-Attias, A. Toshev and S. Singh, "No fuss distance metric learning using proxies," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] J. L. R. J. R. Hershey, Z. Chen and S. W. D. clustering, "Discriminative embeddings for segmentation and separation," *ICASSP*, 2016.

[11] B. B. U. Buchler and B. Ommer, "Improving spatiotemporal self-supervision by deep reinforcement learning," *the European Conference on Computer Vision(ECCV)*, pp. 770–786.

[12] E. T. M. A. Bautista, A. Sanakoyeu and B. Ommer, "Cliquecnn: Deep unsupervised exemplar learning," *the Conference on Advances in Neural Information Processing Systems (NIPS)*, pp. 3846–3854, 2016.

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

[14] E. D. Hyo Jin Kim and J.-M. Frahm, "Learned contextual feature reweighting for image geolocalization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3251–3260, 2017.

[15] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," *the European Conference on Computer Vision (ECCV)*, p. 494–509, 2016.

[16] N. J. Nam Vo and J. Hays, "Revisiting im2gps in the deep learning era," *IEEE International Conference on Computer Vision (ICCV)*, p. 2640–2649, 2017.

[17] T. L. S. Zheng, Y. Song and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," *the IEEE conference on computer vision and pattern recognition(CVPR)*, p. 4480–4488, 2016.

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

[19] D. K. F. Schroff and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[20] Q. W. P. L. Zilin Gao, Jiangtao Xie, "Global second-order pooling convolutional networks," *the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

[21] T.-Y. L. A. R. S. Maji, "Bilinear cnn models for fine-grained visual recognition," *arXiv:1504.07889*, 2015.

[22] A. J. S. C.-Y. Wu, R. Manmatha and P. Krahenbuhl, "Sampling matters in deep embedding learning," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] P. W. P. P. C. Wah, S. Branson and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," *Technical Report*, 2011.

[24] J. D. J. Krause, M. Stark and L. Fei-Fei, "3d object representations for fine-grained categorization," *In 4th International IEEE Workshop on 3D Representation and Recognition*, 2013.

[25] P. L. S. Q. X. W. Ziwei Liu and X. Tang, "Deepfashion:powering robust clothes recognition and retrieval with rich annotations," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] L. van der Maaten, "Accelerating t-sne using treebased algorithms," *Journal of Machine Learning Research*, 2014.

[27] S. C. Raia Hadsell and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," *the IEEE conference on computer vision and pattern recognition(CVPR)*, pp. 1735–1742, 2006.

[28] J. L. Junlin Hu and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," *the IEEE conference on computer vision and pattern recognition(CVPR)*, pp. 1875–1882, 2014.

[29] N. T. Marc T Law and M. Cord, "Quadruplet-wise image similarity learning," *IEEE International Conference on Computer Vision (ICCV)*, pp. 249–256, 2013.

[30] Y. A. A. Iscen, G. Tolias and O. C. M. on manifolds, "Metric learning without labels," *the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.

[31] D. D. W. Ge, W. Huang and M. R. Scott, "Deep metric learning with hierarchical triplet loss," *the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.

[32] A. J. S. C.-Y. Wu, R. Manmatha and P. Krahenbuhl, "Sampling matters in deep embedding learning," *International Conference on Computer Vision (ICCV)*, 2017.

[33] Y. Z. S.-T. X. Tao Dai, Jianrui Cai and L. Zhang, "Second-order attention network for single image super-resolution," *e IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11 065–11 074, 2019.

[34] A. Z. M. Jaderberg, K. Simonyan, "Spatial transformer networks," *the Conference on Advances in Neural Information Processing Systems (NIPS)*, 2015.

[35] J. W.-W. X. L.-C. Chen, Y. Yang and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," *arXiv:1511.03339*, 2015.

[36] L. S. J. Hu and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[37] C. D. M. H. Schutze and P. Raghavan, "Introduction to information retrieva," *Cambridge University Press*, vol. 39, 2008.

[38] M. D. H. Jegou and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[39] S. R. K. He, X. Zhang and J. Sun, "Deep residual learning for image recognition," *the IEEE conference on computer vision and pattern recognition*, p. 770–778, 2016.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," *In Advances in Neural Information Processing Systems*, pp. 4170–4178,, 2016.

[42] S. W. X. L. J. Wang, F. Zhou and Y. Lin, "Deep metric learning with angular loss," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2612–2620, 2017.

[43] T. K. L. S. I. Y. Movshovitz-Attias, A. Toshev and S. Singh, "No fuss distance metric learning using proxies," *In Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[44] R. S. H. Xuan and R. Pless, "Deep randomized ensembles for metric learning," *arXiv preprint arXiv:1808.04469*, 2018.

[45] U. B. B. O. Artsiom Sanakoyeu, Vadim Tschernezki, "Divide and conquer the embedding space for metric learning," *arXiv preprint arXiv:1808.04469*, 2018.

[46] G. C. I. R. B. Harwood, V. Kumar and T. Drummond, "Smart mining for deep metric learning," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[47] H. P. M. Opitz, G. Waltner and H. Bischof, "Bierboosting independent embeddings robustly," *In International Conference on Computer Vision (ICCV)*, 2017.

[48] ——, "Deep metric learning with bier: Boosting independent embeddings robustly," *arXiv preprint arXiv:1801.04815*, 2018.

[49] K. Y. Y. Yuan and C. Zhang, "Hard-aware deeply cascaded embedding," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[50] G.-j. Q. H. L. Y. Zhao, Z. Jin and X. s. Hua, "An adversarial approach to hard triplet generation," *the European Conference on Computer Vision (ECCV)*, pp. 501–517, 2018.

[51] D. D. W. Ge, W. Huang and M. R. Scott, "Deep metric learning with hierarchical triplet loss," *the European Conference on Computer Vision (ECCV)*, p. 269–285, 2018.