Deep Bi-directional Cross-triplet Embedding for Cross-Domain Clothing Retrieval

Shuhui Jiang[†], Yue Wu[†], Yun Fu[†‡]
[†]Department of Electrical & Computer Engineering, Northeastern University, Boston, USA
[‡]College of Computer & Information Science, Northeastern University, Boston, USA
{shjiang, yuewu, yunfu}@ece.neu.edu

ABSTRACT

In this paper, we address two practical problems when shopping online: 1) What will I look like when wearing this clothing on the street? 2) How to find the exact same or similar clothing that other people are wearing on the street or in a movie? In this paper, we jointly solve these two problems with one bi-directional shop-to-street street-to-shop clothing retrieval framework. There are three main challenges of cross-domain clothing retrieval task. First is to learn the discrepancy (e.g., background, pose, illumination) between street domain and shop domain clothing. Second, both intra-domain and cross-domain similarity need to be considered during feature embedding. Third, there is large bias between the number of matched and non-matched street and shop pairs. To solve these challenges, in this paper, we propose a deep bi-directional cross-triplet embedding algorithm by extending the start-of-the-art triplet embedding into cross-domain retrieval scenario. Extensive experiments demonstrate the effectiveness of the proposed algorithm.

Keywords

Clothing retrieval; Cross domain; Triplet embedding

1. INTRODUCTION

Studies such as clothing parsing (predicting pixel-wise labelings of garment items) [26, 22], fashion style classification [13, 11, 3], clothing recommendation [15, 9, 8, 21] and clothing retrieval [6, 5, 17, 12, 4] are receiving increasing attentions recently mainly due to the huge clothing market. Using AI technology to facilitate online clothing shopping has gradually become a research hotspot because of a growing e-commerce operation. When shopping online, people may ask two questions. How to find a specific item (e.g., dress, shirts, hats) which is very attractive when other people wearing on the street or in movie? Users may want to search an exact same or similar clothing with a photo captured in unconstrained environment (e.g., pose, illumination, background). Another question is how do I look when wearing this item on? Most online shopping websites only provide the pictures of goods, or better, show how the clothing wearing on skinny model on clean background taken by professional photographers. However, how will ordinary people look when wearing this item on the street?

Existing works mainly focus on street-to-shop scenario [17, 5, 12, 6] by assuming user would manually provide bounding box of the query item, and the retrieval shop items are in clear background with almost frontal view. However, in shop-to-street scenario, we need to search one item in street images with various poses and complex background. How to learn the discrepancy [18] (e.g., background, pose, illumination) between street domain and shop domain clothing remains a challenging problem. In this paper, we address street-to-shop and shop-to-street clothing retrieval with a jointly bi-directional clothing retrieval framework using deep cross-triplet embedding with convolutional network.

Deep feature embedding with convolutional neural networks receives a lot of attentions recently [1, 19, 20, 2, 16, 23]. Bell et al. [1] introduced contrastive embedding [7] for interior design search. FaceNet [19] applied triplet embedding [24] for face verification and recognition by learning deep feature embedding, and achieved state-of-the-art performance. However existing triplet embedding methods do not focus on cross-domain scenarios. Especially, in our cross-domain clothing retrieval task, we have three main challenges. First is to learn the discrepancy (e.g., background, pose, illumination) between street domain and shop domain clothing. Second, intra-domain and cross-domain similarity need to be considered simultaneously during feature embedding. The cross-domain similarity need to be placed in a higher priority due to the large discrepancy. Third, there is large bias between the number of matched and non-matched street and shop pairs. We have very less positive pairs (exact match) and far larger number of negative pairs (not match).

To solve these problems, we propose a deep cross-triplet embedding algorithm by extending the triplet embedding into cross-domain scenario. We fine-tune the AlexNet [14] with the cross-triplet embedding to model the item similarity. For both intra and cross domain, the training process enlarges the difference between points in matched pairs and narrows the difference between points in non-matched pairs. The cross-domain triplet sampling strategy is conducted for solving large unbalance of matched and non-matched pairs. The cross-domain triplets is assigned with higher weights than intra-domain triplets to better solve the discrepancy across domain. The extensive experiments well demonstrate the effectiveness of the proposed framework.

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2967182

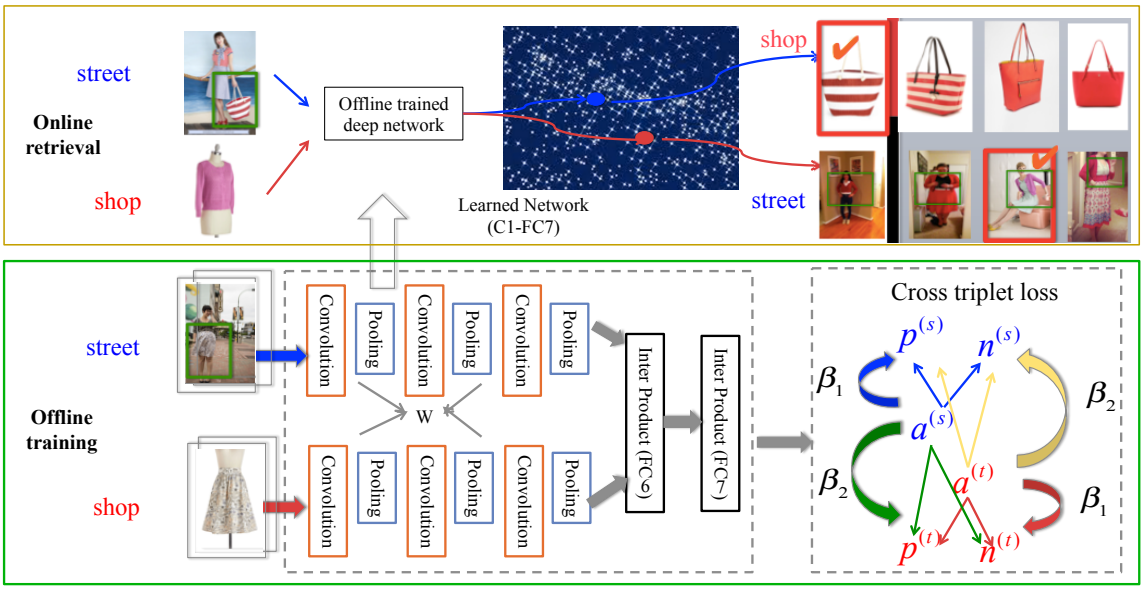


Figure 1: Bi-directional clothing retrieval framework. The inputs of offline training module are street images with bounding boxes and shop images. A cross-triplet embedding is used to train the deep convolutional network. The off-line learned network (C1-FC7) is used to embed the online query from either street or shop domain. In cross-triplet loss, $\{a^{(s)}, p^{(s)}, n^{(s)}\}$ and $\{a^{(t)}, p^{(t)}, n^{(t)}\}$ present the triplets in source and domain and β_1 and β_2 present the weight of intra-domain loss and cross-domain loss.

2. BI-DIRECTIONAL CLOTHING RETRIEVAL

In this section, we first introduce the framework of bi-directional clothing retrieval. Second, we introduce the cross-triplet embedding, and discuss the differences between cross-triplet embedding and existing deep embedding methods.

2.1 Framework

As shown in Figure 1, the framework of bi-directional clothing retrieval contains offline training and online retrieval modules. The offline trained deep network is applied for feature embedding in online retrieval modules.

The inputs of online training consist of both street and shop domain images. For the street domain images, the category of the item, and a bounding box around the item are provided manually by the user or detected by clothing parsing [25]. For the shop domain images, the entire image is used as input due to the large volume.

Although our clothing dataset contains more than 20,000 street images, and 200,000 shop images, it is still far less than the number used for training AlexNet on ImageNet dataset including 1,000 categories [14]. Thus, instead of fully relying on clothing data to newly train a deep network, we keep the pooling layers and convolution layers of AlexNet and fine-tune a three layer fully-connected network with a newly proposed cross-triplet embedding. More specifically, we fine-tune a category-specific model for each category (e.g., hat and pants) by modeling the similarity between a query feature descriptor and cross-domain feature descriptor with cross-triplet embedding.

The fine-tuning is based on the well-known Caffe[10] framework. The learning rate is initially set to 10^{-5} and decreases by factor of 10 every 70 epochs. The model is fine-tuned for about 140 epochs. The batch size is set as 512 images. The other setting of the network is the same with the original AlexNet [14]. The training is conducted on a single GTX Titan X with about 5 GB GPU memory.

In online retrieval, the input could be either a street or shop domain image. The off-line trained network is applied for feature extraction. Then we calculate the cosine similarity to rank the cross-domain items in this category.

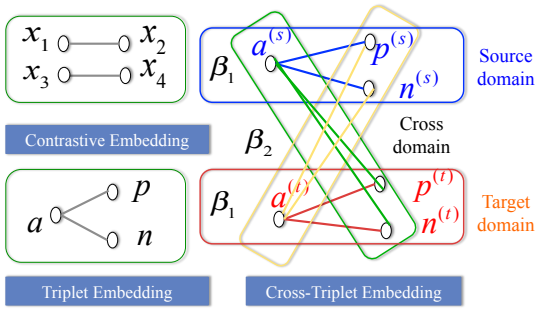


Figure 2: Illustration of contrastive embedding, triplet embedding and cross-triplet embedding. In contrastive embedding, $\{x_1, x_2\}$ and $\{x_3, x_4\}$ are two pairs could be either positive or negative pair. In cross-triplet embedding, similar as in Figure 2, $\{a^{(s)}, p^{(s)}, n^{(s)}\}$ and $\{a^{(t)}, p^{(t)}, n^{(t)}\}$ present the triplets in source and domain and β_1 and β_2 presents the weight of intra-domain loss and cross-domain loss.

2.2 Cross-Triplet Embedding

Triplet embedding [19, 24] is trained on the triplet data $\{(x_a^{(i)}, x_p^{(i)}, x_n^{(i)})\}$ where $(x_a^{(i)}, x_p^{(i)})$ have the same class labels and $(x_a^{(i)}, x_n^{(i)})$ have different class labels. The $x_a^{(i)}$ is denoted as an anchor of a triplet. The training encourages the network to find an embedding where the distance

$D_{ia,in} = ||f(x_i^a) - f(x_i^n)||_2$ between negative pairs is larger than distance $D_{ia,ap} = ||f(x_i^a) - f(x_i^p)||_2$ between matched pairs plus the margin parameter α:

$$J = \frac{3}{2m}\sum_{i}^{m/3}[D_{ia,ip}^2 - D_{ia,in}^2 + \alpha]_+^2. \quad (1)$$

Although the existing triplet embedding methods have achieved satisfactory performance in many fields [19], they do not well consider the cross-domain retrieval scenario. It is easy to think about still using the triplet embedding and neglecting the domain information. It means that we randomly sample a, p and n from two domains. However such strategy may cause some problems. First, existing triplet embedding assigns the same weight for all the triplets when calculating the total triplet loss. Thus, it may treat the intra-domain and cross-domain equally. However, when facing cross-domain scenario, the cross-domain triplet loss needs to be assigned higher weights due to the difficultly of learning large cross-domain discrepancy. Secondly, if we randomly sample a, p and n from two domains, we will have eight different situations including the four conditions shown in cross-triplet embedding in Figure 2, plus $\{(a^{(s)}, p^{(t)}, n^{(s)})\}$, $\{(a^{(s)}, p^{(s)}, n^{(t)})\}$, $\{(a^{(t)}, p^{(s)}, n^{(t)})\}$ and $\{(a^{(t)}, p^{(t)}, n^{(s)})\}$. It is difficult to decide the different margins α and weights for such complex situations. For example, it is hard to judge whether the loss of positive pair (e.g., $D(a^{(s)}, p^{(t)})$) across domain should be larger than the loss of negative pair intra domain (e.g., $D(a^{(s)}, n^{(s)})$)?

To address these concerns, we extend the existing tippet embedding into a cross-domain scenario. As shown in Figure 2, cross-triplet embedding is trained on both source domain and target domain data with domain information. Note that, in our bi-directional framework, in street-to-shop direction, street domain is denoted as source domain and shop domain is denoted as target domain, and vice-versa. The cross-domain triplet loss considers four conditions:

$$J = \underbrace{\beta_1 J^{s,s} + \beta_1 J^{t,t}}_{\text{intra domain}} + \underbrace{\beta_2 J^{s,t} + \beta_2 J^{t,s}}_{\text{cross domain}}, \quad (2)$$

where $J^{s,s}$ denotes the intra-domain lost function of triplet data in source domain $\{a^{(s)}, p^{(s)}, n^{(s)}\}$. $J^{t,t}$ denotes the in target domain $\{a^{(t)}, p^{(t)}, n^{(t)}\}$. $J^{s,t}$ denotes the cross domain triplet $\{a^{(s)}, p^{(t)}, n^{(t)}\}$ and similarly, $J^{t,s}$ denotes the cross domain triplet $\{a^{(t)}, p^{(s)}, n^{(s)}\}$. β_1 is used as the weight for intra-domain loss and β_2 is used as the weight for cross-domain loss. In order to emphasis more on the cross domain triplet, we set $\beta_2/\beta_1 > 1$. Within each condition, the triplet loss is calculation as Eq (1).

As we face the challenge that the number of exact match pairs is far less than the number of negative pairs, we propose a cross-triplet sampling algorithm. For each sample a, we extract the product id as the label and random sample an image with the same product id, either in the same domain or cross domain. Since the number of same exact item is very small, if there is no sample with same product id, we choose the sample itself as the positive pair. We compare the domain information of a and p. If a and p are from same domain we set the weight of this triplet as β_1. If a and p are from different domain, we set the weight of this triplet as β_2. Then we sample a negative sample with the same domain as p but with different label as a.

The objective function could be easily solved by Stochastic Gradient Descent (SGD). Each triplet could be optimized by applying standard triplet embedding optimization method with multiplying corresponding weight β_1 and β_2.

3. EXPERIMENT

3.1 Dataset

We use the Exact Steet2Shop dataset [6][1]. It contains 20,357 street photos and 204,795 shop photos in 11 categories. Street photos are captured in everyday uncontrolled environment (e.g., pose, background). Shop photos are from 25 different online clothing stores. The evaluation dataset is constructed by items appear in both street and shop domain.

3.2 Comparison Methods

AlexNet [14]: AlexNet [14] is one of the most widely used CNN model, which is also applied as baseline in [6]. The activation of the 4096-dimensional fully-connected layer FC6 is used as feature presentation.

Where to buy it (Where)[6]: Where to buy it paves the way of Exact Street to Shop. It learns the similarity between street and shop domain by minimizing the cross-entropy error.

Triplet embedding (Triplet) [19]: In order to show the effectiveness of our cross-triple embedding, we compare the existing cross-triple embedding [19] as shown in Eq (1). In this method, the triplets are sampled randomly from two domains without considering the domain information.

Cross-triple embedding(Ours): This method contains the whole framework of our deep bi-directional embedding with convolutional network.

3.3 Experimental results

The experimental settings are following [6]. For clothing retrieval task, since the category of item is pre-provided either by human label or clothing parsing, the retrieval experiments are conducted within-category. For each category, the exact matching pairs are split into training and test with ratio of 4:1. For street-to-shop task, the query is with bounding box and for shop-to-street task, the whole image is used as query with the category label. Top-k accuracy is to measure the performance.

Table 1 presents the exact matching performance for k=20. The left of Table 1 shows the top 20 accuracy in street-to-shop direction with AlexNet [6], Where [6], Triplet [19] and ours. We copy the results of Where shown in [6]. In the right of Table 1, since no previous methods mainly focused on shop-to-street clothing retrieval, we report the performance implemented using AlexNet [6] and Triplet embedding [19] to solve the shop-to-street retrieval task. We observe that:

(1) In street-to-shop direction, our proposed method performs the best (in bold fond) or the second best (in red color) except outerwear and tops. In street-to-shop direction, in all the categories our method achieves the best performance. Note that in Where [6], a selective search process is conducted for the shop domain images, which do not conducted in AlexNet [6], Triplet [19] and ours. In Where [6], 100 proposals are kept for each shop image. That results in the time complexity is 100 times larger than simply using the shop image itself. Also, in the shop-to-street scenario, it makes

[1]http://www.tamaraberg.com/street2shop.

Table 1: Top 20 accuracy for exact-street-to-shop task and exact shop-to-street retrieval tasks. The best and second best results in each domain is shown in bold fond and red color respectively.

	street-to-shop				shop-to-street		
Category	AlexNet[6]	Where[6]	Triplet[19]	Ours	AlexNet[6]	Triplet[19]	Ours
Bags	23.56	**37.4**	25.12	27.01	56.28	56.42	**59.13**
Belts	4.49	13.5	15.73	**16.85**	70.27	71.21	**72.97**
Dresses	18.10	**37.1**	19.91	21.18	31.26	28.42	**31.42**
Eyewear	3.62	**35.5**	20.41	25.36	62.16	65.21	**70.27**
Footwear	5.06	9.6	10.32	**14.25**	22.88	23.17	**26.36**
Hats	11.63	**38.4**	27.32	29.07	61.19	65.67	**73.13**
Leggings	9.48	**22.1**	11.42	14.51	37.85	39.23	**41.59**
Outerwear	6.31	**21.0**	16.12	15.71	33.23	36.11	**40.28**
Pants	15.38	29.2	30.02	**32.16**	52.24	53.15	**61.94**
Skirts	11.26	**54.6**	11.24	10.32	47.92	48.93	**56.04**
Tops	11.93	**38.1**	13.51	17.42	39.33	40.28	**42.73**

Figure 3: Visualization of bi-directional street-to-shop (upper) and shop-to-street (bottom) retrieval results. The exact correct retrieval result is shown with red frame and checkmark.

less sense to retrieval street images from 100 proposals instead of using the original shop image. Thus, we conduct experiments of the street-to-shop and the shop-to-street under the same condition with the original shop image. From results in the table, we can observe that even without using the proposals, our method can match or exceed the results in Where [6] in some cases. This demonstrates the superiority of our method.

(2) In both two directions, in all the categories our method achieves better performance than Triplet [19], except outerwear and tops in the street-to-shop direction. It indicates that by extending existing triplet embedding into cross-domain scenario, our cross-triplet embedding could better learn the discrepancy between two domains.

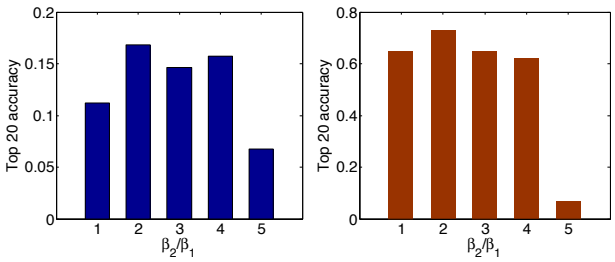
Figure 3 provides two visualization examples.

3.4 Discussion of Cross-triplet Embedding

In this section, we discuss the impact of different ratios of cross-domain weight and intra-domain weight, denoted as β_2/β_1 in Eq.(2). Figure 4 discusses the impact of ratio of $\beta_2/\beta_1 = 1, 2, 3, 4, 5$. We fix $\beta_1 = 1$ and show the top 20 accuracy when $\beta_2 = 1, 2, 3, 4, 5$. We fix the margin $\alpha = 0.5$. We could see that in Figure 4 (a), the performance under $\beta_2/\beta_1 = 2, 3, 4$ is higher than $\beta_2/\beta_1 = 1$. In figure 4 (b), we could see that the performance under $\beta_2/\beta_1 = 2, 3$ is higher than $\beta_2/\beta_1 = 1$. It indicates the effectiveness of assigning a relatively high weight of cross-domain triplet loss to accelerate learning the cross-domain discrepancy.

4. CONCLUSIONS

In this paper, we proposed a cross-triplet embedding with deep convolutional network to jointly solve the bi-directional shop-to-street and street-to-shop clothing retrieval problems. We fine-tuned AlexNet and applied the cross-triplet embedding to model the similarity between cross-domain photos. We extended existing triplet embedding into cross domain-scenario and discussed the ratio of weights in intra-domain loss and cross-domain loss. In the future, we plan to expand the dataset by adding run-way (e.g., New York fashion week) fashion data to provide users both trend-setting and everyday-esthetic styles.



(a) Street2shop (b) Shop2street

Figure 4: Top 20 accuracy of street-to-shop and shop-to-street clothing retrieval of "belts" category, across different ratios of cross-domain weight and intra-domain weight in Eq.(2) denoted as β_2/β_1.

5. ACKNOWLEDGEMENT

The research is supported in part by the MIT Lincoln Labs Grant G00004573, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

6. REFERENCES

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics*, 34(4):98, 2015.

[2] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a âĂIJsiameseâĂİ time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[3] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324, 2015.

[4] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 8–13, 2013.

[5] J. Fu, J. Wang, Z. Li, M. Xu, and H. Lu. Efficient clothing retrieval with semantic-preserving visual phrases. In *Proceedings of the Asian Conference on Computer Vision*, pages 420–431. Springer, 2012.

[6] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3343–3351, 2015.

[7] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE computer society conference on Computer vision and pattern recognition*, volume 2, pages 1735–1742. IEEE, 2006.

[8] T. Iwata, S. Wanatabe, and H. Sawada. Fashion coordinates recommender system using photographs from fashion magazines. In *International Joint Conference on Artificial Intelligence*, volume 22, page 2262. Citeseer, 2011.

[9] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Large scale visual recommendations from street fashion images. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1925–1934. ACM, 2014.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[11] S. Jiang, M. Shao, C. Jia, and Y. Fu. Consensus style centralizing auto-encoder for weak style classification. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI, 2016.

[12] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 105–112. ACM, 2013.

[13] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *Proceedings of the European Conference on Computer Vision*, pages 472–488. Springer, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[15] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 619–628. ACM, 2012.

[16] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-cnn meets knn: quasi-parametric human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1419–1427, 2015.

[17] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3330–3337. IEEE, 2012.

[18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

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

[20] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. *arXiv preprint arXiv:1511.06452*, 2015.

[21] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015.

[22] S. Vittayakorn, K. Yamaguchi, A. C. Berg, and T. L. Berg. Runway to realway: Visual analysis of fashion. In *IEEE Winter Conference on Applications of Computer Vision*, pages 951–958. IEEE, 2015.

[23] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[24] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[25] K. Yamaguchi, M. Kiapour, and T. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3519–3526, 2013.

[26] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577. IEEE, 2012.