Learning Unified Embedding for Apparel Recognition

Yang Song
Google
yangsong@google.com

Yuan Li
Google
liyu@google.com

Bo Wu
Google
bowu@google.com

Chao-Yeh Chen
Google
chaoyeh@google.com

Xiao Zhang
Google
andypassion@google.com

Hartwig Adam
Google
hadam@google.com

Abstract

In apparel recognition, specialized models (e.g. models trained for a particular vertical like dresses) can significantly outperform general models (i.e. models that cover a wide range of verticals). Therefore, deep neural network models are often trained separately for different verticals (e.g. [7]). However, using specialized models for different verticals is not scalable and expensive to deploy. This paper addresses the problem of learning one unified embedding model for multiple object verticals (e.g. all apparel classes) without sacrificing accuracy. The problem is tackled from two aspects: training data and training difficulty. On the training data aspect, we figure out that for a single model trained with triplet loss, there is an accuracy sweet spot in terms of how many verticals are trained together. To ease the training difficulty, a novel learning scheme is proposed by using the output from specialized models as learning targets so that L2 loss can be used instead of triplet loss. This new loss makes the training easier and make it possible for more efficient use of the feature space. The end result is a unified model which can achieve the same retrieval accuracy as a number of separate specialized models, while having the model complexity as one. The effectiveness of our approach is shown in experiments.

1. Introduction

Apparel recognition has received increased attention in vision research ([7, 4, 11, 1, 14, 20]). Given a piece of garment, we want to find the same or similar items. This technology has great potential in assisting online shopping and improving both image search and mobile visual search experience.

Apparel retrieval is a challenging problem. The difficulties are multifold. It is an object instance recognition problem. The appearance of the item changes with light-
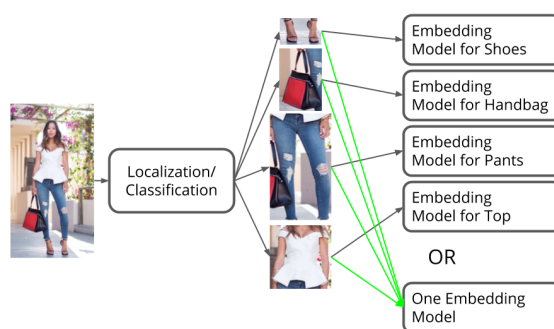

Figure 1. The paper addresses the following question: can a unified embedding model be learned across all the verticals in apparel recognition?

ing, viewpoints, occlusion, and background conditions. For apparels, the images from online shopping sites may differ from those taken in "real life" under uncontrolled conditions (also called street photos [7]). Different verticals (or categories) also have different characteristics. For instance, images from the *dress* vertical may undergo more deformations than those from the *handbags* vertical.

In fine-grained/instance recognition, separate models are often used for different verticals. For example, in [9, 8], separate models are built for birds, dogs, aircrafts, and cars. Similarly, in apparel recognition, separate models are trained for different verticals/domains ([4, 7]). In [4], the embedding models for images from shopping sites and from streets are learned using separate sub-networks. In [7], the network for each vertical (such as *dress, handbags, sunglasses, and pants*) is fine-tuned independently in the final model training. While using separate models can help improve accuracy, it brings extra burden for model storage and deployment. The problem becomes more severe when the models are used on mobile devices. Therefore it is desirable to learn a unified model across different apparel verticals.

This paper studies the problem of learning unified models for apparel recognition. Our goal is to build a unified model which can achieve comparable retrieval accuracy as separate models, with the model complexity no bigger than a single specialized model. As shown in (Figure 1), the clothing item is first detected and localized in the image. An embedding (a vector of floats) is then obtained from the cropped image to represent the item and is used to compare the similarity for retrieval. We focus on the embedding model learning in this paper.

One way to learn the unified model is to combine training data from different verticals. As shown in our experiments (Section 4.3) and in [7], data combination may cause performance degradation. To avoid the performance degradation, we have developed a selective way to do vertical combination. Unfortunately, such "smart" data combination strategies are not enough - we cannot learn one unified model with satisfying accuracy. Is it possible to obtain such a model? Is the limitation intrinsic in model capacity or is it because of the difficulties in model training? Triplet loss is used to learn embedding for individual verticals, which has shown powerful results in embedding learning [21, 13]. However, as noted in [16, 13] and also observed in our experiments, triplet-based learning can be hard due to slow convergence and the nuances in negative sampling strategy. In this work, we seek new approaches to ease the difficulty in triplet training so that a unified model can be learned.

This paper presents a novel way to learn unified embedding models for multiple verticals. There are two stages in model training. The first stage tackles a relatively easier problem - learning embedding models for individual verticals or a small number of combined verticals. In the second stage, the embeddings from the separate models are used as learning target and L2 loss is deployed to train the unified model. The second stage uses the feature mapping learned in the first stage, and combines them into one single model. As shown in Figure 3 and Section 3.2, the learned unified model can make better and broader use of the feature space.

In summary, this paper proposes a two-stage approach to learn a unified model for apparel recognition. The new approach can help alleviate the training difficulty in triplet-based embedding learning, and it can make more efficient use of the feature space. We have also developed ways to combine data from different verticals to reduce the number of models in the first stage. As a result, a unified model is successful learned with comparable accuracy with separate models and with the same model complexity as one model.

The rest of the paper is organized as follows. Section 2 describes how feature embeddings are learned for individual verticals. Our work on how to learn a unified model across verticals is presented in Section 3. Experiments are shown in Section 4, and it concludes at Section 5.

2. Learning Individual Embedding Models

As shown in Figure 1, we adopt a two-step approach in extracting embedding feature vectors for object retrieval. The first step is to localize and classify the apparel item. Since the object class label is known from the first step, specialized embedding models can be used in the second step to compute the similarity feature for retrieval. We describe how we train the specialized embedding models in this section. Unified model learning will be depicted in section 3.

2.1. Localization and Classification

An Inception V2 ([6]) based SSD ([10]) object detector is used. Other object detection architecture and base network combination can also work [5]. This module provides bounding boxes and apparel class labels, i.e., whether it is a handbag or a pair of sunglasses or a dress. Features are then extracted on the cropped image using an embedding model.

2.2. Embedding Training with triplet loss

We use triplet ranking loss [21, 13] to learn feature embeddings for each individual vertical. A triplet includes an anchor image, a positive image, and a negative image. The goal for triplet learning is to produce embeddings so that the positive image gets close to the anchor image while the negative is pushed away from the anchor image in the feature space. The embeddings learned from triplet training are suitable for computing image similarity. Let $t_i = (I_i^a, I_i^p, I_i^n)$ be a triplet, where I_i^a, I_i^p, I_i^n represent the anchor image, positive image and negative image respectively. The learning goal is to minimize the following loss function,

$$
\begin{aligned}
l(I_i^a, I_i^p, I_i^n) = \\
\max\{0, \alpha + D(f(I_i^a), f(I_i^p)) - D(f(I_i^a), f(I_i^n))\}
\end{aligned}
\tag{1}
$$

where α is the margin enforced between the positive and negative pairs, $f(I)$ is the feature embedding for image I, and $D(f_x, f_y)$ is the distance between the two feature embeddings f_x and f_y.

In our applications, the positive image is always of the same product as the anchor image, and the negative image is of another product but in the same vertical. Semi-hard negative mining [13] is used to pick good negative images online to make the training effective.

2.2.1 Network Architecture

Figure 2 shows the network architecture. We use Inception V2 ([6]) as the base network, chosen mainly for efficiency reasons. Any other base network (e.g. [2, 15, 17, 18]) can also be used.
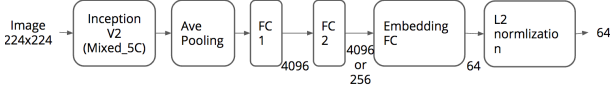
Figure 2. Network architecture for feature extraction. "FC" means fully connected layer, and the numbers are the output dimension of the layer.

3. Learning Unified Embedding

Section 2 shows how embeddings for individual verticals are learned. Given enough training data for each vertical, good performance can be achieved. However, with more verticals in the horizon, having one model per vertical becomes infeasible in real applications. This section describes how a unified model across all verticals is learned.

3.1. Combining Training Data

One natural way to learn a model which can work for multiple verticals is to combine training data from those verticals. With the combined training data, models can be learned in the same way as described in Section 2.

However as shown in our own experiments (Section 4.3) and in [7], training models with combined data may cause accuracy degradation compared to models trained for each individual vertical. To prevent performance degradation, a greedy strategy is developed to decide data from which verticals can be combined. Starting from one vertical, we add data from other verticals in to see if the model learned from the combined data causes accuracy degradation. We keep adding until degradation is observed and keep the previous best combination of verticals. We end up with a number of specialized models, each covering a subset of verticals, while maintaining the best possible accuracy. In our experiments, this results in four specialized models for all apparel verticals.

3.2. Combining Specialized Models

Combining the training data can only somewhat alleviate the coverage scalability issue. Is it possible to learn a unified model with the sample model complexity as one model and no accuracy degradation? Is model capacity the bottleneck or the difficulty in training?

Deep neural networks can be hard to train. The challenge of triplet training has been documented in literature [21, 13, 16]. As exemplified in the Resnet work by He *et al.* in [2], making the training easier can lead to substantial performance improvement. We propose a solution from a similar angle – to ease the difficulty in model training.

We want to learn a unified model such that the embeddings generated from this model is the same as (or very close to) the embeddings generated from separated specialized models. Let $V = \{V_i\}_{i=1}^K$, where each V_i is a set of

verticals whose data can be combined to train an embedding model (Section 3.1). Let $M = \{M_i\}_{i=1}^K$ be a set of embedding models, where each M_i is the model learned for vertical set V_i. Let $I = \{I_j\}_{j=1}^N$ be a set of N training images. If the vertical-of- $I_j \in V_s$, $s = 1 \ldots K$, its corresponding model M_s is used to generate embedding features for image I_j. Let f_{sj} denote the feature embeddings generated from M_s for image I_j. We want to learn a model U, such that the features produced from model U are the same as features produced from separate models. Let f_{uj} denote the feature embeddings generated from model U. The learning goal is to find a model U, which can minimize the following loss function,

$$L = \sum_{j=1}^N \|f_{uj} - f_{sj}\|^2 \qquad (2)$$

Note that features f_{uj} is computed from model U, while f_{sj} may be computed from different models.

The above learning uses L2-loss, instead of triplet loss. L2-loss is easier to train than triplet loss. It is also easier to apply learning techniques such as batch normalization [6] on L2-loss. The above approach allows the use of more unlabeled data. In triplet loss, the product identity (e.g. "Chanel 2.55 classic flap bag") is needed for generating the training triplet. Here only the vertical labels are needed, which can be generated automatically by the localization/classification model.

3.2.1 Visualization

The visualization of the features sheds lights on why our approach works. Figure 3 shows the t-SNE projection (Barnes-Hut-SNE by Maaten [19]) of the features generated from the separate models, i.e, f_{sj}. f_{sj} is 64-d in our experiments. It includes two thousand images from each vertical, and the features are projected down to 2-d space for visualization. From Figure 3 we can see that the feature embeddings f_{sj} are separated across verticals in the space. In other words, the embedding model for each vertical f_{sj} (from model M_s) only uses part of the high dimensional (64-d in our case) space. Therefore one unified model can be learned to combine all of them. This answers our earlier question: the model capacity is not the bottleneck but rather the difficulty in training is.

3.2.2 Relation to the Distillation work

Our work is inspired by the distillation work in [3]. [3] focuses on classification models, and our work is to learn feature embeddings. In [3], an ensemble of models are trained for the *same* task, and then the knowledge in the ensemble is compressed into a single model. In contrast, the separate models M_s in our work are trained for different tasks. As shown in Figure 3, the feature embeddings from different
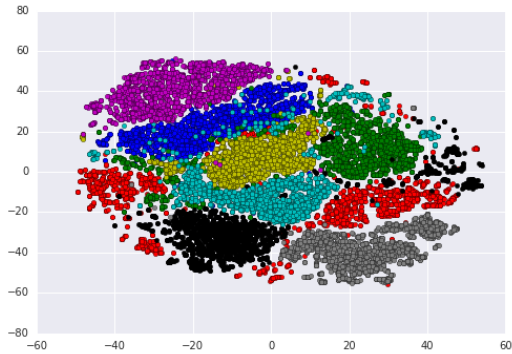
Figure 3. The T-SNE projection for the embeddings. The original embedding are 64-d floats, 2000 samples from each apparel vertical. Blue: dresses; Red: footwear; Green: outerwear; Yellow: pants; Black: handbags; Grey: sunglasses; Cyan: tops; Magenta: skirt.

verticals occupy different areas in the feature space. Our unified model is to consolidate multiple tasks in one model, and to make more efficient use of the feature space.

4. Experiments

4.1. Training Data

Our base network (Figure 2) is initialized from a model pre-trained using ImageNet ([12]) data. For the triplet feature learning (Section 2.2), there are two parts of training data. The training data are first collected from 200,000 search queries using Google Image Search. A text query parser is run to get the apparel class (vertical) label for the text query. The text queries are from the following verticals, *dresses, tops, footwear, handbags, eyewear, outerwear, skirts, and pants*. The search queries are specific product names crawled from online merchants. We take the top 30 images for each search query. The anchor and the positive images for the triplets are from the same search query, and the negatives are from a different search query, but in the same vertical as the anchor. We call these triplets "Image Search triplets". We send a subset of triplets $(20,000$ triplets for each vertical) to human raters and verify the correctness of them. We call this second set of triplets "clean triplets".

For learning the unified embedding model (Section 3.2), we can use the same training data as for triplet feature learning. Since the unified embedding learning only needs the vertical label, which can be obtained via the localizer/classifier (Section 2.1), it is possible to use more training data. However, in our experiments, we didn't observe significant performance improvement when using additional data. Therefore, in the unified model learning, the

same training images are used as those in triplet embedding learning.

4.2. Evaluation Metrics

The retrieval performance is measured by *top-k accuracy*, i.e, the percentage of queries with at least one correct matching item within the first k retrieval results. From the definition of the metric, for the same model, the bigger k is, the higher the *top-k accuracy* is. Data published in [7] are used in evaluation. The unified model is also evaluated on [11] data.

4.3. Effect of Combining Training Data

This section presents our findings on combining different verticals of training data (Section 3.1). To calibrate the performance, triplet loss is first used to learn embeddings for each vertical (Section 2.2). The goal for vertical combination is to use fewer number of models, but without retrieval accuracy degradation. The data from *dresses, tops, footwear, handbags, eyewear, outerwear, skirts, and pants* verticals in [7] are used in evaluation. These verticals are chosen because of their importance for our applications. The embedding models are based on the network depicted in Figure 2, with the $FC2$ output dimension being 4096-d.

The first three rows of Table 1 show the *top-1 accuracy* of (1) models trained individually on each vertical; (2) the model trained with all verticals combined; (3) models trained with the following vertical combination. Training data from *dresses and tops* are combined to train one model; *footwear, handbags and eyewear* are combined to train one model; *skirts and pants* are combined; *outerwear* is trained on its own. This selected vertical combination is obtained by the method described in Section 3.1. From Table 1, models trained with the selected vertical combination give comparable results with individual models. However, the model trained with all verticals combined gives inferior results on some verticals such as eyewear, dresses, tops and outerwear. This shows that it is not trivial to obtain a satisfying unified model by combining all the training data, and we can only combine some verticals to achieve comparable accuracy with the individually trained models.

The above models are trained using "Image Search triplets". To further improve the retrieval performance, "Clean triplets" are used to fine-tune the models. The last two rows of Table 1 shows the *top-1* accuracy comparison results. "No FT" models are trained with Image Search triplets, and the "with FT" models are further fine-tuned using clean triplets. This shows that fine-tuning with the clean data is an effective way to improve retrieval accuracy.

4.4. Effect of Combining Models

By combining the training data, the selected vertical combination results in four separate embedding models. A

Method	bags	eyewear	footwear	dresses	tops	outerwear	pants	skirts
Individual models (no FT)	57.5	53.1	26.6	48.6	24.8	26.7	25.5	37.1
All data combined (no FT)	55.6	35.8	25.1	30.9	18.5	17.4	21.9	30.9
Selected vertical combination (no FT)	56.3	46.2	27.6	48.9	27.6	26.7	24.2	35.2
Selected vertical combination (with FT)	66.9	48.3	35.7	59.1	35.2	29.6	27.6	46.4

Table 1: Comparison of top-1 retrieval accuracy. Individual Model = models trained individually on each vertical; All data combined = the model trained with all verticals combined (one model); Selected vertical combination = models trained with selected vertical combinations, four models in total (see text for details). "FT" means fine-tuning, indicating whether the models are fine-tuned with the clean triplets.

unified model for all the verticals is then learned via what we proposed in Section 3.2. The unified embedding model is also based on the network depicted in Figure 2, with the $FC2$ output dimension being 256-d. Therefore the unified model is even smaller in size than one single separated model.

Figure 4 shows how the *top-1* accuracy changes with training steps (batch size is 32 for each step). Different verticals achieve highest accuracy at different steps. The model at step 3-million is chosen for further experiments.
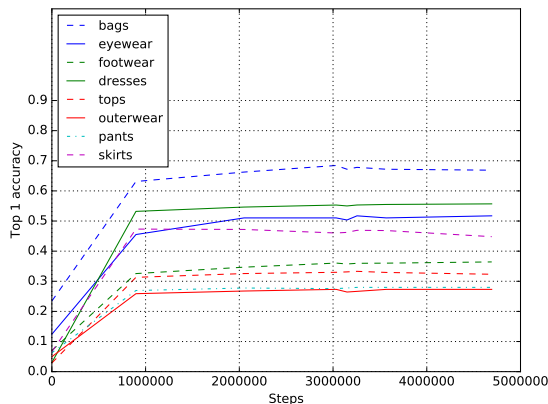

Figure 4. Top-1 accuracy vs. training steps.

Table 2 shows the results of the unified model. The row "WTB paper" represents the best *top-20* accuracy reported in paper [7] (Table 2 in that paper). The rows with "Separate models" are results from the selected vertical combination (Section 4.3). The rows with "Unified Model" are from the one unified model presented in this section using approach in Section 3.2. Note that our models and the models from [7] are trained from different training data.

The "Separate models" provide learning targets for the "Unified Model". In Table 2, the results from the "Unified Model" are very comparable to the results of "Separate models". For some verticals, the "Unified Model" performs even slightly better than "Separate models". We postulate

two explanations. One is the natural variations in the evaluation metric; another reason is that the "Unified Model" can have better generalization performance than the "Separate models" since it is trained on data from all the verticals. Figure 5 shows how the *top-k* accuracy changes with the number of retrieved items (k).
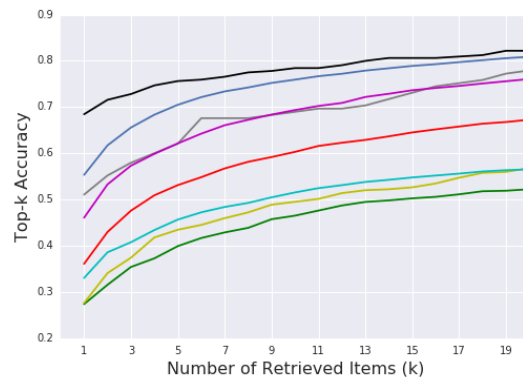

Figure 5. Top-k accuracy vs. the number of item retrieved (k). Black: handbags; Blue: dresses; Grey: eyewear; Magenta: skirts; Red: footwear; Cyan: tops; Yellow: pants; Green: outerwear.

The unified model is also evaluated DeepFashion consumer-to-shop data [11]. Using the ground-truth bounding boxes, our retrieval performance is 13.9% (top-1) and 39.2% (top-20), while it is 7.5% (top-1) and 18.8% (top-20) in Fig 9(b) in [11]. Note that the numbers are not directly comparable as we use the GT bounding boxes. However, it serves the purpose of confirming the quality of our embedding model.

4.4.1 Retrieval Examples

Figures 6 and 7 show some sample retrieval results by using the "Unified Model". Figures 6 shows the successful examples with correct item present in the top-4 returned results. Figures 7 gives some not-so-successful examples with no correct item present in the top-4 returned results. As shown in the figure, even when our model fails to get the correct

Method	bags	eyewear	footwear	dresses	tops	outerwear	pants	skirts
WTB paper [7] (top-20)	37.4	42.0	9.6	37.1	38.1	21.0	29.2	54.6
Unified Model (top-20)	**82.2**	**77.9**	**67.3**	**80.8**	**56.5**	**52.2**	**56.8**	**76.0**
Separate models (top-1)	66.9	48.3	35.7	**59.1**	**35.2**	**29.6**	27.6	**46.4**
Unified Model (top-1)	**68.4**	**51.0**	**36.0**	55.4	33.0	27.3	27.6	46.0
Separate models (top-5)	**76.3**	**64.1**	52.4	**74.6**	**49.2**	**45.9**	43.2	**62.4**
Unified Model (top-5)	75.6	62.1	**53.1**	72.5	47.6	43.9	**43.4**	62.1

Table 2: Comparison of retrieval accuracy. The "top-k" inside the brackets shows which *top-k* accuracy is evaluated. The "Separate models" are trained with the selected vertical combination as in Section 4.3. The "Unified Model" is learned by the approach in Section 3.2.

Figure 6. Retrieval results: successful examples. The images to left of the dashed lines are query images. The items in green bounding boxes are the correct retrieval results.

product, the retrieved items are quite similar to the query.

5. Conclusion

This paper presents our approach and discoveries on how to learn a unified embedding models across all the apparel verticals. We figure out that for a single model trained with triplet loss, there is an accuracy sweet spot in terms of how many verticals are trained together. A novel way is proposed to ease the difficulty in training embeddings for multiple verticals. It uses embeddings from separate specialized models as learning target. The training becomes easier and makes full use of the embedding space. Successful retrieval results are shown on the learned unified model. The unified model has comparable accuracy with separate models and the same model complexity as one individual model. The unified model can make more efficient use of the feature space. Future work includes to extend this work to other fine-grained categories.

References

[1] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. V. Gool. Apparel classification with style. *ACCV*, 2012. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 3

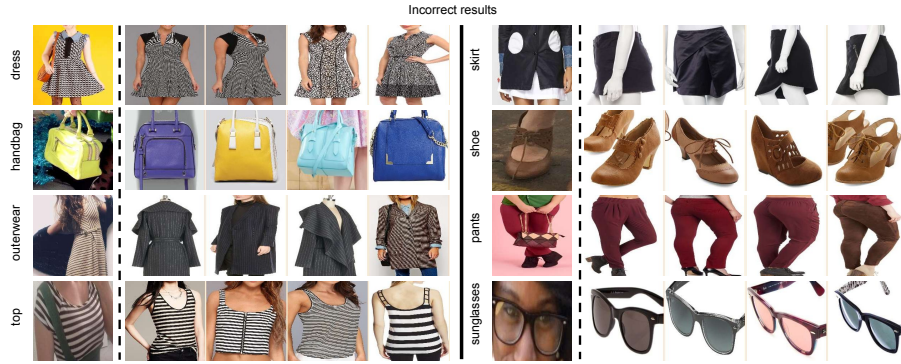[3] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *ArXiv e-prints*, Mar.

Figure 7. Retrieval results: not-so-successful examples (no correct item present in the top-4 returned results).

2015. 3

[4] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *International Conference on Computer Vision*, 2015. 1

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*. 2

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2, 3

[7] M. Kiapour, X. Han, S. Lazebnik, A. Berg, and T. Berg. Where to buy it:matching street clothing photos in online shops. In *International Conference on Computer Vision*, 2015. 1, 2, 3, 4, 5, 6

[8] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*, 2016. 1

[9] T. Lin, A. Roy Chowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *International Conference on Computer Vision*, 2015. 1

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 2

[11] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 4, 5

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 4

[13] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 3

[14] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, , and R. Urtasun. Neuroaesthetics in fashion: Modeling the perception of fashionability. *CVPR*, 2015. 1

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[16] K. Sohn. Improved deep metric learning with multiclass n-pair loss objective. In *NIPS*, 2016. 2, 3

[17] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 2

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 2

[19] L. van der Maaten. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*, 2013. 3

[20] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. *ICCV*, 2015. 1

[21] 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 2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2, 3