Learning Unified Embedding for Apparel Recognition

Yang Song Yuan Li Bo Wu Chao-Yeh Chen Xiao Zhang Hartwig Adam
Google
yangsong@,liyu@,bowu@,chaoyeh@,andypassion@,hadam@google.com

Abstract

In apparel recognition, 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.

1. Introduction

Apparel recognition has received increased attention in vision research ([7, 4, 11, 1, 14, 17]). Given a piece of garment, we want to find the same or similar items. Apparel retrieval is a challenging object instance recognition problem. The appearance of the item changes with lighting, viewpoints, occlusion, and background conditions. Images from online shopping sites may differ from those taken in "real life" (also called street photos [7]). Different verticals also have different characteristics. For instance, images from the *dress* vertical may undergo more deformations than those from the *handbags* vertical.

In fine-grained 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 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.

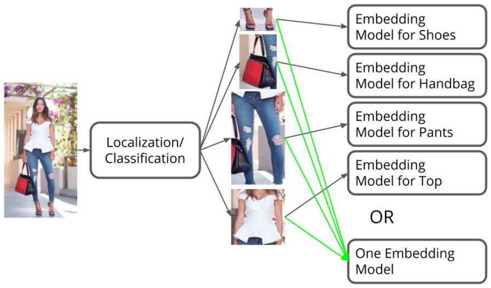

Figure 1. Can a unified embedding model be learned across all the verticals in apparel recognition?

This paper studies the problem of learning unified models for apparel recognition. Our goal is to build a unified model which can achieve comparable 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 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 4.1) 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 [18, 13]. However, as noted in [15, 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 2 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.

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.

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 [18, 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.

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.

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 4.1) 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 [18, 13, 15]. As exemplified in the Resnet work ([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- $I_j \in V_s$, $s = 1 \dots K$, 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,

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

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]. The above approach allows the use of more unlabeled data because the product identity (e.g. "Chanel 2.55 classic flap bag") is needed for generating the training triplet, while here only the vertical labels are needed.

3.2.1 Visualization

Feature visualization sheds lights on why our approach works. Figure 2 shows the t-SNE projection ([16]) of the features generated from the separate models, i.e, f_{sj}. It includes two thousand images from each vertical, and the features are projected down to 2-d space for visualization. From Figure 2 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.

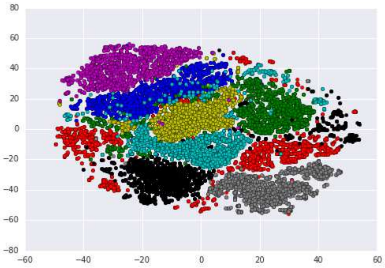

Figure 2. T-SNE projection for embeddings. Blue: dresses; Red: footwear; Green: outerwear; Yellow: pants; Black: handbags; Grey: sunglasses; Cyan: tops; Magenta: skirt.

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

4. Experiments

We use Inception V2 ([6]) as the base network, pretrained with ImageNet ([12]) data. Other base network can also be used. For the triplet feature learning (Section 2.2), the training data are first collected from 200,000 search queries using Google Image Search, taking 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". In the unified model learning, the same training images are used as those in triplet embedding learning.

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.

4.1. Effect of Combining Training Data

This section presents results on combining different verticals of training data (Section 3.1). Triplet loss is used in training (Section 2.2). 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.

The above models are trained using "Image Search triplets". To further improve the retrieval performance,

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. "FT" means fine-tuning, indicating whether the models are fine-tuned with the clean triplets.

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 4.1. The "Unified Model" is learned by the approach in Section 3.2.

"Clean triplets" are used to fine-tune the models. The last two rows of Table 1 shows the *top-1* accuracy comparison results. This shows that fine-tuning with the clean data is an effective way to improve retrieval accuracy.

4.2. Effect of Combining Models

After obtaining separate models according to the selected vertical combination, a unified model for all the verticals is learned via algorithm in Section 3.2. Table 2 shows the results. The row "WTB paper" represents the best *top-20* accuracy in [7] (Table 2). Note that our models and the models from [7] are trained using different data. The rows with "Separate models" are from the selected vertical combination (Section 4.1). The rows with "Unified Model" are from the one unified model (Section 3.2). The results from the "Unified Model" are very comparable to those of "Separate models". Figures 3 shows sample retrieval results.

The unified model is also evaluated on DeepFashion 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 [11]. Note that the numbers are not directly comparable as we use the ground-truth bounding boxes. However, it serves the purpose of confirming the quality of our embedding model.

5. Conclusion

This paper presents our discoveries on how to learn a unified embedding model across all apparel verticals. A novel way is proposed to ease the difficulty in triplet-based embedding training. Embeddings from separate specialized models are used as learning target for the unified model. The training becomes easier and makes full use of the feature space. Successful retrieval results are shown on the learned unified model.

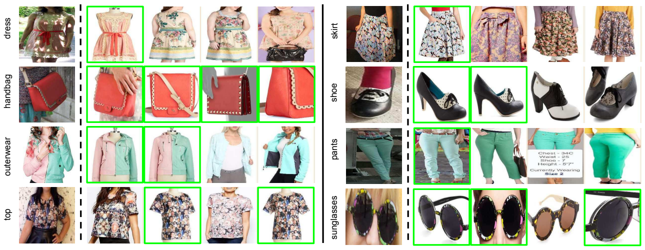

Figure 3. Sample retrieval results from the unified model. The images to left of the dashed lines are query images. The items in green bounding boxes are the correct retrieval results.

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] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *ArXiv e-prints*, Mar. 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 *ICCV*, 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 *ICCV*, 2015. 1, 2, 4

[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 *ICCV*, 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 *CVPR*, June 2016. 1, 4

[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. *IJCV*, 2015. 3

[13] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[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. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016. 2

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

[17] 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

[18] 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 *CVPR*, 2014. 2