# XMU_PAMI at JD AI Fashion Challenge: Fashion-Item Matching

Yu Zhan, Jie Lin, Wan-Lei Zhao
*Fujian Key Laboratory of Sensing and Computing for Smart City*,
School of Information Science and Technology, *Xiamen University*
Xiamen, P. R. China
{yeezytaughtme, linjie037}@stu.xmu.edu.cn, wlzhao@xmu.edu.cn

***Abstract*—XMU_PAMI is a team from Computer Science Department, Xiamen University. The research interests of our team are mainly on content-based image retrieval and instance search. In this paper, the motivations, methodologies, configuration tests and our performance on the JD AI Fashion Challenge are presented. To address the matching-by-retrieving problem, a triplet network based on ResNet-50 is constructed. A novel way of triplet training for fashion-item recognition is proposed. The feature representation for each fashion item is average-pooled from convolutional feature maps. As indicated by our experiments, it exhibits satisfying performance in the fashion-item matching task.**

***Index Terms*—CBIR, instance search, triplet training**

## I. Introduction

JD Fashion-Item Matching Challenge is a task aims to retrieve the relevant e-commerce images from a large fashion-item database when a specific user-shot image is given. Due to the convenience of online shopping, more and more people prefer to buying all types of products online. Recently, extensive research efforts have been devoted to improving the users' shopping experience. One of the desired scenario is that users are allowed to upload product photos that they shot off-line, and search for the same product on the shopping website. Such that users are able to make a comparison and possibly make a transaction. This is where JD Fashion-Item Matching Challenge fits in.

This task is challenging from several perspective of views. First of all, although user-shot photos and the relevant e-commerce images in the database are originated from the same object, they have been produced in different manners. Usually, images in the database are captured in good illumination condition and with clean background. In contrast, photos from the end-user might be captured under various conditions. It is therefore a cross-domain issue when we try to couple these two types of images. In addition, fashion items are often subject to deformation and occlusion in one way or another. In such circumstance, traditional content-based image search approaches based on image global features or keypoint features fail to deliver decent results [1].

Recently, the research trend has moved to deep learning based methods since the AlexNet achieved the best performance in ILSRVC 2012. Deep learning methods, especially the convolutional neural networks (CNN) have been successfully used in various computer vision tasks, such as object detection, face recognition and semantic segmentation. Encouraging results have also been achieved in image search task due to the introduction of CNN. In [8], in order to extract more precise features of instances in an image, the region proposal network (RPN) [7], which is widely used in object detection area, is applied to localize the potential instances. DeepFashion [6] learns more discriminative features through a joint prediction of key points and attributes.

In this paper, a method inspired by FaceNet [9], is proposed to address fashion item search. One of the critical steps in image retrieval is to design discriminative and robust feature representation. CNN has good capability of producing such compact image feature. In our method, a triplet network using triplet loss is trained for fashion-item recognition. The triplet network directly calculates the distance between fashion-item images in Euclidean space, where we try to minimize the distance between e-commerce image and the relevant user-shot image while maximizing the distance between dissimilar fashion-items. ResNet [5] is selected as back-bone for triplet network. The embedding average pooled from convolutional feature maps is used as feature representation for fashion-item matching.

## II. Framework for Fashion Item Search

The original training data provided by the competition organizer are *12K* pairs of fashion images. For every pair of images, fashion item in each image is annotated with bounding-box. The item in one image comes from e-commerce shop and another comes from user-shot. The one from e-commerce has been well pre-processed and in good quality. While the one from user-shot could be under various photolistic conditions. In the following, the one from e-commerce shop is called as "anchor" item, the one from user-shot is called as "positive". The goal of fashion-item matching is to search out the relevant "anchor" item from the database of e-commerce for the given user-shot.

In the image search task, the feature representation plays the key role to the success of a system. On one hand, it is required that the features derived from the same fashion items should be close in the feature space. On the other hand, the distance between dissimilar items should be as large as possible. Since there are hundreds of item types and there are only two positive examples for each type, it is unfeasible to

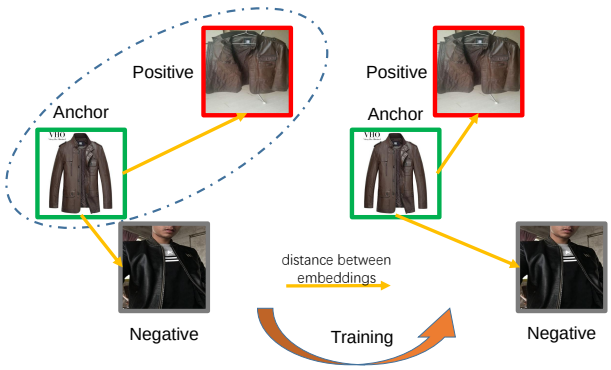


Fig. 1. Instead of directly using training pair (grouped in the blue dashed ellipse), item triplet is created for training to learn the potential relation between e-commerce image and user-shot image.

treat this problem as a visual object classification problem. Alternatively, the feature learning is modeled as a recognition task. In general, our solution follows the idea of FaceNet [9], which has been proposed for face recognition. In the training, item triplets are fed to the network. The triplet is composed by the item pairs provided by the organizer and another randomly selected negative item. So the network is designed to learn the feature representation that maximizes the distances from two positive items to the negative and minimizes the distance between two positive items. This idea is illustrated in Figure 1. In the following, the details of our solution to fashion item search are presented.

### A. Triplet Network

As presented in Figure 2, we adopt ResNet as back-bone network. The fully connected layers are omitted. As pointed out in many lately literatures [2], [3], [10], features derived from convolution layers exhibit superior performance than those derived from fully connected layers for image retrieval task. Thus, the back-bone network is mainly made up of *4* stages of several convolution blocks. The embedding is the output of global average pooling layer stacked on ResNet's last stage, namely the feature maps output by the last stage are average pooled to function as image representation. Notice that all three images forward through only one back-bone network as one triplet. Triplet loss manages to optimize network to produce accurate and discriminative embedding for mini-batch of triplets during back propagation. The details of triplet loss is introduced in the Section II-B.

### B. Triplet for Training

Consider an image $x$ input to the triplet network, the forward propagation is easily regarded as an embedding function $f(x) \in \mathbb{R}^d$. The network outputs a feature vector with dimensionality of $d$. Triplet loss was introduced in [9], which is originally used in producing compact human face representation. As presented in Figure 1, for $i$-th triplet that consists of $x_i^a$ (anchor), $x_i^p$ (positive) and $x_i^n$ (negative), it is desired that $f(x_i^a)$ is closer to $f(x_i^p)$ than $f(x_i^n)$, which is

$$\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha < 0, \forall x_i^a, x_i^p, x_i^n \in \Gamma. \tag{1}$$

In Eqn. 1, $\alpha$ is an enforced distance margin between positive and negative items. $\Gamma$ is the set of all $N$ triplets. Then triplet loss $L$ over $\Gamma$ is given as

$$L = \sum_i^N \max(\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha,\ 0) \tag{2}$$

As shown in Eqn. 2, we try to minimize the distance between positive item and anchor item, while maximizing the distance between anchor item and negative item.

To generate appropriate triplets that contribute to the training, we formulate the following strategies:

- For all images, regions that are outside the bounding-boxes will be cut off, and the regions of interest will be resized to the size of $224 \times 224$ pixels.
- For each training pair, e-commerce image is set as anchor and the corresponding user-shot image plays the role of positive. Another *4* user-shot images are selected from other pairs as negatives to create *4* triplets respectively.
- Two types of triplets are produced for training, namely hard and easy triplets. The triplet is considered as a hard one when the negative image comes from the same category as "anchor" and "positive". The triplet is viewed as easy one as the negative is selected from different categories from "anchor" and "positive". In order to avoid slow convergence during training, the number of easy triplets we use for training is one fourth of the number of hard triplets.

We produce not only one triplet out of image pair for the purpose of scaling up the training data. The reason why easy triplets are selected for training is that we want the embedding of image not only to be item-discriminative but to be category-discriminative as well. Figure 3 presents *3* example triplets for *3* categories.

During the training stage, both ResNet-18 and ResNet-50 are tested as back-bone network respectively. This is to see which is more effective for our task. Due to the high memory demands for deeper ResNet, ResNet goes beyond *50* layers is not tested. The bigger mini-batch size for triplet training also induces big amount of memory consumption. Due to the constraint of hardware setup, the mini-batch size is set to *16* during Stochastic Gradient Descent (SGD). Pre-trained model on the ImageNet [4] for image classification task is used to initialize the back-bone. Initialization with pre-trained model allows the triplet network to be able to classify the images coming from different categories. As revealed later in the Section III, networks using random initialization (without pre-training) performs considerably worse than pre-trained models in the validation test.

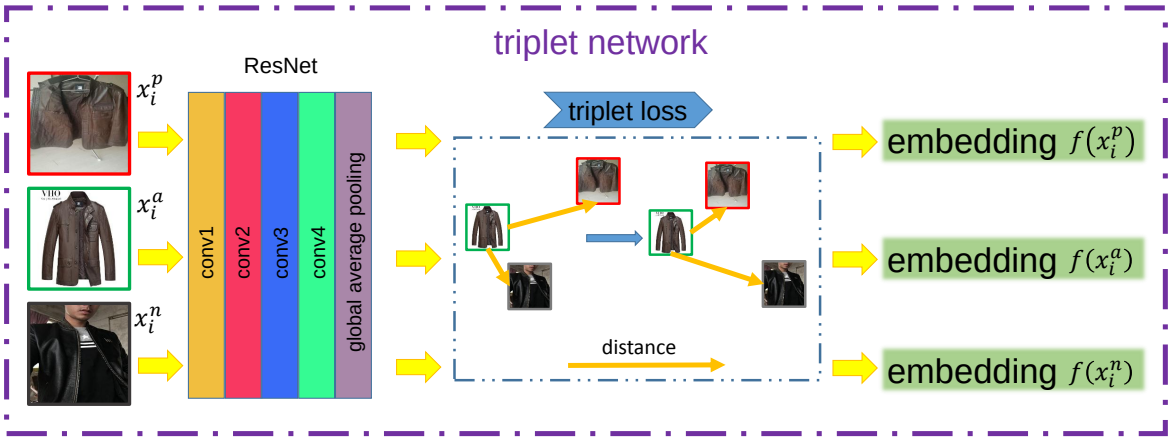


Fig. 2. Framework of the triplet network.


Fig. 3. Three examplar triplets used for training.

## III. Experiments

In this section, the performance of the proposed method for fashion-item recognition is reported. Nearly *110K* triplets are produced for training, which are derived from *12K* image pairs. *20%* of the produced triplets are selected out for validation. The remaining *80%* are used for training.

To kick off training with SGD, a learning rate is set to $10^{-3}$, and the momentum is set to *0.9*. We conduct *4* epochs of training, and at the end of each epoch, a round of validation test is conducted to verify the model's performance. For the third and fourth epoch, the learning rate is decreased to $10^{-4}$ and $10^{-5}$ respectively.

### A. Validation Test over the Proposed Framework

In our first experiment, we investigate the feasibility of triplet training for fashion-item recognition. As presented in Table I, only hard triplets are used for training in order to achieve fast convergence. The performance is evaluated by the accuracy that network assigns the correct user-shot image to the corresponding e-commerce image based on the embeddings. It is observed that models initialized with pre-trained weights achieve better accuracy than those being random initialized. From these results, we believe that only triplets data are insufficient for the network to kick off training from scratch to learn an appropriate representation, compared to the data scale of ImageNet. When the margin $\alpha$ is increased from *1.0* to *4.0*, the recognition accuracy improves for networks with different initialization strategies. It basically indicates that increasing margin $\alpha$ helps to generate more discriminative embedding since it further enforces the distance restriction.

TABLE I
Performance of fashion-item recognition on ResNet-18 with different initialization strategy, margin, and triplet type settings.

| Depth | Initialization | Margin $\alpha$ | Triplets' type | Train accuracy | Validation accuracy |
|---|---|---|---|---|---|
| *18* | random | 1.0 | hard | 94.96% | 89.65% |
| *18* | random | 4.0 | hard | 94.25% | 90.62% |
| *18* | ImageNet [4] | 1.0 | hard | 99.43% | 97.13% |
| *18* | ImageNet [4] | 4.0 | hard | 99.89% | 97.71% |

### B. Network Enhancement & Parameter Tuning

TABLE II
Performance of fashion-item recognition on ResNet-50 with different margin, and triplet type settings.

| Depth | Initialization | Margin $\alpha$ | Triplets' type | Train accuracy | Validation accuracy |
|---|---|---|---|---|---|
| *50* | ImageNet [4] | 1.0 | hard | 99.59% | 97.65% |
| *50* | ImageNet [4] | 2.0 | hard | 99.92% | 97.92% |
| *50* | ImageNet [4] | 4.0 | hard | 99.92% | 98.08% |
| *50* | ImageNet [4] | 8.0 | hard | 99.66% | 97.59% |
| *50* | ImageNet [4] | 4.0 | hard & easy | 99.97% | 98.71% |
| *50* | ImageNet [4] | 8.0 | hard & easy | 99.95% | 98.71% |

After the validation test with ResNet-18 as back-bone, the model is trained with ResNet-50 to boost the performance of our method. Additionally, other hyper-parameters are also fine-tuned for better performance.

Table II summarizes the performance of network with different margin size and triplets' types. Compared to the

Fig. 4. Visualization of top-*10* retrieved similar e-commerce images given with *6* query examples.

performance shown in Table I, deeper ResNet does show slightly better performance. As expected, the larger of the margin size is, the higher is the accuracy that the network achieves. However, when $\alpha$ reaches to *4.0*, simply increasing $\alpha$ no longer lead to higher recognition accuracy. When the easy triplets are joined into the training, the best performance is observed. This indicates that easy triplets help to improve the discriminativeness of the network.

Based on above experiments, the final configuration for our submission is given as follows. ResNet-50 is selected as the back-bone network. The network is pre-trained on ImageNet. The margin size $\alpha$ is set to *4.0*. During the training, both the hard and easy triplets are used.

### C. Retrieval with Embedding

TABLE III
PERFORMANCE OF THE SUBMITTED RUNS.

| Depth | Initialization | Margin $\alpha$ | Triplets set | mAP |
|---|---|---|---|---|
| *50* | ImageNet [4] | 4.0 | train | *0.2127* |
| *50* | ImageNet [4] | 4.0 | train + validation | *0.2735* |

Two runs are submitted for our method. The results are presented in the Table III. The model used for retrieval test is configured with the settings described in the III-B. The first run using the model only trained on the train set of triplets, it gives the mAP of *0.2127*. When we train the model on both train and validation set of triplets, mAP of *0.2735* is achieved. Figure 4 presents six search results by our best-performance model, all the top-*10* results for each individual query are meaningful. Even user-shot images undergo various irregular geometric transformation, they are still able to match with the correct e-commerce images. Owing to strong power of the embeddings produced by triplet network, details of fashion items are well captured and preserved for item-matching.

## IV. CONCLUSION

By participating in fashion-item matching task in JD AI Fashion Challenge, we propose a way of using triplet training for fashion-item recognition and matching by inspiration of FaceNet and triplet loss. With careful architecture design and parameter tuning, the embedding average pooled from convolutional feature maps presents satisfying performance in later retrieval stage.

## REFERENCES


[1] G. Awad, W. Kraaij, P. Over, and S. Satoh, "Instance search retrospective with focus on TRECVID," *IJMIR*, vol. 6, no. 1, pp. 1–29, 2017.

[2] A. Babenko and V. S. Lempitsky, "Aggregating deep convolutional features for image retrieval," *CoRR*, vol. abs/1510.07493, 2015.

[3] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky, "Neural codes for image retrieval," in *ECCV*, 2014, pp. 584–599.

[4] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[6] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016, pp. 1096–1104.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[8] A. Salvador, X. G. i Nieto, F. Marques, and S. Satoh, "Faster r-cnn features for instance search," in *CVPR*, 2016, pp. 9–16.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[10] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *ICLR*, 2016.