# Customizing Clothing Retrieval Based on Semantic Attributes and Learned Features

Huawei Liu, Peng Zhou, and Yudi Tang

School of Software
TongJi University, Shanghai, 201804, China

**Abstract.** Clothing attributes recognition and clothing retrieval have drown a lot of attentions in recent years. However, most of works focus on retrieving the most similar clothing to match the given query image, and ignore user's extra requirements. To address the problem of rigid way for retrieving clothing images, we propose a novel approach for clothing retrieval based on semantic attributes and learned features with deep learning in a customized manner. Given a clothing query and user's extra clothing attributes demands, we use a clothing detector to crop the clothing part out and feed it into a convolution neural network to extract the learned features and recognize semantic attributes, by taking both predicted attributes and required attributes into consideration, we can filter out a lot of irrelevant images in database, at last, we will use the learned features as clothing representation to retrieve remaining clothing images in database. In this way, we will give better matching images according to user's demands and speed up the process of retrieval by filtering out irrelevant images in an elegant way.

**Keywords:** Clothing retrieval, Customizing retrieval, Semantic attributes

## 1 Introduction

There have been a lot of works for clothing retrieval [2][5][6][11][12] in the field of computer vision. But most of the works could be divided into two streams, the first is content-based clothing retrieval and another is attributes-based clothing retrieval. Here, we give an analysis of both ideas and illustrate the inspirations of our work.

When doing content-based image retrieval, people usually extract hand-draft features of different parts in the image and take the combined features as the representation of the image or directly learn features of the whole image using deep learning techniques, then, they will compute the similarity between the query clothing image and clothing images in the database based on their feature representations. For attributes-based works, people define a lot of semantic attributes to annotate the clothing images in database, and train an attribute classifier to recognize defined semantic attributes when given a query clothing, the predicted attributes will be converted into binary codes and act as the feature

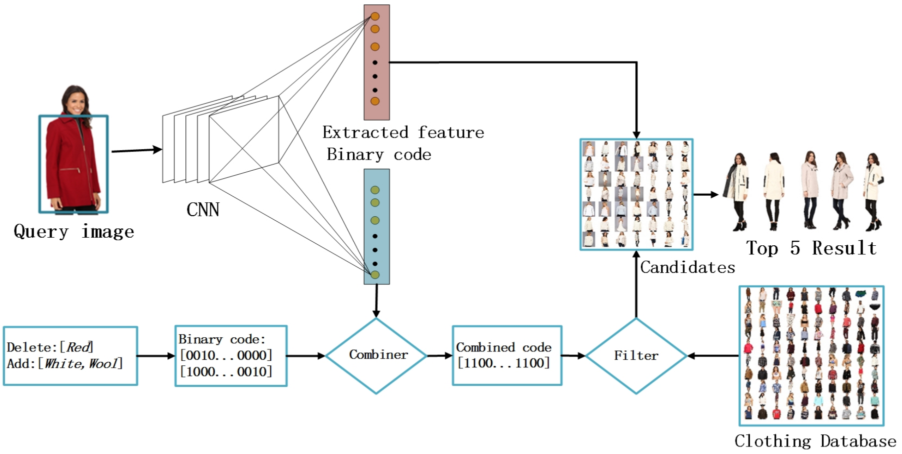


**Fig. 1.** The framework of our work. Two inputs will be given from the user, the first input is a query image, which is necessary. The second input is user's optional requirements, which include two parts, one for collecting undesired attributes, and another for holding supplementary attributes. Clothing part will be cropped from the query image and fed into a VGGNet to extract learned features(a 264-d feature vector) and recognize attributes(a 264-d binary code). User's additional attributes will be transformed into binary code and combined with predicted binary code to filter out irrelevant images in database. Finally, clothing are retrieved by computing similarity between learned features of query image and filtered candidates.

representation of the query clothing, finally, they will compute the similarity of images based on the binary codes of clothing attributes. In summary, there are pros and cons for both methods.

By using hand-draft features or learned features to represent the clothing images, we can give better results, because we use features of the whole image as feature representation, but when given a query image, we need to compute similarity with all images in the database, and it seems awful when the image database is quite large. However, by using binary codes of classified attributes as the image representation, we are able to filter out a lot of irrelevant clothing images and compute similarity with only small subset of images in database, but attribute-based similarity computing is not a safe way, for example, a query clothing is tagged *blue, sleeveless, t-shirt, animal-print*, you may find a lot of exact matches in your database, while the content and clothing details may differ.

In addition, almost all previous works focus on retrieving most similar ones with a query image, and don't take user's other requirements into consideration. Imaging the following scenario, i take an image of a fantastic t-shirt, but i expect a red one and in lace material. It is practical for users to have additional requirements, but existed methods can not solve this kind of demands.

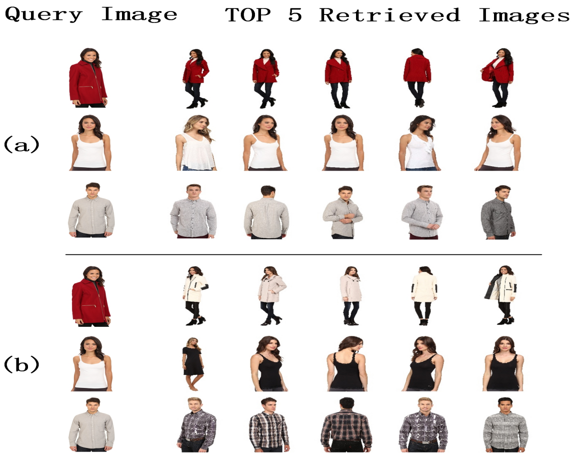


**Fig. 2.** Comparisons between general clothing retrieval and our customizing clothing retrieval. The first column is query images, and followed 5 columns are the top-5 retrieval results. Group (a) shows results of general clothing retrieval. Group (b) shows results of our method, corresponding deleting lists are separately [*Red*], [*White*], *empty*, and adding lists are separately [*White, Wool*], [*Black, Lace*], [*Plaid*].

In our paper, we take all three ingredients into consideration, which include clothing content, semantic attributes to describe the clothing and the most important one, additional user requirements. Whole framework of our work has been shown in Figure 1. We will use a lot of attributes to annotate each clothing(264 attributes in our experiments). Given a query image, we are using a VGGNet [13] to extract learned features and recognize clothing attributes, by combing the predicted attributes and user's additional requirements, we can construct a new attribute vector to describe the query image. Then, the new constructed attributes will be used to filter out a lot of irrelevant images in the database, this saves us a lot of time for unnecessary distance computing with the irrelevant images. At last, we will use the feature vector extracted from the VGGNet to compute the similarity between query image and remaining images in the database, and recommend clothing images with close similarity to users. Comparisons of the retrieval results have been shown in Figure 2.

In this way, with binary codes of clothing attributes, we are able to save time on unnecessary computing. By introducing learned features, we can give better retrieval results. Additionally, with user demands specified, we can recommend clothing user favors the most. Concretely, contributions of our paper are as follows.

1. We use VGGNet to recognize clothing attributes, and experimental results indicate that it performs better than previous methods.

2. We compare the performances on general clothing retrieval by separately using binary codes and extracted features as clothing representation, and exper-

imental result shows extracted feature vector can give better general retrieval accuracy .

3. We propose a novel idea of customizing clothing retrieval with semantic attributes and learned features based on deep learning, by taking both user requirements and clothing content into considerations, we can recommend more consilient clothing to users.

The remainder of our paper is organized as follows. In section 2, we will review a lot of related works. In section 3, we will introduce all the building blocks of our overall framework. In section 4, we will give the experimental deigns of our framework and show the experimental results. Finally, conclusions of this paper will be made in section 5.

## 2 Related work

Clothing related studies are a big family, which include clothing and people description [4][7][18], clothing items detection [2][8][18][19], clothing attributes recognition [3][6][7], clothing retrieval and recommendation [2][5][6][11][12] and so on. Here we will mainly introduce clothing attributes recognition and clothing image retrieval, which are two main building blocks of this paper. Both the fields are needing expressive features to represent clothing images, and there are mainly two ways to learn the features, the first is traditional feature learning method and the other is feature leaning based on deep learning methods.

In the past, most of works use hand-craft features and followed with traditional machine learning methods such as SVM [22] or Logistic Regression [23] to recognize clothing attributes. The representative work is [3], which uses a upper body detector to locate human parts in an image, and extracted hand-craft features of the located parts, these features are then fed into two high level classifiers, a random forest for classifying the type of clothing and several Support Vector Machines(SVMs) for describing the style of the apparel. Liu *et al.* [2] extract HOG, LBP and other hand-crafted features from human parts and use a nearest neighbor search to solve a cross-scenario clothing retrieval problem.

However, during recent years, deep learning methods have shown its power on great number of computer vision tasks. People come to use deep learning methods to learn feature representations of clothing images. Hara *et al.* [8] use R-CNN based detectors to detect various fashion items a person in an image is wearing or carrying, and they use the pose estimation results as a prior to correct the item locations. Lao *et al.* [9] uses fine-tuned convolutional neural networks to do clothing classification by replacing the output number of soft-max layer into the number of clothing categories. Gupta *et al.* [10] is another work to classify clothing type and attributes using deep learning, which uses pre-trained AlexNet [20] to extract the image features without fine-tuning the whole network, and feeds the features to a multi-class learner based on Logistic regression.

Similar condition happens in the filed of clothing retrieval. Lin *et al.* [6] use the fine-tuned AlexNet with an additional latent layer to learn hashes-like representation of clothing images, and taking a coarse-to-fine strategy to perform fast clothing retrieval on a large clothing dataset. This work gives us a lot of inspirations to speed up our retrieval process, but the difference is, they add an additional latent layer to learn binary codes, and use 4096-d fc7 layer output as the feature representation, which needs a lot of computing resource. Our work can learn both binary codes and feature representation in a row as shown in Figure 3, and our learned feature is only 264-d, which can give better speed when computing similarity between clothing images. Huang *et al.* [11] use a cnn-based dual attribute-aware ranking network for the problem of cross-domain image retrieval, given a user's clothing image, it can retrieve the same or attribute-similar clothing items from online shopping stores. Liu *et al.* [12] use a modified VGG-16 deep CNN network for multi-label clothing predictions, they implement clothing retrieval by taking the predicted attributes as the feature representations of clothing images.

Existing approaches for clothing retrieval is a little rigid which always focus on retrieving the most similar ones to recommend to users and can not take user's additional requirements into consideration. In our work, we will give a novel idea of customizing clothing retrieval based on semantic attributes and learned feature with deep learning techniques, by taking both query image and user's attributes requirements into consideration, we provide a flexible way for users to choose clothing that they favor the most.

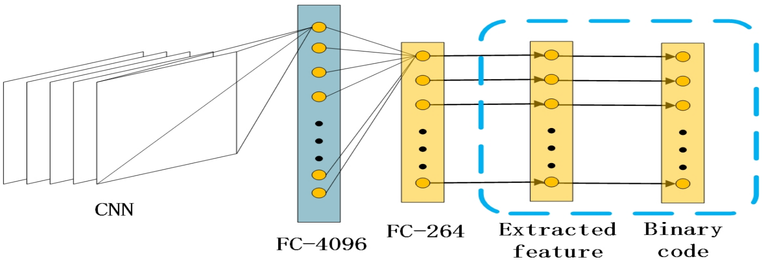


**Fig. 3.** FC-4096 is fully-connected layer before the output layer, and fc-264 is our score layer, as our task is a multi-label classification problem, we will use sigmoid function to the score output, activations of sigmoid function will be our extracted feature vector. Then, we will cut the activations at threshold of 0.5 to get our binary codes, with each entry indicates whether corresponding attribute exists in the query clothing.

## 3 FrameWork

Entire framework of our model has been shown in Figure 1. It contains: 1) a faster-rcnn-based [14] clothing detector to crop out the clothing part of an image; 2) cropped clothing will be fed into a VGGNet [13] to recognize the attributes and extract the learned feature to represent the clothing; 3) take both the recognized attributes and user's requirements into consideration and use the combined attributes to filter out images in the database, remaining images will be the candidates of final outputs; 4) compute similarity between the learned feature and feature representations of candidates which are pre-computed and return the top-k most relevant clothing to the user.

### 3.1 A clothing detector based on faster-rcnn object-detecting algorithm(optional)

Faster rcnn [14] is a object detection network the achieves nearly real-time object detection by introducing a region proposal network based on fast rcnn [21]. It achieves state-of-art object detection accuracy on PASCAL VOC 2007 and 2012 dataset while reducing detection time from 2.2s per image to 0.2s per image.

In our work, we train a clothing detector by training faster rcnn on a lot of clothing images with necessary annotations. Clothing detector is only an optional process in our work, which is mainly used to crop clothing part out of street photo or photos with a lot of background noise, however, for clothing images in online shopping mall, we can skip this step.

### 3.2 Recognize clothing attributes and extract clothing features based on VGGNet

Although there are a lot of models that can be used for clothing attributes recognition task, VGGNet [13] gives us the best accuracy on attributes classification, detailed results will be illustrated in next section.

Although we can represent our clothing with binary codes of predicted attributes, each entry indicates whether specific attribute appears in that clothing. But for scenario like, predicted attributes are *Black, Plaid, LongSleeve, Blouse*, there are a great number of clothing in database that have exact the same attributes but maybe in different content or style. So, only considering the predicted attributes is not enough, we also need to consider the clothing content. In our work, we extract another feature vector to represent each clothing, and experimental result indicates, it gives us better accuracy in general clothing retrievel problem as shown in Table 2.

So, in our work, clothing attributes are important factors for filtering clothing images in database which can save us a lot of time for unnecessary computing. And clothing features that extracted from VGGNet can act as clothing representation for computing similarity between query image and images in the database.

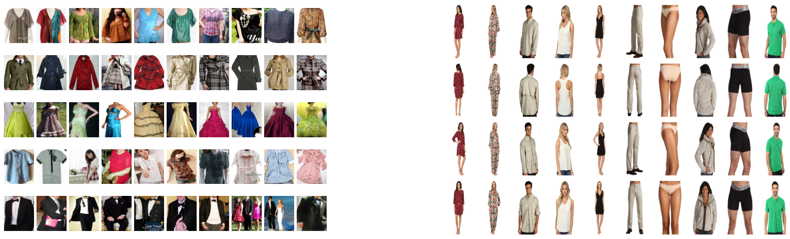
### 3.3 Combine clothing attributes and user's attributes requirement

With the above two modules, we can recommend the most similar clothing images in the database to user when given a query clothing image, and achieve time speedup by filtering out irrelevant clothing images in database using the predicted clothing attributes. But, as we have said in the beginning, this is a rigid way for recommending clothing and without taking user's specific attributes requirements into consideration. Here, we can address this case by combing user's attributes requirement with our predicted clothing attributes.

The main intuition is, we can give user a list of attributes that they can choose, and another two lists that one for collecting undesired attributes, and the other for holding supplementary attributes. Then, given a query clothing image and the two optional lists, we run a VGGNet to predict binary codes of clothing attributes and extract the learned feature to represent this clothing, and each entry of binary codes vector indicates whether specific attribute exists, so for each attribute in adding list, we will change the corresponding code into one in binary vector, and for each attribute in deleting list, we will change the corresponding code into zero. Final binary code vector will act as the required attributes to filter the clothing images in the database. For example, a query image is predicted as [*Blue, T-shirt, AnimalPrint*], and adding list is [*Red*], deleting list is [*Blue*], so final attributes vector will be [*Red, T-shirt, AnimalPrint*], and we will filter out all images that without specified attributes. This is just an illustrating example, in our experiment, we will convert lists into binary vectors in advance for fast computing. By the way, if the adding and deleting list are empty, our model is the same as retrieving the most similar clothing images with the queried one.

### 3.4 Recommend top-k clothing images in database

The above three modules have almost given the full concept of our work, now, we have feature representation of the query image and filtered candidates in the database which is a small subset of the original image database, equally, we hold the corresponding feature representations for each of the filtered images. The task here is to compute the *L2* norm distances between images based on respective feature representations and rank the images with respect to the distance, finally, top-k clothing will be returned to user as the feedback.



(a) ACS samples (b) MVC samples

**Fig. 4.** Sample images of our dataset

# 4 Experiments

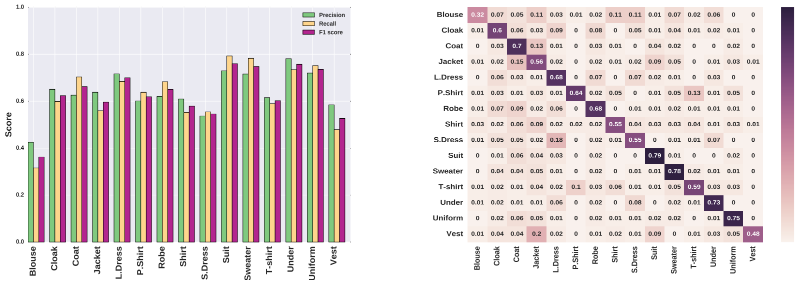
## 4.1 Datasets introduction

In our experiments, we will use two public datasets, the first is Apparel Classification with Style(ACS) [3] Dataset, which contains 89,484 clothing images of 15 different categories that has been cropped based on bounding box to catch the main clothing part on individuals upper body. This dataset will be used to test the performance of VGG model for clothing type classification.

Another is Multi-View Clothing(MVC) [12] Dataset, which contains more than 160,000 clothing images with each clothing contains four different views and is annotated with 264 attributes to describe clothing appearance. It will be used for two experiments. The first is the general clothing retrieval, we will compare the retrieval accuracy by separately using the attribute-based binary codes and feature vector extracted from VGG model. The second is our customizing clothing retrieval experiment, it will act as the backend training dataset. Sample images of both datasets have been shown in Figure 4.

**Table 1.** Comparison of results for the clothing type classification task. Our VGGNet gives better results than previous methods.

| **Models** | **Accuracy** |
|---|---|
| SVM(Bossard et al.) | 35.0% |
| Random Forest(Bossard et al.) | 38.3% |
| Transfer Forest(Bossard et al.) | 41.4% |
| Fine-tuned Fully-Connected Layers CaffeNet(Brain et al. [9]) | 46.0% |
| Fine-tuned All Layers CaffeNet(Brain et al. [9]) | 52.0% |
| **Our fine-tuned VGGNet** | **66.54%** |



(a)

|  | Blouse | Cloak | Coat | Jacket | L.Dress | P.Shirt | Robe | Shirt | S.Dress | Suit | Sweater | T-shirt | Under | Uniform | Vest |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Blouse | 0.32 | 0.07 | 0.05 | 0.11 | 0.03 | 0.01 | 0.02 | 0.11 | 0.11 | 0.01 | 0.07 | 0.02 | 0.06 | 0 | 0 |
| Cloak | 0.01 | 0.6 | 0.06 | 0.03 | 0.09 | 0 | 0.08 | 0 | 0.05 | 0.01 | 0.04 | 0.01 | 0.02 | 0.01 | 0 |
| Coat | 0 | 0.03 | 0.7 | 0.13 | 0.01 | 0 | 0.03 | 0.01 | 0 | 0.04 | 0.02 | 0 | 0 | 0.02 | 0 |
| Jacket | 0.01 | 0.02 | 0.15 | 0.56 | 0.02 | 0 | 0.02 | 0.01 | 0.02 | 0.09 | 0.05 | 0 | 0.01 | 0.03 | 0.01 |
| L.Dress | 0 | 0.06 | 0.03 | 0.01 | 0.68 | 0 | 0.07 | 0 | 0.07 | 0.02 | 0.01 | 0 | 0.03 | 0 | 0 |
| P.Shirt | 0.01 | 0.03 | 0.01 | 0.03 | 0.01 | 0.64 | 0.02 | 0.05 | 0 | 0.01 | 0.05 | 0.13 | 0.01 | 0.05 | 0 |
| Robe | 0.01 | 0.07 | 0.09 | 0.02 | 0.06 | 0 | 0.68 | 0 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.01 | 0 |
| Shirt | 0.03 | 0.02 | 0.06 | 0.09 | 0.02 | 0.02 | 0.02 | 0.55 | 0.04 | 0.03 | 0.03 | 0.04 | 0.01 | 0.03 | 0.01 |
| S.Dress | 0.01 | 0.05 | 0.05 | 0.02 | 0.18 | 0 | 0.02 | 0.01 | 0.55 | 0 | 0.01 | 0.01 | 0.07 | 0 | 0 |
| Suit | 0 | 0.01 | 0.06 | 0.04 | 0.03 | 0 | 0.02 | 0 | 0 | 0.79 | 0.01 | 0 | 0 | 0.02 | 0 |
| Sweater | 0 | 0.04 | 0.04 | 0.05 | 0.01 | 0 | 0.02 | 0 | 0.01 | 0.01 | 0.78 | 0.02 | 0.01 | 0.01 | 0 |
| T-shirt | 0.01 | 0.02 | 0.01 | 0.04 | 0.02 | 0.1 | 0.03 | 0.06 | 0.01 | 0.01 | 0.05 | 0.59 | 0.03 | 0.03 | 0 |
| Under | 0.01 | 0.02 | 0.01 | 0.01 | 0.06 | 0 | 0.02 | 0 | 0.08 | 0 | 0.02 | 0.01 | 0.73 | 0 | 0 |
| Uniform | 0 | 0.02 | 0.06 | 0.05 | 0.01 | 0 | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | 0 | 0.01 | 0.75 | 0 |
| Vest | 0.01 | 0.04 | 0.04 | 0.2 | 0.02 | 0 | 0.01 | 0.02 | 0.01 | 0.09 | 0 | 0.01 | 0.03 | 0.05 | 0.48 |

(b)

**Fig. 5.** (a) *precision, recall,* and *F1 scores* on a per-class basis; (b) *Confusion matrix* of our model results on ACS dataset

### 4.2 Results

A 16-layer VGG model is used for all the following three experiments. And all the experiments are implemented with Caffe [1] framework.

**Clothing attributes classification:** We use 16-layer VGG model to classify clothing categories, which is multi-class classification task. The VGG model pretained on ImageNet [17] dataset is fine-tuned on our ACS dataset, we modify the final fully connected layer output from 1000 to 15, because in ImageNet dataset, there are 1000 classes to classify, but in our ACS dataset, there are only 15 classes. As we have enough data to afford the capacity of the model, we finetune the whole network by setting the base learning rate of 0.0001, momentum weight update with momentum of 0.9, weight decay of 0.001, inv policy is used to decay our learning rate with gamma of 0.0001 and power of 0.75. Our model achieves 66.54% test accuracy on ACS dataset after 90,000 iterations. While previous models get best test accuracy of 52%. Details about the results have been shown in Table 1. We also present the *confusion matrix* of final results and compute *precision, recall, and F1 scores* on a per-class basis in Figure 5.

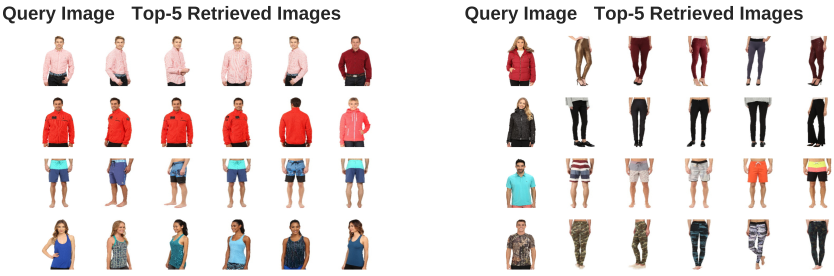
**General clothing retrieval:** Here, a 16-layer VGG model is mainly used to extract the features to represent different clothing images. For each image in MVC dataset, it is annotated with 264 different labels with each entry indicates whether specific label appears in the clothing. So, it is a typical multi-label classification problem. In this experiment, the soft-max loss function which is used for single-label classification in the original VGG model is modified into sigmoid cross-entropy loss for multi-label training in our task, and the last fully connected layer is replaced with 264 outputs. Detailed setting of the optimization is the same as previous experiment. And we train the model 100,000 iterations and achieve 74.6% *average f1 score* over all 264 attributes. But our main job is to extract the binary codes and feature vectors of specific images and compare their performance in general clothing retrieval task. We use a small trick as shown in Figure 3, as we know, the output of sigmoid layer is 264-d probability vector with each entry represent the probability of corresponding attribute existing in that clothing, this vector will act as the extracted feature vector. For binary codes, we cut the probability into 0 or 1 by setting threshold of 0.5, so entry below 0.5 will be assigned 0 and otherwise it will be assigned 1. With this small trick, we can achieve both binary codes and learned features without modifying any part of our model. Results of general retrieval has been shown in Table 2. We can see, by using extracted features as clothing representation, we achieve better retrieval accuracy. As a note here, both binary codes and feature vector are playing important roles in our work, here we just want to show, by using feature vector instead of binary codes as our clothing representation, it can give us better retrieval accuracy, however, binary codes will be used to filter images in the databased in case of unnecessary computing.

**Customizing clothing retrieval:** Lastly, exactly the same 16-layer VGG model from experiment two will be used in our final customizing clothing retrieval, which is the biggest contribution of this paper. We will not change any setting of the model. Both two experiments above are just building blocks of final experiment here. For experiment one, we just want to show you the powerful performance of VGG model in clothing attributes classification, and this is why we use it the recognize labels in clothing image. For the second experiment, we just want to give you the fact that using extracted feature vector as image representation can give better performance than binary label codes for clothing retrieval.

We give our final results in four different scenarios. For scenario one, given only the query image with both adding and deleting lists are empty, it's just retrieving most similar clothing like what previous works have done, detailed results have shown in Figure 6(a). For scenario two, users provide query clothing image and required attributes, and we will return results in a customizing manner as shown in Figure 2(b). Scenario three is a little special, our method can recommend corresponding matching clothing, as shown in Figure 6(b), given a *red coat* image and adding list of *leggings*, we will return a lot of pretty *leggings* for matching the coat. Last scenario is street photos, with faster rcnn specified in Section 3.1, we can crop the clothing part out of the photo and take cropped part as query image and attribute requirements as model input, and remaining part is the same as scenario two, Figure 7 gives us the details.

**Table 2.** Accuracy for top-1 and top-5 exact clothing match

| **Clothing Representation** | **top-1** | **top-5** |
|---|---|---|
| Binary codes | 17.8% | 35.2% |
| Extracted features | 48.6% | 67.8% |



(a) Scenario One (b) Scenario Three

**Fig. 6.** (a) Results of scenario one, just retrieve most similar clothing without additional attribute requirements; (b) Results of scenario three, we can recommend matching clothing

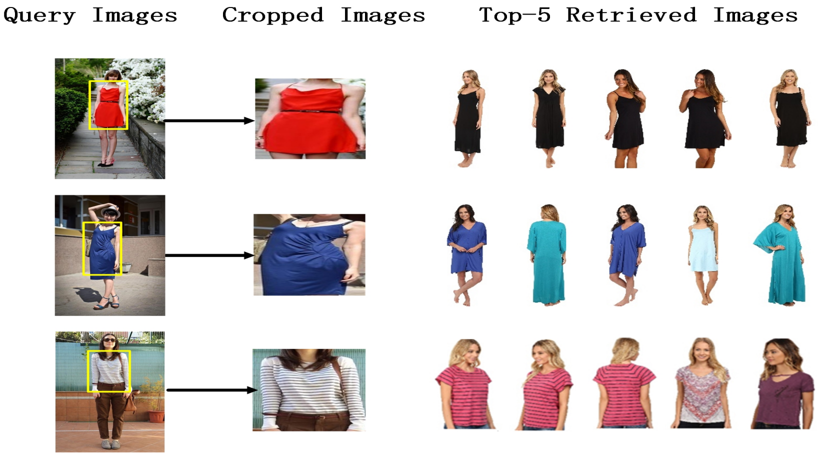


**Fig. 7.** Results of scenario four, given a street photo and additional user requirements, we crop the clothing part out with a clothing detector and with remaining process the same as scenario two. Here, adding lists are separately [*Black, Lace*], *empty*, [*Red, ShortSleeve*] and deleting lists are all *empty*.

## 5 Conclusions

We have presented a novel idea of customizing clothing retrieval by combining user requirements and predicted attributes to recommend consilient clothing images to users. To speed up the retrieving process, we use binary codes of combined attributes to filter out irrelevant clothing images in database. To achieve better retrieval results, we use learned features as image representation to compute similarity between clothing images. We also present experiments in four different scenarios and experimental results have shown that our method gives more flexibility than previous methods. In the future, we attempt to use more semantic attributes to annotate clothing, and try to use only binary codes of attributes to simultaneously filter clothing images and compute similarity, because distance computing between binary code vectors are very fast.

## References

1. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
2. Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-Shop: Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set. CVPR (2012)
3. L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. In Computer VisionACCV 2012, pages 321335. Springer (2013)
4. H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In Computer VisionECCV 2012, pages 609623. Springer (2012)

5. B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 801808. IEEE (2011)
6. K. Lin, H.-F. Yang, K.-H. Liu, J.-H. Hsiao, and C.-S. Chen. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pages 499502. ACM (2015)
7. L. Bourdev, S. Maji, and J. Malik. Describing People: A Poselet-Based Approach to Attribute Classification. In ICCV (2011)
8. Hara, Kota, Vignesh Jagadeesh, and Robinson Piramuthu. Fashion apparel detection: the role of deep convolutional neural network and pose-dependent priors. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2016)
9. Lao, B., & Jagadeesh, K. Convolutional Neural Networks for Fashion Classification and Object Detection (2015)
10. Gupta S, Agarwal S, Dave A. Apparel Classifier and Recommender using Deep Learning[J].
11. J.S. Huang, R.S. Feris, Q. Chen, S.C. Yan. Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network. arXiv:1505.07922v1 2015
12. Liu, Kuan-Hsien, Ting-Yen Chen, and Chu-Song Chen. MVC: A Dataset for View-Invariant Clothing Retrieval and Attribute Prediction. Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval. ACM (2016)
13. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
14. Ren, Shaoqing, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems (2015)
15. M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In Image Processing (ICIP), 18th IEEE International Conference on, pages 29372940. IEEE (2011)
16. H. N. Ng and R. L. Grimsdale. Computer graphics techniques for modeling cloth. Computer Graphics and Applications, IEEE, 16(5):2841 (1996)
17. Deng, Jia, et al. Imagenet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE (2009)
18. Chen, Qiang, et al. Deep domain adaptation for describing people based on fine-grained clothing attributes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
19. Yamaguchi, Kota, et al. Parsing clothing in fashion photographs. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE (2012)
20. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems (2012)
21. Girshick, Ross. Fast r-cnn. Proceedings of the IEEE International Conference on Computer Vision (2015)
22. Hearst, Marti A., et al. Support vector machines. IEEE Intelligent Systems and their Applications 13.4 : 18-28 (1998)
23. Hosmer Jr, David W., and Stanley Lemeshow. Applied logistic regression. John Wiley & Sons (2004)