FASHION CLOTHES MATCHING SCHEME LEARNED FROM FASHIONISTA’S SUGGESTIONS IN MICROBLOG

Guangyu Gao¹, Yihang Zhang¹, Songyang Du²

¹School of Software, Beijing Institute of Technology. Beijing 100081, China
²Beijing Special Vehicle Research Institute. Beijing 100072, China

guangyugao@bit.edu.cn,719360563@qq.com o_o918@163.com

ABSTRACT

Intelligent fashion analysis for clothing is receiving a great deal of attentions recently, since there is a huge profit potential in the fashion industry. Although there are several related studies about clothes matching, clothing matching and recommendation is still an challenging problem. This paper mainly introduces a fashion clothes matching scheme with consideration of the labeled data from fashionista in MicroBlog and unlabeled data from online shopping website. More specifically, the gallery data is collected starting from the matching clothes outfits recommended by fashionista in MicroBlog firstly. Then, these matching clothes are semi-automatically segmented into matched upper and lower parts as positive samples. Meanwhile, a semi-supervised clustering based assembling was proposed to generate negative samples to form a comprehensive dataset. After that, a classifier is trained to determine the testing upper and lower samples as matching or not. Finally, we conduct extensive experiments and empirical evaluations on a set of clothes images to demonstrate the usefulness and effectiveness of the scheme.

Index Terms—Fashion Analysis, fashionista, clothes matching, MicroBlog.

1. INTRODUCTION

In the past double 11 shopping festival in China, the Alibaba’s Tmall e-commerce site had sold more than 120.7 billion yuan (around $17.79 billion) worth of goods¹. That is to say, the potential market is expected to break enormous wealth, and studies on clothing are receiving increasing interests due to the huge market. Clothing relevant researches are flourished along with this. Liu et al. [1] provided an overview of such techniques and representative works on clothing analysis. However, related research literatures are still quite limited and most of them focus more on clothing segmentation, clothing recognition and clothing retrieval.

⁵This work was supported by the National Natural Science Foundation of China under Grant 61401023.
⁶https://www.chinainternetwatch.com/19280/singles-day-top-categories-2016/

Clothing segmentation is a basic step for other clothing based tasks, which refers to predict each pixel in a clothing image into different garment items. Clothing segmentation is an extremely challenging problem because of the large number of possible garment items and the variations in configuration, garment appearance, layering, and occlusion [1]. Kota Yamaguchi et al. developed an effective method for parsing clothing in fashion photographs, and also provided a large and novel dataset and tools for labeling garment items [2, 3]. In [4], a data-driven framework was proposed, which consists of two phases, the ‘image co-segmentation’ and the ‘region co-labeling’. By utilizing parselets and mixture of joint-group templates as the representations for semantic parts, Dong et al. [5] seamlessly formulated the human parsing and pose estimation problem jointly via a tailored And-Or graph.

Clothes recognition and retrieval is to identify clothing in images. In [6], Kalantidis et al. proposed the cross-scenario retrieval, and the query is a real-world image, while the products from online shopping catalogs are usually presented in a clean environment. A practical problem of cross-scenario clothing retrieval was also addressed in [7]. Kiapour et al. [8] matched a real-world example of garment item to the same item in an online shop, and also proposed a new dataset containing 404,683 shop photos and 20,357 street photos.
Meanwhile, they also proposed to discover elements of fashion style, based on a new dataset of clothing outfits with associated style rating for 5 styles categorized by crowd source.

In most previous approaches, the street photos or shop photos are collected and used mostly. Actually, by reviewing different social networks, we found that several fashionista publish the outfits, including matched clothes and accessories everyday, as shown in Fig. 1. It triggered us that if it is possible to use these images as expert data or gallery data for clothes matching assistance and recommendation. Therefore, in this paper, we propose a fashion clothes matching scheme with consideration of the labeled data from fashionista in MicroBlog and unlabeled data from online shopping website. The main contributions of this paper are:

- We focus more on outfit data published by the fashionista in social networks. And, these data from fashionista, combined with several unlabeled data from online shopping website, are used for upper and lower clothing matching determination.

- Outfits published by fashionista can be only look as positive sample for learning, a semi-supervised clustering based assembling was proposed to generate more reasonable negative samples to form a comprehensive dataset.

- The classifier was learned on these data, and a series of experiments was conducted to demonstrate the usefulness and effectiveness of the scheme.

2. PROPOSED ARCHITECTURE

2.1. Clothes Image Collection

Several clothing datasets have been constructed [9, 10], but nearly none of them is exactly suitable for clothes matching task. Nevertheless, Liu et al. [11] constructed the ‘What-to-Wear’ dataset, containing 24,417 clothing, which provided complete attribute and occasion annotations. Meanwhile, a large-scale clothes database, the DeepFashion Dataset is also proposed in [12]. Nearly all these images are life photo with people in different posture. Actually, for clothing matching task, the input or query images are always sole clothes images, thus, such datasets are not so suitable for our task.

We found that, several fashionistas, i.e. the VIP (Very Important Person) in Weibo or Twitter, regularly released matching outfits images, as shown in Fig. 1. These images could be looked as completely expert data for matching analysis. Therefore, these professional images are collected for our clothing matching task, as shown in Fig. 2. More specifically, with a crawler script, thousands of such images are downloaded automatically. After that, in order to get more precise matching parts, noise or unsuitable images are removed manually. As shown in Fig. 3, the defined noise or unsuitable images, are those images: 1) without clear boundary between upper and lower parts; 2) jumpsuit and one-piece clothes; 3) images with folded clothes. Finally, there remained 4498 matched expert images as the positive sample (matched upper and lower pairs), as shown in Fig. 2.

Since matching determination is subjective, we involved more general samples to enrich the positive samples from Fashionista. Actually, several E-shopping stores (Taobao.com etc.) shown their clothes with a fashionable model. The E-shopping stores always recommend a lower/upper clothes when they are propagandizing the upper/lower clothes. That is to say, these pairs of recommendations are also excellent expert data candidates with populace aesthetic for our task. Based on this, we manually collected another 500 matched images from such E-shopping stores. Thus, finally, we get 4998 matched clothes parts as positive samples.

2.2. Gallery Dataset Construction

2.2.1. Pre-Processing

Because the clothes images are collected from totally different sources, we need to do some pre-processing before the following image processing and machine learning. Firstly, we resize all those upper images to resolutions of $450 \times 450$, and lower images to resolutions of $300 \times 450$. In addition, since we need to extract and analyze the features on only clothes regions, the contour detection and region segmentation method is used to remove background (to set background region as white) [13] presented a unified approach to contour detection and image segmentation. So far, all those collected samples are transferred to unified size and format.
Fig. 3. Different noise or unsuitable images in the collected set, i.e., images without clear boundary, jumpsuit, one-piece clothes, and image with folded parts.

2.3. Feature Extraction

Actually, we extract features in upper and lower clothes separately, and concatenate the two part features together. In our scheme, color, texture and style are the most important properties when we say that a upper clothes is matched with the lower clothes. Thus, four types of state-of-the-art features are considered on both upper and lower clothes, i.e. LBP (Local Binary Pattern), HSV Color Histogram (HSV-CH), SIFT (Scale Invariant Feature Transform), and MSER (Maximally Stable Extremal Regions) features. Details of these features are listed below:

- Local Binary Pattern Feature (LBP). LBP captures the contrast information of the central pixel and its neighbors. We use a variant of LBP, i.e., the multiblock LBP [14]. In the feature selection, the image is firstly segmented into several blocks to keep a certain amount of geometric information. Each face image is divided into $5 \times 4$ sub-regions and then for each subregion uniform patterns are extracted and concatenated as bins for a histogram representation.

- HSV Color Histogram. Most of the clothes matching rules refer to color characteristics, thus, color feature always has more important impact. We analyze the color features on HSV (Hue, Saturation and Value) color space since it is useful in content based image retrieval. In HSV color space, the Hue is defined as an angle in the range $[0, 2\pi]$ relative to the red axis with red at angle $0$, green at $2\pi/3$, blue at $4\pi/3$ and red again at $2\pi$. Saturation is the depth or purity of the color and is measured as a radial distance from the central axis with value between $0$ at the center to $1$ at the outer surface.

- Scale Invariant Feature Transform (SIFT). We utilize SIFT descriptors [15] with scales of each interest point varying from 20 to 120 pixels. It describes the static appearance over spatial histogram.

- Maximally Stable Extremal Regions. Every extremal region is a connected component of a thresholded image. The regions are obtained by thresholding the intensity image and tracking the connected components as the threshold value changes. The idea is due to the work of Matas et al.[16].

As described in [17], the SIFT features are invariant to image scale/rotation and robust to changes in illumination, noise, and minor changes in viewpoint. Meanwhile, SIFT features exhibit the highest matching accuracies for an affine transformation of 50 degrees. But, after this transformation limit, results start to become unreliable. However, MSER as a method of blob detection in images, has been proved as one of the most robust feature detectors on invariance of affine transformation [18]. In order to generate the invariant description for MSER feature, an elliptical image region is used to cover the distinguished regions with the interest point as the center. Meanwhile, the elliptical region normalized to a circle, and the SIFT descriptor for the central point is output as the final MSER feature description.

Since different clothes image will have totally different numbers of SIFT/MSER features, the Bag of Word idea is introduced to map each image into a unified feature vector. That is to say, when all the sample images, including upper and lower parts, are collected together, the SIFT/MSER features are extracted at first. And then, all these feature descriptors are clustered into $k$ clusters with the k-means clustering. Thus, for both SIFT and MSER features, each upper/lower image will be represented as a feature vector $f = (c_0, \ldots, c_i, \ldots, c_{k-1})$, where $c_i$ means the number of feature points belongs to the $i$th cluster. Finally, we extract features in upper and lower clothes separately, and then to concatenate these two part features together.

2.4. Comprehensive Training Dataset

So far, we have 4998 pairs clothes images. For learning problem, both positive and negative samples are needed. In our approach, those collected matched images can be looked as the positive samples only, that is to say we still need the negative samples (pairs of image which is not matching). In fact, we need to construct a reasonable training dataset with both positive and negative samples. In other word, we need some more efficient negative samples.

Actually, we considered two ways to generated the negative samples for learning. An intuitive idea, is to randomly pair the upper and lower clothes from the collected upper and lower parts to form a batch of clothes pairs. Essentially, these clothes pairs are negative sample candidates. However,
these negative samples are with sparse consistency, and the inter-difference between matching and non-matching is not so obvious. Therefore, we proposed another negative sample generation method with consideration of more compact inner-class and intra-class difference.

As mentioned before, we have 4998 pairs matching clothes (upper and lower clothes). Firstly, the 4998 upper and lower clothes are both clustered into \( N \) classes separately, using the \( k \)-means algorithm with the Euclidean distance between feature vectors. A similarity graph \( G = \{ V, E \} \) is constructed based on these clusters. Each cluster is a vertex in \( G \), and all the vertices are separated into two parts corresponding to upper and lower parts. The edge in \( G \) refers to original matching pairs. That is to say, edges only exist between clusters of upper and clusters of lower clothes, and the weight of edge means the number of matching pairs. Ideally, there will be \( N \times N \) connections in graph \( G \). Each connection (edge) \( v_{i,j} \) is defined as \( e(u_i, l_j) \), where \( u_i \) and \( l_j \) means the \( i_{th} \) and \( j_{th} \) cluster in the upper and lower clothes images separately. The connection weight \( w(e) \) is defined as,

\[
w(e) = \sum_i \sum_j m(u_i, l_j)
\]

where \( m(u_i, l_j) \) refers to that if \( u_i \) and \( l_j \) come from the same matching outfit image. Negative samples should be pairs with very low matching probability. Therefore, negative samples should be cluster pairs with very small connection weight. Actually, for each upper clothes cluster \( u_i \), we choose the \( j_{th} \) lower cluster with \( l_j = \text{argmin}_i \sum_j m(u_i, l_j) \) as the negative cluster pairs. After that, by randomly matching the upper and lower clothes in these two chosen negative clusters, a batch of clothes pairs are generated as the negative samples for learning. That is to say, for the upper and lower clusters with smallest connection weight, the \( x_{th} \) image \( c_{u_i,x} \) in cluster \( u_i \) and the \( y_{th} \) image in cluster \( l_j \) can form a series of negative samples as,

\[
\text{Neg} = (c_{u_i,x}, c_{l_j,y}), \text{where } 0 < x < ||u_i||, 0 < y < ||l_j||, \text{and } m(c_{u_i,x}, c_{l_j,y}) = 0.
\]

where \( ||u_i|| \) and \( ||l_j|| \) means the number of images in clusters \( ||u_i|| \) and \( ||l_j|| \), and the function \( m(c_{u_i,x}, c_{l_j,y}) \) means that if the two images are a matched pairs from the fashionista or not (if it is, it will be 1, otherwise, 0). Actually, the number of negative samples depends on the number of images in cluster \( u_i \) and \( l_j \). Therefore, this value will be very small if there are little images in these two clusters. If so, upper and lower clusters with the second or the third smallest connection weight will also be considered, to make sure the ratio of positive and negative sample to be about 10/8. So far, a reasonable and comprehensive dataset was generated.

3. CLASSIFIER LEARNING

Let \( D = \{ D_p, D_n \} \) be the dataset for training and testing, where \( D_p \) and \( D_n \) refer to the positive and negative samples separately. Since we have 4998 positive samples, and 4000 negative samples finally. We choose 7000 samples for training, including 4000 positive samples and 3000 negative samples, and the remained are used for testing. Here, each sample is a pairs of images (upper and lower clothes). For each image, four types of features, i.e. LBP, HSV histogram, SIFT and MSER, are extracted as mentioned before.

After that, the classifier is learned for matching recommendation. Since the Support Vector Machine (SVM) is a typical classification model for learning problems, we adopt the libsvm toolbox\(^2\). In the following paragraph, we briefly review the SVM based learning for matching recommendation based on the comprehensive dataset.

Given that, each sample, or pairs of images are represented as \( x_i \). In SVM, classifier \( f \) is defined as,

\[
f(x) = \omega^T x + b
\]

and, \( f \) learned by minimizing the objective function:

\[
\mathcal{J}(f) = \frac{1}{n} \sum_{i=1}^{n} \max(1 - y_i f(x_i), 0) + \lambda \|\omega\|^2,
\]

where the parameter \( \lambda \) determines the tradeoff between increasing the margin-size and ensuring that the \( x_i \) lie on the correct side of the margin. Meanwhile, in order to create nonlinear classifier by applying the kernel trick to maximum-margin hyperplane, that is to say, every dot product is replaced by a nonlinear kernel function.

4. EVALUATION

In this section, we evaluate the fashion clothes matching scheme with consideration of the comprehensive dataset, which is consisted by data from fashionista in MicroBlog and unlabeled data from online shopping website. Meanwhile, the SVM classifier is trained and used to classify testing samples into matching or not. In the following subsections, we first demonstrate the learning performance on the comprehensive dataset, when we separate all those data samples into training and testing subset. Then, we also show a simple fashion clothes matching or recommendation application, and its effectiveness was evaluated by user study.

4.1. Performance on the Comprehensive Dataset

In this subsection, we mainly evaluate the classification performance of SVM classifier with different parameter configurations on the comprehensive dataset, and the effectiveness of the proposed comprehensive dataset.

\(^2\)https://www.csie.ntu.edu.tw/~cjlin/libsvm/
Firstly, three common kernels with different parameter configurations are considered to test the learning performance. Actually, we used three types of kernels,

- Polynomial kernel (inhomogeneous):
  \[ k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d. \]

- Gaussian radial basis kernel:
  \[ k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\lambda \|\mathbf{x}_i - \mathbf{x}_j\|^2). \]

- Sigmoid kernel:
  \[ k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\nu \mathbf{x}_i^T \mathbf{x}_j + c). \]

As shown in Table 1, we found that most of the time the SVM with Gaussian kernel get a better performance. In this paper, we mainly describe the novelty and reasonable of the construction of comprehensive dataset and the flowchart of the clothes recommendation scheme. Therefore, if the SVM is replaced by some other more robust classifier, the performance will be better. In Table 1, we also found that Sigmoid kernel with some specific parameters roughly got the same performance as Gaussian kernel. While using the Sigmoid kernel, it is similar to realize a multilayer perceptron neuron network. That is to say, with the popular deep neuron network architecture, the performance will be better and better, and that’s our future direction.

Except for that, a baseline is also defined to show the effectiveness of the comprehensive dataset. By completely random choosing a upper and lower image to form a pair as negative samples, we get the baseline dataset with the same size of the comprehensive dataset. The best configuration of SVM in Table 1 was implemented again on the baseline dataset (Random Dataset, RData), and the effectiveness of the proposed Comprehensive Dataset (CData) was also shown in Table 1.

### 4.2. Fashing Clothes Matching Application

Except for the learning performance evaluation on the objective dataset, we also proposed a simple recommendation application based on the proposed scheme. The user study is also adopted to show the availability of this scheme. With the proposed classifier learning on the comprehensive dataset, twelve participants (10 female and 2 male with their age range from 19-30) are involved. The application is simple, which is queried by a upper or lower clothes, and it returns back the best matched lower or upper parts. User impression evaluates the user experience in viewing the recommended upper or lower clothes, by their satisfaction and acceptability. Actually, we ask each user to response that how good and satisfactory the recommend result is. The user impression is evaluated in terms of three criteria: feasibility (if this application is feasible), acceptance (whether the users like the matched clothes), and satisfactions (if the application and results is comprehensive to the user). Each user is asked to assign a score of 1 to 10 (higher score indicating better experience) to the above criteria. Fig. 4 shows the averages of the three criteria. We can see that it was well enough for the feasibility and acceptability, but not so good for satisfactions.

We also evaluated the application in terms of the top 3 and 5 results. That is to say, after the scheme returns the best matching part, this part is removed, and the new best part will be returned again in the remained ones. Finally, the three/five returned parts are presented to ask if there is any one of the parts make him/her satisfactory. While each user is required to query five times, there are totally 60 queries. Although the experiments of satisfaction in Fig. 4 was not so good, in these 60 queries, 52 times are labeled as yes for top 3 results and 48 for top 3 results, as shown in Fig. 5.

### 5. CONCLUSIONS

In this paper, we introduced a fashion clothes matching scheme with consideration of the data from social networks and online shopping website. Firstly, the gallery data was collected by crawling the fashionista recommended outfits from social network. Secondly, these outfit images are segmented
into matched upper and lower parts as positive samples. Meanwhile, a semi-supervised clustering based assembling was proposed to generate negative samples to form a comprehensive dataset. Thirdly, we learned a robust SVM classifier with different parameter configurations. Finally, extensive experiments are conducted on a comprehensive dataset.

6. REFERENCES


