A CNN-based Model with Multiple Binary Classifiers for Clothing Style Recognition

Junyu Lin, Xianjing Han, Xin Yang, Xuemeng Song
School of Computer Science and Technology
Shandong University, Tsingtao, China
{linjunyu1120, hanxianjing2018, joeyangbuer, sxmustc}@gmail.com

***Abstract*—Convolutional neural networks (CNNs) are at the core of many state-of-the-art computer vision solutions for a variety of tasks, such as object detection, video classification, and object tracking. In this paper, we propose a CNN-based method with multiple binary classifiers to efficiently recognize the clothing styles. We benchmark our method on the JD AI Fashion Challenge dataset. To ensure the quality of the training process, we apply the data augmentation to the uneven data before training. By experimenting on the dataset, we show the effectiveness of the proposed method in clothing style recognition, where we can achieve the F2 scores of 0.7319 and 0.5654 for the validation and test dataset, respectively.**

I. Introduction

With the increasing purchasing power of customers, recent years have witnessed the prosperity of the fashion market. Overall, all the clothes can be broadly classified into several styles, such as the *Ladylike*, *Sportive* and *Casual*. Different people may prefer different clothing styles due to their personal tastes. Moreover, even the same person can prefer different clothing styles for different occasions. Therefore, clothing style recognition can facilitate many practical applications, such as the personalized clothing recommendation, occasion-oriented clothing recommendation and clothing retrieval, and does merit our special attention. We thus participated the JD AI Fashion Challenge and investigated the task of clothing style recognition.

However, accurately recognizing the clothing styles is non-trivial due to the following challenges. 1) The real-world clothing images can be rather noisy, due to the different model poses, various lighting conditions, and the noisy backgrounds. 2) Clothing style recognition is a multi-label task, where each given clothing can belong to multiple styles. Namely, the number of labels for each clothing is not constant. 3) The positive and negative image sample for each style can be rather uneven.

In fact, several efforts have been made in the area of clothing style recognition. In [1], the authors proposed a method to describe clothing appearance with semantic attributes. Kiapour et al. [2] tackled the task of matching a real-world example of a garment item to the same item in an online shop. The most related work to ours is the apparel classification with styles [3]. Bossard et al. [3] proposed a multi-class classifier for recognizing and classifying clothing. However, this classifier can only tell the clothing category (e.g., sweater, skirt and coat), rather than the clothing styles.

TABLE I
THE NUMBER OF POSITIVE AND NEGATIVE SAMPLES IN EACH STYLE.

ID	Style	#Positive	#Negative
1	Sportive	104	54,804
2	Casual	4,292	50,616
3	OL	49,893	5,015
4	Japanese	262	54,646
5	Korean	18,722	36,186
6	Western	8,882	46,026
7	England	197	54,711
8	Girlish	1,356	53,552
9	Ladylike	49,389	5,519
10	Minimalist	7,094	47,814
11	Natural	3,712	51,196
12	Punk	1,249	53,659
13	Ethnic	361	54,547

In this paper, we propose a CNN-based method with multiple binary classifiers to realize style recognition for clothing. The dataset we use is the JD AI Fashion Challenge Dataset, which is a collection of annotated clothing images. Each clothing image has 13 labels with the value of 1 or 0, representing if the clothing belong to the specific style or not. We thus train 13 binary classifiers for all these 13 styles. Moreover, to boost the performance of our model, we deal with the uneven dataset with the data augmentation. The experimental results show that the proposed method is effective and efficient for the clothing style recognition.

The rest of this paper is organized as follows: Section II describes the dataset. Section III presents the proposed model. The performance evaluation of the proposed model is described in Section IV. Finally, conclusion is drawn in Section V.

II. Dataset

The dataset we use is provided by the JD AI Fashion Challenge Committee, which contains 54,908 clothing images. There are two kinds of images. One is the picture of one single model wearing clothes. These models are in a wide variety of poses, and photographed indoors under varying lighting conditions in cluttered scenes. The other is the picture of an item of clothing. Each clothing image has 13 binary labels, corresponding to the 13 styles. Table I lists the number of positive and negative samples in each style.

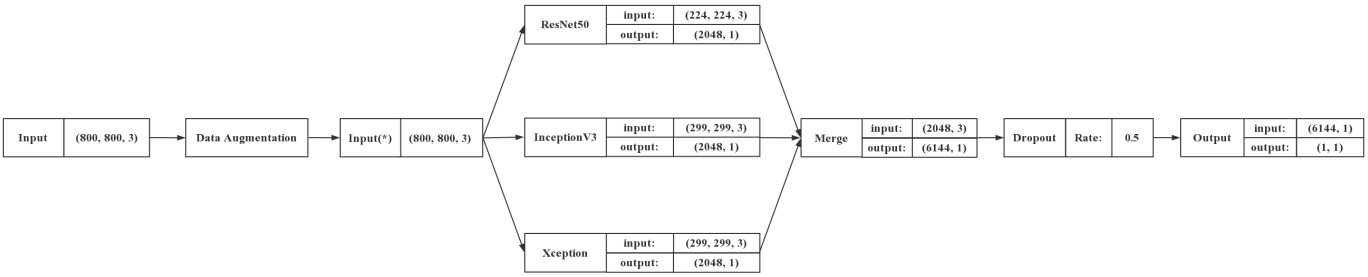

Fig. 1. The architecture of a binary classifier with multiple convolutional neural networks for one single clothing style recognition. The RGB clothing images will be received by the classifier as input, run through these CNNs (ResNet50, InceptionV3, Xception). Finally, the classifier will give an output with the value of either 1 or 0, representing if the clothing belong to the specific style or not.

III. Proposed Model

Since Alex Krizhevsky won the 2012 ImageNet Competition [4], their complex convolutional neural network AlexNet has been successfully applied to a larger variety of computer vision tasks (e.g., object-detection, segmentation, human pose estimation, video classification, and object tracking). These successes spurred a new line of research that focused on finding CNNs with higher performance. Recently, the quality of network architectures has been significantly improved by utilizing deeper and wider networks. As CNNs are efficient and effective [5], we use 13 CNN-based binary classifiers instead of traditional image processing methods [6] to recognize the clothing styles. Without loss of the generality, we show the architecture of a single binary classifier designed for one clothing style in Figure 1.

A. Input Layer

The input is an RGB clothing image with the size of $800 \times 800 \times 3$. It will be fed into the image generator for the data augmentation.

B. Data Augmentation

Fig. 2. A data augmentation example of images generated from the original image in *Style 2 Casual* with multiple image operations (i.e., horizontal flipping, rotating, and zooming). We perform the data augmentation on each of the positive samples 12 times.

Due to uneven distribution of positive and negative samples, it is necessary for us to implement data augmentation on the dataset. Data augmentation refers to generating batches of augmented images with multiple image operations (e.g., horizontal flipping, rotating, and zooming). Let N_{pi} be the number of positive samples and N_{ni} be the number of negative samples in *Style* i. We implement data augmentation s times on each positive image in *Style* i, where $s = \lceil \frac{N_{ni}}{N_{pi}} \rceil$. Then augmented image data can be put into the queue as the training set.

An example of the positive image generation in *Style 2 Casual* is shown in Figure 2. There are 4292 positive samples and 50616 negative samples in *Style 2 Casual*, so we implement data augmentation on these positive samples $\lceil \frac{50616}{4292} \rceil = 12$ times. Thus there will be $12 \times 4292 = 51504$ positive samples and 50616 negative samples put into the queue as the training set in *Style 2 Casual*.

In each style, we only implement data augmentation on its positive samples if $N_{pi} < N_{ni}$. In *Style 3 OL* and *Style 9 Ladylike*, there will be no data augmentation on the positive. The augmented images will be put into the CNN layer for feature extraction.

C. CNN Layer

Here we choose three pre-trained CNNs (ResNet50, InceptionV3, Xception) with weights pre-trained on ImageNet to constitute our CNN layer for clothing style recognition.

ResNet50 is a deep residual convolutional neural network and obtained excellent results on the ImageNet Classification Challenge [7]. The 152-layer residual net is the deepest network ever presented on ImageNet, while still having low complexity. The extremely deep representations also have excellent generalization performance on other recognition tasks such as detection and localization.

InceptionV3 is a much more complex convolutional neural network designed by Google [8]. It can be applied in big-data scenarios [9], especially where the huge amount of data need to be processed at reasonable cost and computational memory is inherently limited. Thus it can be an ideal choice of neural networks for clothing style recognition due to our limited computational capacity.

Xception proposes a method to replace Inception modules with depth-wise separable convolutions in neural computer vision architectures [10]. It actually has a similar parameter count as Inception V3, but small gains in classification per-

Algorithm 1 Adadelta Training Procedure

Require: $DecayRate\ \rho, Constant\ \epsilon, InitialParam\ \theta$
$Initialize\ accumulation\ variables\ E[g^2]_0 = E[\Delta\theta^2]_{0=0}$
for $t = 1 : T$ **do**
$Compute\ Gradients : g_t$
$Accumulate\ Gradient : E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2$
$Compute\ Update : \Delta\theta = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} \cdot g_t$
$Accumulate\ Updates : E[\Delta\theta^2]_t = \rho E[\Delta\theta^2]_{t-1} + (1-\rho)\Delta\theta^2$
$Apply\ Update : \theta_{t+1} = \theta_t + \Delta\theta_t$
end for

formance on the ImageNet dataset and large gains on the JFT dataset [11].

The pre-trained CNNs which don't include the fully-connected layers at the top of the network are suitable for feature extraction [12]. The images after data augmentation will be put into these CNNs without top fully-connected layers. Here we extract three feature vectors of ResNet50, InceptionV3 and Xception respectively, and merge them into one feature vector with the size of 6144×1.

D. Dropout

The next layer is dropout [13] with the rate of 0.5. When a large feedforward neural network is trained on a small training set, it typically performs poorly on test data. This overfitting can be reduced by randomly omitting half of the feature detectors on each training case. This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Each neuron learns to detect a feature that is generally helpful for predicting the correct answer given the large variety of internal contexts in which it must operate. Random dropout achieves obvious improvements on the task of clothing style recognition.

E. Output Layer

The output layer is a vector with the size of 1×1. Let h_t be the output of *Style* t. h_t is formulated as follows:

$$h_t = \begin{cases} 1 & s(x_t) \geq r \\ 0 & s(x_t) < r, \end{cases} \tag{1}$$

where x_t is the output of last layer in the t-th style. $s(\cdot)$ is sigmoid function, where the threshold $r = 0.5$. The value of the output vector is either 1 or 0, which tells if the clothing belong to the specific style or not. We get this binary classifier for one specific style. In the JD AI Fashion Challenge dataset, there are 13 clothing styles for us to recognize. As each binary classifier can only recognize one specific style, we need to train 13 classifiers for all 13 styles. The loss function we adopted is the binary cross entropy:

$$J(\theta) = -\sum_{i}^{N}(y_i \times log(z_i) + (1 - y_i) \times log(1 - z_i)) \tag{2}$$

The final output of our proposed model is a sequence of vectors $\mathbf{h} = \{h_1, h_2, ..., h_T\}$ with $h_t \in \{1, 0\}$ and $T = 13$ where T denotes the total number of styles.

IV. Experiments

A. Experimental Settings

Our system model is built with Keras [14]. Keras is a high-level neural networks API. The reason why we use Keras instead of Tensorflow is its high modularity and extensibility. Most importantly, it has these CNN models (ResNet50, InceptionV3, Xception, etc.) that are made available alongside pre-trained weights on ImageNet.

Moreover, Keras has its powerful image generator for image generation. We implement the generator for data augmentation. We put the training set into the generator, set up generation parameters (e.g., horizontal flipping, rotating, and zooming), and the image data generator can automatically generate batches of tensor image data with real-time data augmentation. Then it can put tensor image data into the queue for training.

Experiments are conducted over a workstation equipped with an NVIDIA GTX 1060 GPU. We divide 80% dataset into the training set and 20% into the validation set. At the same time, we train all the models with mini-batch gradient descent with Adadelta [15]. We use a held out validation set to monitor the training progress, and all the models are trained for 8 epochs. The Adadelta training procedure is shown in Algorithm 1.

B. Experimental Results

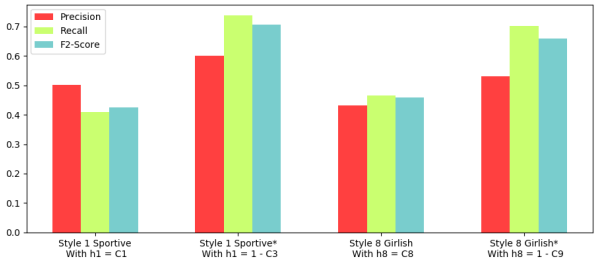

Fig. 3. The improvements we made on the F2-Scores in *Style 1 Sportive* and *Style 8 Girlish* with the method of mutual exclusion, compared with the original method.

We conduct experiments on both the validation set and the test set. The JD AI Fashion Challenge adopted average F2-Score in 13 styles as their evaluation. The F2-Score formula is as follows:

$$F_2Score = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times precision + recall} \tag{3}$$

where $\beta = 2$. According to the formula, recall has a much stronger influence on F2-Score than precision. So it is important to focus more on recall rate than precision rate. We train our model for each style and get 13 F2-scores.

However, due to few positive samples for training, even being data-augmented, the recall rates are much lower in Style

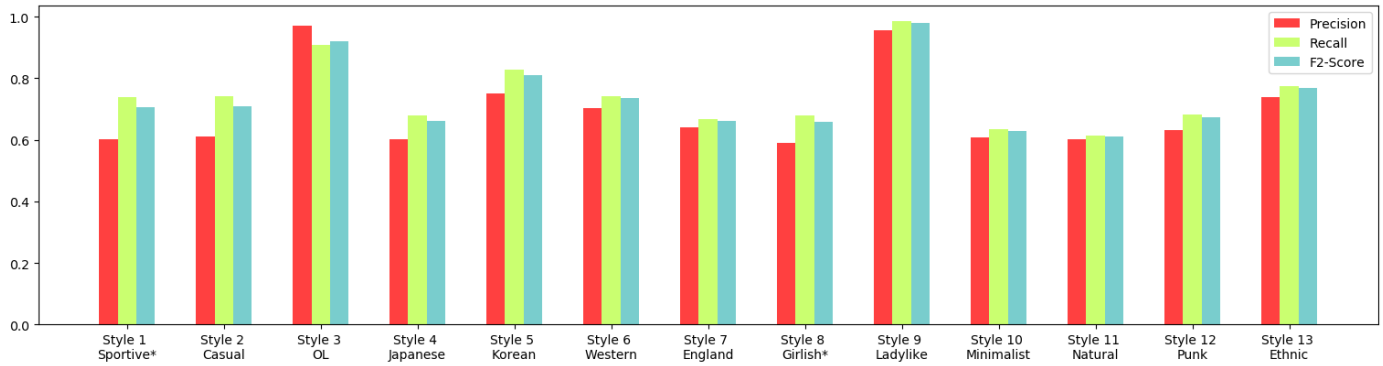

Fig. 4. The precisions, recalls and F2-Scores on the validation set. For the binary classifiers in *Style 1 Sportive* and *Style 8 Girlish*, we use the idea of mutual exclusion, and decide to adopt the reverse binary classifiers in *Style 3 OL* and *Style 9 Ladylike* to recognize *Style 1 Sportive* and *Style 8 Girlish*.

1 Sportive and Style 8 Girlish than in the others, which hurts the performance on the F2-Scores. We use the idea of mutual exclusion to solve this problem.

In fact, we find out that *Style 1 Sportive* is opposite to *Style 3 OL*, so is *Style 8 Girlish* to *Style 9 Ladylike*. We decide to adopt the reverse binary classifiers in *Style 3 OL* and *Style 9 Ladylike* to recognize *Style 1 Sportive* and *Style 8 Girlish*. Concretely, let us formulate the output given by classifier in *Style* i be C_i, and the final output be h_i. Thus $h_1 = 1 - C_3$ and $h_8 = 1 - C_9$, instead of $h_1 = C_1$ and $h_8 = C_8$. We get a better performance on the F2-Scores in these two styles with this method, as is shown in Figure 3.

The final precisions, recalls and F2-Scores on the validation set is shown in Figure 4. The average F2-Score on the validation set is **0.7319**. After test data is open, we submit our results on the test set, and the final average F2-Score is **0.5654**. The experimental results show the effectiveness and efficiency of our proposed CNN-based method with multiple binary classifiers.

V. Conclusion

In this paper, we proposed a CNN-based method with multiple binary classifiers for clothing style recognition. In particular, we trained 13 binary classifiers for 13 styles. Each binary classifier consists of multiple convolutional neural networks (ResNet50, InceptionV3, and Xception), which gives the output with the value of either 1 or 0, representing if the clothing belong to the specific style or not. In order to prevent from being affected by uneven distribution and make the training process more reasonable, we implement the data augmentation. We also employ the idea of the mutual exclusion with the opposite styles to help improve the F2-Scores in *Style 1 Sportive* and *Style 8 Girlish*.

Finally, the model consisting of the binary classifiers has an excellent performance in the JD AI Fashion Challenge and achieves the F2 scores of 0.7319 and 0.5654 for the validation and test dataset, respectively. The results verify the effectiveness of our proposed method in the clothing style recognation.

References

[1] Huizhong Chen, Andrew C Gallagher, and Bernd Girod. Describing clothing by semantic attributes. *european conference on computer vision*, pages 609–623, 2012.

[2] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. *international conference on computer vision*, pages 3343–3351, 2015.

[3] Lukas Bossard, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack, and Luc Van Gool. Apparel classification with style. *asian conference on computer vision*, pages 321–335, 2012.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[5] Yann Lecun, Koray Kavukcuoglu, and Clement Farabet. Convolutional networks and applications in vision. *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256, 2010.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *computer vision and pattern recognition*, pages 770–778, 2016.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *computer vision and pattern recognition*, 2016.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *international conference on machine learning*, pages 448–456, 2015.

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. *computer vision and pattern recognition*, 2017.

[11] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marcaurelio Ranzato, Andrew W Senior, Paul A Tucker, Ke Yang, et al. Large scale distributed deep networks. pages 1223–1231, 2012.

[12] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. Neural compatibility modeling with attentive knowledge distillation. *international acm sigir conference on research and development in information retrieval*, 2018.

[13] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv: Neural and Evolutionary Computing*, 2012.

[14] Franois Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[15] Matthew D Zeiler. Adadelta: An adaptive learning rate method. *arXiv: Learning*, 2012.