Applications of Deep Learning in Fashion Attributes Detection

Wenshan Wang, Xiuqi Shao, Jingyuan Bian
Columbia University
ww2468@columbia.edu, xs2327@columbia.edu, jb4076@columbia.edu

Abstract

Detecting detailed apparel attributes using deep learning methods gets increasing attentions. The goal of our project is to build an automatic fashion attributes detection system which can detect eight detailed apparel attributes such as collar design and skirt length. We applied deep learning techniques including transfer learning and fine tuning, used Resnet50 and InceptionV3 as our base model. Besides, image augmentations and other techniques are applied to improve our models. We trained and validated our models on images from a real e-Commerce platform, which makes the performance of our system even more trustworthy.

Keywords: component; attributes detection; deep learning; image augmentation

I. Introduction

Detecting detailed apparel attributes is a topic receiving increasing attentions, which also has wide applications. Recent year, the demands of online shopping for fashion items grow a lot, which raises problems such as the sellers provide information not consistent with the real stuff, different sellers have inconsistent understandings of apparel styles. An automatic fashion attributes detection system can help overcome these problems by providing precise and consistent taggings or descriptions of apparel from their pictures. This technique can be applied to various areas such as apparel image searching, navigating tagging, and mix-and-match recommendation, etc.

The hierarchical attributes tree (Figure 1) is constructed as a structured classification target to describe the cognitive process of apparel. In this paper, we designed an algorithm to recognize 8 major attributes of apparel images, which are skirt length, coat length, collar design, neck design, neckline design, pant length, sleeve length and lapel design.

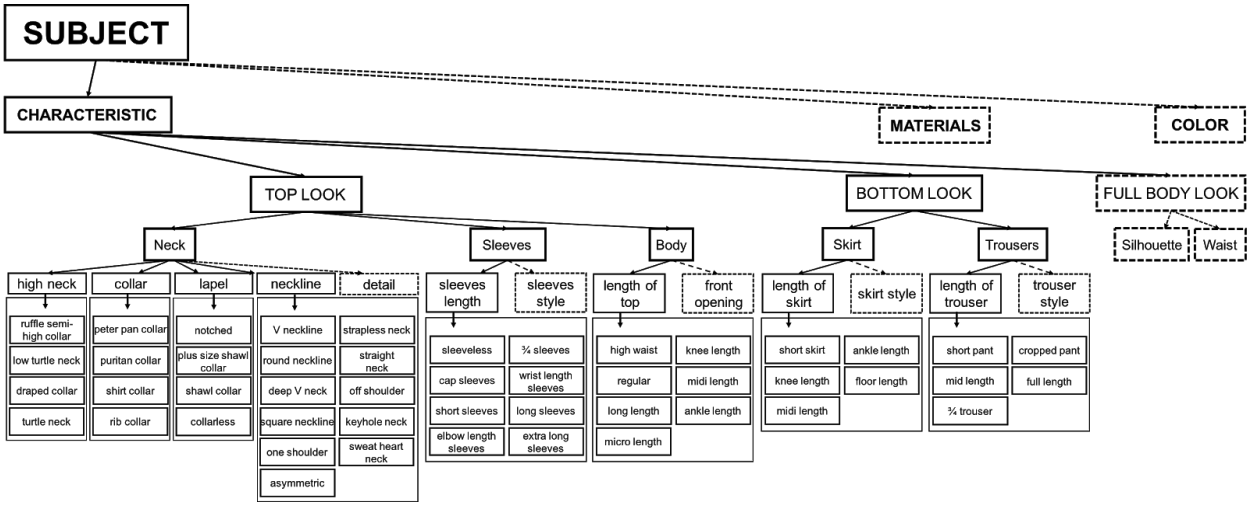


Figure 1

II. Related Works

In earlier times, non-convolutional-neural network approaches are used in the apparel attributes detection such as Markov Random Field [1] and SVM model [2]. With the rapid development of convolutional neural network and deep learning, great network architectures giving excellent performance on image classification areas show up, such as Deep Residual Networks (ResNets) [3] and the Inception Architecture [4]. Moreover, attributes detection topics develop from only detecting low-level attributes such as gender and clothing categories [5] to detecting high-level attributes such as collar presence and neckline shape [6]. In our work, we apply transfer learning using ResNets and Inception nets as base models, and focus on detailed apparel attributes such as sleeve length, neck design and collar design.

III. System Overview

3.1 Dataset description and modification

Dataset we use is from Alibaba 2018 FasionAI Global Challenge. It includes totally 79,573 images from Alibaba e-Commerce platform and labels describing 8 attribute dimensions which are neckline design, collar design, high neck design, lapel design, sleeves length, length of top, length of skirt, and length of trousers. They are separately given in 8 different folders so the each image only has one label from one of the 8 dimensions above. Table 1 shows the attribute values corresponding to attribute dimensions respectively.

Table1

Attribute Dimensions	Attribute Values
Sleeve length	Invisible, sleeveless, cap sleeves, short sleeves, elbow length sleeves, ¾ sleeves, wrist length sleeves, long sleeves, extra long sleeves
Skirt length	Invisible, short skirt, knee length, midi length, ankle length, floor length
Coat length	Invisible, high waist, regular, long length, micro length, knee length, midi length, ankle length
Pant length	Invisible, short pant, mid length, ¾ trouser, cropped pant, full length
Neck design	Invisible, ruffle semi-high collar, low turtleneck, draped collar, turtle neck
Collar design	Invisible, peter pan collar, puritan collar, shirt collar, rib collar
Lapel design	Invisible, notched, plus size shawl collar, shawl collar, collarless
Neckline design	Invisible, V neckline, round neckline, deep V neck, square neckline, one shoulder, asymmetric, strapless neck, straight neck, off shoulder, keyhole neck, sweetheart neck

We are not using the original images directly since their sizes vary from each other. We resize them to be 399 by 399 pixels, which is the largest size our memory can handle. Also, the 8 attribute dimensions are treated independently, and for each of them, we seperate 80% images as the training set, and the rest images as the test set.

3.2 Model architecture and implementation

We take into consideration that our test set photo may exist in a variety of conditions, such as different orientation, angle, scale, brightness. Involving image augmentation could our model more robust and invariant to translation, viewpoint, size or illumination. Therefore, we used the techniques provided in package “Imgaug” to implement image augmentation in both our training set and test set. The resized images will first go through heavy augmentations [7].

We use the same structure for all 8 networks which is show in figure 2. Input image is designed to be a 399 by 399 image with 3 color planes, it goes through a preprocessing layer in which it will be zero-centered for each color channel. Then the output goes into a base model. Here, we compare two deep neural networks ResNet50 and InceptionV3. The last layer is a fully-connected layer with softmax as the activation function. It takes the output from our base model as input where a 50% dropout is added to prevent overfitting. The final output is a number corresponding to a specific category in that attribute dimension.

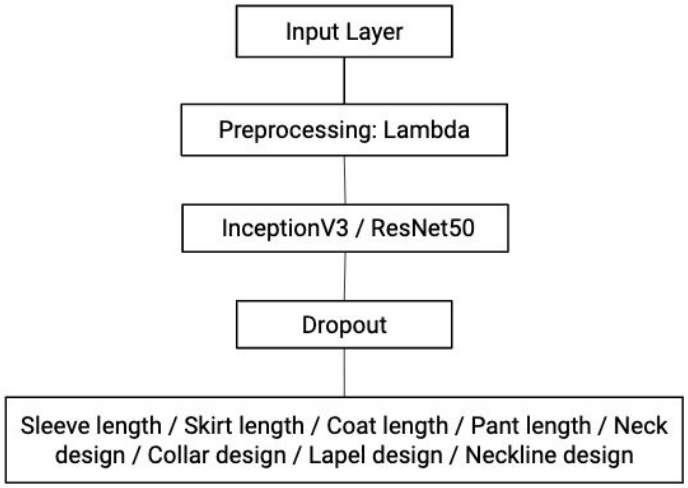


Figure 2

We use a combination of Adam and SGD optimizer with a mini-batch size of 16. We start from using Adam with a decaying learning rate. For most of our tasks, we start the learning rate from 0.0001, and divide it by 0.25 for about every 3 epoches. SGD is used at the last epoch with small learning rate to seek for a more precise result.

IV. Algorithm

4.1 Image Augmentation

As mentioned before, we applied heavy augmentations to our input images. To avoid the augmentations break images by changing them too much, we actually reduced the strength of some single augmentors, and decreased the probabilities of using some augmentors as well.
The detailed process of the heavy augmentations is shown as follows:

- 50% probability [scale, translate, rotate, shear]
- apply one of the 4 augmentors:
    - 10% probability [superpixel representation, blur]
    - [sharpen]
    - [emboss]
    - 10% probability [mark and overlay edges]
- [add gaussian noise]
- [dropout]
- 10% probability [invert]
- 50% probability [add value to each pixel]

- [change brightness, normalization, greyscale]
- 25% probability [distort local areas with varying strength]
- 25% probability [move pixels locally around]

Image augmentations are used to prevent our neural networks from learning irrelevant patterns, thus boosting overall performance. In this project, we find this method also very efficient in handling overfitting issues. Before we involved image augmentations, training accuracy was around 99.7% while testing accuracy were stuck under 82% even after we added dropout layer in the network. After image augmentations, training accuracy decreased to around 97% and testing accuracy increased to around 87%. Thus, we conclude that image augmentations help with prevent overfitting.

4.2 Transfer Learning plus Fine Tuning

The dataset that we have for training has approximately 10,000 photos per task, whose size might not be sufficient for us to train a deep learning image recognition model from scratch. Therefore, we decided to use transfer learning and fine tuning to complete the tasks. We used weights from a pre trained model as our initial weights, then added input, preprocessing, and dropout layer, and finally fine tuned the whole model using our fashion dataset. We also notice that, comparing with only fine tuning the last layer, fine tuning the whole model would largely increase the model performance.

The two pretrained models we used, ResNet50 and InceptionV3, were pre trained using ImageNet, which has 1,200,000 photos per 1000 classes. ResNet50 is a short name for 50 layers residual network. It has proved that training process using residual form is easier than training simple deep convolutional neural networks and also the problem of degrading accuracy is resolved. The Inception network is known for increasing the classification accuracy while controlling computation and parameter cost. Both of the pretrained weights could be downloaded using keras applications.

4.3 Other Techniques to Improve Accuracy and Efficiency

Besides image augmentation, transfer learning and fine tuning, there are several techniques that helped improving the model. First, increasing the size of pictures that model reads from training set. In our project, we choose width 399, the largest sizes that our memory can handle. The future work are suggested to increase the width, with higher computation ability and larger memory available. Second, comparing different pretrained models for each of the eight tasks and choosing the better could increase the final performance of the model. In our case, we compared VGG16, InceptionV3 and ResNet50, respectively. Thirdly, to improve the efficiency and shorten the training time cost, we start with adam using decaying learning rates to converge quickly and then finish with SGD (stochastic gradient descent) using a small learning rate to converge more precisely.

V. Experiment Results

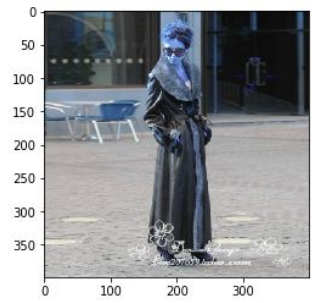
Table 2 has shown our training and test accuracy for all 8 models. As you could see that all of our 8 models have training accuracy above 94.5% and all of our test accuracy are above 85%.

Table 2

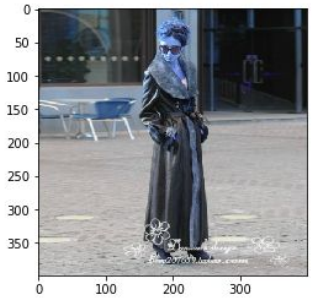
	Sleeve Length	Skirt Length	Coat Length	Pant Length	Neck Design	Collar Design	Lapel Design	Neckline Design
Training Accuracy	0.9655	0.9839	0.9895	0.9758	0.9626	0.9786	0.9641	0.9457
Test Accuracy	0.8992	0.8656	0.8825	0.8646	0.8632	0.8696	0.8927	0.8592

We could directly predict the attributes of 8 labels of the given picture, and results of an example from test set are shown as below:

```
The skirt length of the image below is Floor Length
 The coat length of the image below is Ankle&Floor Length
 The neck design of the image below is Invisible
 The collar design of the image below is Invisible
```




```
The lapel design of the image below is Plus Size Shawl
 The neckline design of the image below is Deep V Neckline
 The pant length of the image below is Invisible
 The sleeve length of the image below is Long Sleeves
```



VI. Conclusion

Detecting detailed apparel attributes is a time consuming and computationally expensive task. We used 8 CPU and 4 GPU with extended memory on google cloud platform, and training each model required half day. Using larger image size did improve the model performance a lot, however, it requires memory. Fine tuning the whole model gives our model a huge progress comparing to only tuning the last layer, but it requires more computational ability. Balance is needed between the performance of models and the

expense, and some techniques helped us a lot in the process. Image augmentations helped increase the generalization ability of our model and prevent the overfitting. Careful choice of optimization algorithms helped us reduce the training time.

It should be admitted that the performance of our models can still be improved given more computational ability and time. What is more, we noticed the complexity of the images from real e-Commerce platform. For example, models are using various poses, some standing, some lying, and the images are not always full-body shot, lower body shots widely used to show short skirts. Therefore, a possible way to improve the models' performance is to classify different types of images before they going to different deep neural networks.

References

[1] Anguelov, D., Lee, K., Gokturk, S.B., Sumengen, B.: Contextual identity recognition in personal photo albums. In: CVPR (2007)

[2] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In ECCV, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[5] Bourdev, L.,Maji, S., Malik, J.: Describing people: Poselet-based attribute classification. In: ICCV (2011)

[6] Chen H., Gallagher A., Girod B. (2012) Describing Clothing by Semantic Attributes. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C. (eds) Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7574. Springer, Berlin, Heidelberg

[7] https://imgaug.readthedocs.io/en/latest/source/examples_basics.html