DeepFashion Attribute Prediction Challenge Report

Xingxuan Li[1,2]
[1]Nanyang Technological University, Singapore [2]DAMO Academy, Alibaba Group
xingxuan001@ntu.edu.sg

Abstract

DeepFashion Attribution Prediction Challenge is to identify the attribute labels depicted in a fashion photograph. And the dataset used in this challenge is DeepFashion, which includes 6000 images. And the dataset is divided into 5000 images for training and 1000 for validations. By fine-tuning a pre-trained SE-ResNet-50 model with a 32×4d template on ImageNet, the model achieves 82.65 % accuracy on test set. The report details the techniques applied on model optimization and regularization, as well as a full analysis of all the experiments conducted on the challenge.

1. Introduction

DeepFashion dataset [4] (Figure.3) has in total 26 attributes that describe garments commonly. And the attributes are grouped into 6 major categories. Every single image has 6 attributes, one each from the 6 major categories (Table.1).

The challenge is to predict attributes of all 6 categories for each image. And the accuracy is calculated as the average accuracy across all attributes.

2. Method

Our goal is to accurately predict the attributes of all 6 categories. The challenge here is that different categories may have various learning behaviours. Categories which learn slower will converge slower, while other categories may converge faster and start to overfit. As a result, the overall accuracy will drop as well. In this section, we introduce all the methods and set of techniques we experiment on.

2.1. Data augmentation

Data augmentation is essential in this challenge. Models we select to use are all pre-trained on ImageNet [1], which has 14 million images. Comparing with 5000 training images in DeepFashion dataset, it would be very difficult to fine-tune any ImageNet pre-trained model. Furthermore, we are predicting on categories of garments. For most of the images, the essential information lie in part of the images. And the rest could affect the model negatively.

To address the problem of lack of training images, we adopt random horizontal flip to add more training images. By performing this technique, we can also prevent the models from learning irrelevant patterns. For example, in Figure.1, the two images have exactly the same attributes and prevent model making predictions based on the directions.

Generally, models we use in this competition take the

Figure 1. Example of random horizontal flip.

Figure 2. Example of cropping by bounding box.

Figure 3. Example images for selected attributes.

Category 1	Category 2	Category 3	Category 4	Category 5	Category 6
floral	long_sleeve	maxi_length	crew_neckline	denim	tight
graphic	short_sleeve	mini_length	v_neckline	chiffon	loose
striped	sleeveless	no_dress	square_neckline	cotton	conventional
embroidered			no_neckline	leather	
pleated				faux	
solid				knit	
pleated					

Table 1. 26 Attributes and 6 Categories

resolution of 224^2. Thus, we resize the images into required size. And we do not perform further cropping as the images already contain minimum essential information after we crop them by bounding box.

Last but not least, we normalize the images using:

$$\begin{cases} mean = (0.485, 0.456, 0.406) \\ std = (0.229, 0.224, 0.225) \end{cases} \quad (1)$$

Adopting the mean and standard deviation of ImageNet will prevent distribution shock during fine-tuning.

To extract essential information from the images, we crop the image by the bounding box provided in the dataset. For example, in Figure.2, only the part in the red box is cropped and feed into the model.

2.2. Model selection

Firstly, we adopt ResNet-50 [2] as our baseline model, which is a variant of ResNet. ResNet is immensely successful in image classification task. While making use of residual blocks to improve the accuracy of the models, the skip connection (Figure.4) allows the model to be trained very deep. And it ensures the higher layers of the model do not perform any worse than the lower layers at the same time. ResNet-50 is simply a 50-layer-ResNet. And it is build with 3-layer bottleneck blocks (Figure.4). We take the 1000-dim final output layer of ResNet-50 as a feature representation of the images. And we attach 6 classifier heads to the representation layer, which indicates the 6 categories. The dimension of each classifier matches the number of attributes under the corresponding category. To learn further representation from ResNet-50, we also experiment on adding fully connected layers in between the model and the classifiers. We will analyse the result in the next section.

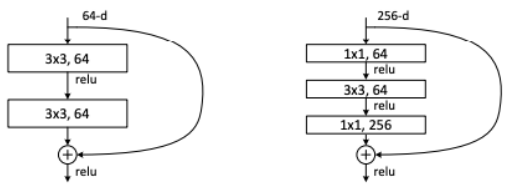

Figure 4. Building block. Left: ResNet-34. Right: ResNet-50

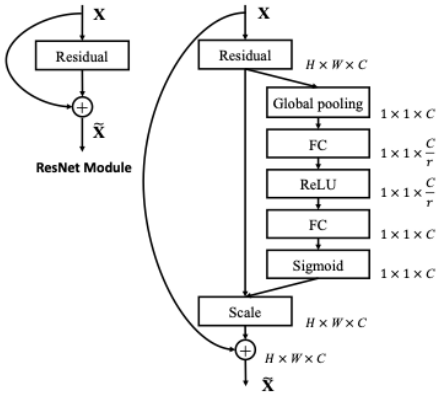

Figure 5. Building block of SE-ResNet.

SE-Net [3] uses Squeeze-and-Excitation (SE) block that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. As shown in Figure.5, SE-ResNet adds a SE path in between comparing with ResNet. The squeeze part can be interpreted as a collection of the local descriptors whose statistics are expressive for the whole image. And it aims to squeeze global spatial information into a channel descriptor. Thus, channel-wise statistics is achieved by using global average pooling. Followed by 2 FC layers and activation functions, channel-wise dependencies are fully captured by the excitation operation. In this challenge, we adopt SE-ResNet-50 for experiment. And different number of FC layers before the classifiers are also tested. Result will be shown in the next section.

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{i=1}^{W} u_c(i,j) \tag{2}$$

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})) \tag{3}$$

$$\frac{2}{r} \sum_{s=1}^{S} N_s \cdot C_s^2 \tag{4}$$

$$\mathbf{x}_c = \mathbf{F}_{scales}(\mathbf{u}_c, s_c) = s_c \cdot \mathbf{u}_c \tag{5}$$

Furthermore, we also experiment on SE-ResNeXt-50 [3] model with a 32×4d template. ResNeXt [5] adds the next dimension on top of the ResNet, which is called the cardinality dimension. As shown in Figure.7, the dimension of cardinality controls the number of more complex paths. And a nonlinear function is performed for each path.

2.3. Loss function

Initially, we use cross-entropy loss as our loss function. For one image, each classifier has an cross-entropy loss, and

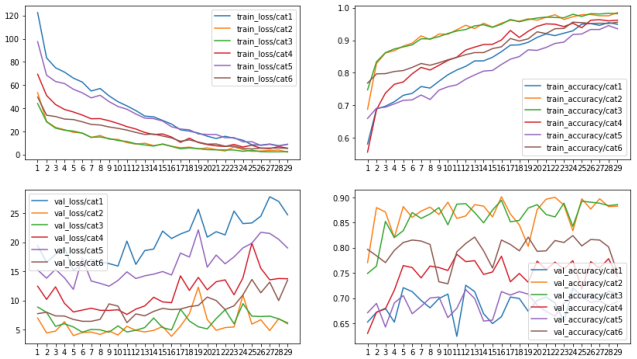

Figure 6. Baseline experiment on ResNet-50 with 0 FC layers and focal loss. Upper left: training loss. Upper right: training accuracy. Lower left: validation loss. Lower right: validation accuracy.

then we sum all 6 categories to form the total loss of this image. And the equation looks like this:

$$Loss = -\sum_{i=1}^{6} \sum_{j=1}^{N_i} y_{i,j} \cdot \log y_{i,j} \tag{6}$$

However, cross-entropy loss does not address the class imbalance problem. And we use focal loss instead to help the model learn better.

2.4. Others

We also perform hyper-parameter tuning during experiments. For example, γ in focal loss, λ the learning rate of the optimizer, etc. We will detail the result in the next section.

3. Experimental Analysis

In this section, we document the experiments that we conducted on multiple models and techniques, as well as findings and analysis of the result.

3.1. ResNet-50

Using ResNet-50 as a baseline, we would like to evaluate whether focal loss works well on each category. We first

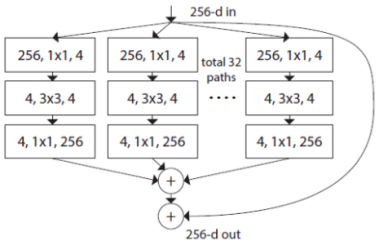

Figure 7. Building block of SE-ResNet.

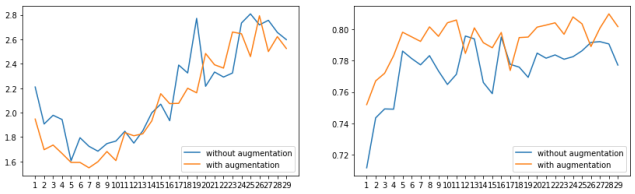

Figure 8. Data augmentation experiment on ResNet-50 with 0 FC layers and focal loss. Left: validation loss. Right: validation accuracy.

fine-tune the pretrained ResNet-50 without adding any FC layers before the classifier head. As we expect, different categories have different learning behaviors (Figure.6). For example, category 5 learns much slower than category 2 and category 3. The model is more difficult to learn the material of the garments than the sleeve style. To address the issue, we later experiment on two different models.

Another finding is that on validation set, loss reaches minimum at around 10th layer and the accuracies of all classes are not improved significantly. This is because ResNet-50 is pretrained ImageNet. And compared with our 5000 training images, it is difficult to finetune the model. To increase the training set size, we adopted random horizontal flip. As shown in Figure.8, the average validation loss and accuracy have been improved in general. We believe with more augmentation techniques, the performance can still be boosted.

3.2. SE-ResNet-50

As we mentioned above, simple ResNet-50 fails to predict categories such as material accurately. In another word, ResNet-50 does not learn local information well. As a result, we experiment on SE-ResNet-50. With the squeeze and excitation blocks, the model can better extract details from the image, which is extremely useful for categories such as material. Because the essential information lies in the details rather than global information such as color or shape. As shown in Figure.9, for each category, the highest accuracy of SE-ResNet-50 outperforms ResNet-50, which proves our interpretation.

3.3. SE-ResNeXt-50

In the original paper of ResNeXt, increasing cardinality achieves better performance than ResNet. However, in our experiment, SE-ResNeXt-50 has worse performance than SE-ResNet-50, as shown in Figure.10. This is because of the number of parameters of SE-ResNeXt is more than SE-ResNet. And it makes it more difficult to finetune the pretrained model.

3.4. Fully connected layers

In order to experiment on the effects of number of FC layers, we conduct experiments for 0, 2, 4 FC layers on all three models. And interestingly, for all three models, more FC layers give worse results (Table.2). This is still due to the number of training images. It is hard for the model to learn a new dimension of representation with such few images.

3.5. Finetune loss function

Last but not least, based on the learning behavior we observe from Figure.7, we believe that the learning speed of each category is different. To adjust the speed, we add a weight to the loss of each category:

$$Loss = \sum_{i=1}^{6} w_i \cdot L_{focal} \tag{7}$$

And the set of weights can be finetuned as hyperparameters. Here we directly set the weights to be (0.95, 0.85, 0.85, 0.9, 1, 0.9) for respective six categories based on experience. And the model achieves best result on testset, which is 0.8265.

4. Conclusion

In conclusion, we use ResNet-50, SE-ResNet-50 and SE-ResNeXt-50 as pretrained models. By finetuning these models with DeepFashion dataset, we aim to solve the multi-label challenge. And SE-ResNet-50 with 0 FC layers

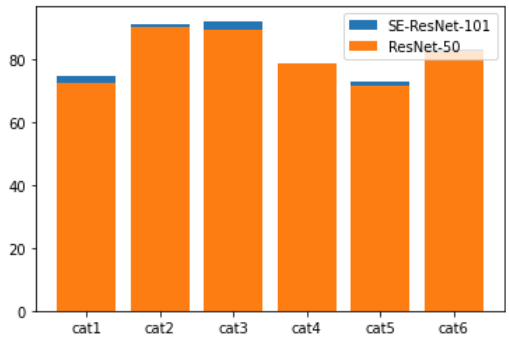

Figure 9. Validation accuracy for each category. (SE-ResNet-50 with 0 FC layers and focal loss vs ResNet-50 with 0 FC layers and focal loss.)

# FC layers	ResNet-50	SE-ResNet-50	SE-ResNeXt
0	0.7985	**0.8235**	0.7964
2	0.7978	0.8172	0.7918
4	0.7743	0.8138	0.7717

Table 2. Experiments on three models with different number of FC layers. Results on the test set.

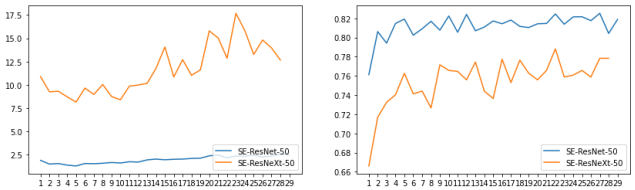

Figure 10. Comparison between SE-ResNet-50 and SE-ResNeXt-50. Left: validation loss. Right: validation accuracy.

G2104240G	43	10/09/21	0.82650 (22)

Figure 11. Final result on test dataset from CodaLab.

and weighted focal loss achieves the best result of 0.8265 (Figure.11).

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255. Ieee, 2009. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2

[3] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 3

[4] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016. 1

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 3