Clothing Image Detection and Recognition Based on Faster R-CNN

Shuaifei Ji¹, Runping Han²*, Jianfeng Wei³ and Rui Wang⁴
¹,²Chinese Fashion Science & Technology Research Institute, Beijing Institute of Fashion Technology, Beijing, China
³,⁴School of Information Engineering, Beijing Institute of Fashion Technology, Beijing, China

*Corresponding author

Abstract. Clothing image detection and recognition algorithm is proposed in this paper, which involves the design of two neural network models based on improved Faster R-CNN [1]. In our work, in order to train neural networks, ModaNet [2] dataset is selected and improvements on it are made, that is, the improper bounding boxes of footwear and boots in ModaNet dataset are modified based on their polygon annotations. Moreover, by making the following improvements on Faster R-CNN, our two neural network models are established, which use VGG16[3] and the first 40 layers of ResNet50 [4] as feature extraction network, respectively. When using ResNet50 for feature extraction, the last 10 layers of ResNet50 are selected as part of the final classification and regression network. In addition, the softmax activation function in RPN is replaced with sigmoid. Through experimental comparison, it is found that both of these two models can well accomplish the task.

1. Introduction
Sketch retrieval [5] has many important applications in the field of computer vision. One of the applications is end-to-end detection and retrieval of clothing image based on fashion illustration. Our team is working on the project that involves this subject. This project results can facilitate the work of fashion designers and provide them with inspiration. As one part of this project, the detection and recognition of clothing image, which is named as fashion image and from e-commerce sites, is studied in this paper.

Fashion images have the following characteristics. Usually, these images are street photos with complicated background; each photo often contains more than one item of clothing; clothing items in the images are often worn by fashion model and deformed because they are not rigid. Due to its above characteristics, clothing image detection and recognition remains a difficult problem. A great quantity of work should be done to achieve higher detection and recognition accuracy.

The main work and innovations of this paper are as follows. Preliminary investigations of clothing image detection and recognition are carried out. Before using ModaNet dataset to train our neural network models, we correct the improper bounding boxes of footwear and boots in fashion images in ModaNet dataset, and make them more accurate. Many targeted improvements on Faster R-CNN are made to design two neural network models which are used for the detection and recognition of clothing image. Tests for evaluating two models are conducted, and test results are analysed.
The arrangement of this paper is as follows. The first section outlines the research background and work of this paper. The second section thoroughly describes Faster R-CNN and the design of our neural network models which are obtained by adjusting faster R-CNN’s structure, loss function, activation function and so on for clothing image detection and recognition. The third section introduces the modification of ModaNet dataset, two models’ training and testing, and test results analysis. The last section concludes the work of this paper and makes a plan for the future research.

2. Clothing Image Detection and Recognition Algorithm
The architecture of clothing image detection and recognition algorithm based on improved Faster R-CNN is shown in figure 1. It consists of feature extraction network, RPN network, ROI Pooling layer and final classification and regression network.

2.1. Feature Extraction
The original image with size $P \times Q$ is scaled to input image with size $M \times N$, and then it is sent into pre-trained ResNet50 or VGG16 to obtain shared feature maps with size $w \times h$. To get larger size of feature maps, which contain more information, VGG16 does not use full connection layers and the last Max Pooling layer, and ResNet50 does not use the last 10 layers. When extracting features with VGG16, input images will be passed through 4 max pooling layers, the stride of each max pooling layer is 2, and the core size is $2 \times 2$. Finally, 512 feature maps of size $\frac{M}{16} \times \frac{N}{16}$ are obtained.

2.2. Region Proposal Network (RPN)
The feature maps are fed into RPN, which can generate proposal regions.

2.2.1. Fully convolutional network (FCN). Feature maps are fed into fully convolutional network (FCN) [6], whose architecture is shown in figure 2. A convolutional layer rpn_conv with kernel size of $3 \times 3$ and relu activation is connected by two parallel convolutional layers rpn_conv_cls and rpn_conv_bbx with kernel size of $1 \times 1$ and relu activation. To preserve the size of input and output feature maps unchanged, the padding method of the three convolutional layers is ‘same’ and the stride is 1. The channel size of branch rpn_conv_cls is 9 and the activation function is sigmoid. This branch estimates whether an anchor is foreground or background. The channel size of branch rpn_conv_bbx is 36 and the activation function is linear. This branch makes regression correction to the anchors to make them closer to the ground truth boxes. When extract features with ResNet50, the number of feature maps changes from 1024 to 512 through rpn_conv layer. When extract features with VGG16, the number of feature maps changes from 512 to 256.
2.2.2. Anchor generation. The sizes of the red, blue and green squares in figure 3 are 8×8, 16×16 and 32×32 respectively. Each scale square is transformed according to the aspect ratio of 0.5, 1 and 2 to get 9 different anchors. Each pixel in the feature map generates 9 anchors. Then these anchors are copied to the input image at the same interval. The sizes of these anchors are scaled by $\frac{M}{W}$. $M$ is the width of the input image, and $W$ is the width of the feature map. The anchor boxes are dropped if their boundaries beyond the $M \times N$ input image. If the IoU of a remaining anchor box with the ground truth bounding box is greater than 0.7, the anchor box is labeled foreground anchors. Anchors with IoU less than 0.3 are labeled background anchors. Other anchors are labeled neutral anchors, which do not participate in training. As shown in figure 4, IoU is the ratio of the intersection of two bounding boxes to their union.

2.2.3. Bounding box regression coefficient. Bounding box regression coefficients are also known as regression targets. The red box in figure 5 is an anchor box, and its upper-left coordinates are $(O_x, O_y)$. The width and height are $O_w$ and $O_h$ respectively. The blue box is a ground truth bounding box, whose upper-left coordinates are $(T_x, T_y)$. Width and height are $T_w$ and $T_h$ respectively. Regression target is the vector $t_i^*$.

$$t_i^* = (t_x^*, t_y^*, t_w^*, t_h^*)$$  (1)
2.2.4. Anchors transformation. The branch of rpn_conv_bbx in RPN predicts vector $t_i$. The better the network trained, the closer the predicted vector $t_i$ is to the regression target $t_i^*$. $t_i$ is used for foreground anchor’s translation and scaling to obtain the predicted anchor boxes, while the background anchors do not need to be transformed. The boxes that don’t make sense are dropped, whose lower-right coordinate is smaller than upper-right coordinate or boundaries beyond the $M \times N$ input image. Due to the uncertainty of $t_i$, the predicted anchor boxes may still contain foreground or background. In addition, there are still a large amount of overlaps between the boxes, which require further selecting.

2.2.5. Proposal layer. Non-maximum suppression (NMS) [7] is used in proposal layer to further select the predicted anchor boxes. Sort the probabilities (between 0 and 1) of predicted anchor boxes in descending order, which indicates an anchor box is more likely to be foreground or background. Take the predicted anchor box with the highest probability. Eliminate its nearby predicted anchor boxes when the IoU of them is greater than 0.9, that is, suppress the nearby boxes with too large overlap area. Repeat this operation to get more proposal boxes, and finally select up to 300 proposal boxes. Calculate the IoU of each proposal box and each ground truth box, and discard the proposal boxes with IoU less than 0.1, because these boxes have no value for training classification and regression network. If the IoU of a proposal box and the corresponding ground truth box is greater than 0.5, record its category label (13 categories). If the IoU is between 0.1 and 0.5, its category label is background. Randomly select $s$ ($s \leq 16$) boxes from foreground proposal boxes, and randomly select $(32 - s)$ boxes from background proposal boxes. Send the 32 proposal boxes into ROI Pooling layer, and the corresponding labels will be used to train the final classification and regression network.

2.3. Classification and Regression Network

Put the 32 proposal boxes in the corresponding location of feature maps, cropping out these regions from the feature maps to get the proposal regions. The proposal regions will be processed in ROI Pooling layer and converted into proposal feature maps with size $7 \times 7$.

![Figure 6. Classification and regression network (when extracting features with VGG16).](image)

When extracting features by VGG16, the architecture of classification and regression network is shown in figure 6. The 512 feature maps with size $7 \times 7$ are flattened to a one-dimensional array with a length of 25088. It is then fed into two full connection layers, each of which has 4096 nodes. The layers are followed by the classification branch fc_cls and the regression branch fc_bbx in parallel. The branch fc_cls is a full connection layer that has 14 nodes and uses softmax activation to predict the categories of proposal regions (14 categories, including background). The branch fc_bbx is a full connection layer that has 52 nodes and uses linear activation to predict regression coefficients.

\[
\begin{align*}
    t_i^*: & \quad \left\{ 
    t_i^x = (T_x - O_x) / O_w, \\
    t_i^y = (T_y - O_y) / O_h, \\
    t_i^w = \log(T_w / O_w), \\
    t_i^h = \log(T_h / O_h) 
    \right. 
\end{align*}
\]
When extracting features with ResNet50, the architecture of classification and regression network is shown in figure 7. Conv is convolutional layer; BN is batch normalization; Act_t is the tth activation layer of ResNet50; ADD represents the addition of feature maps; Flatten represents the transformation from multi-channel feature maps to one-dimension array; Avg_pooling represents average pooling; The branch fc_cls is a full connection layer that has 14 nodes and uses softmax to predict the categories of proposal regions. The branch fc_bbx is a full connection layer that has 52 nodes and uses linear activation function to predict the regression coefficients of foreground proposal regions.

2.4. Loss function

The loss function of RPN is defined as follows.

\[
L(p_i, t_i) = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum_{i=1}^{N_{reg}} p_i^* L_{reg}(t_i, t_i^*)
\]  

(3)

\(i\): the index of an anchor in a batch.

\(p_i\): the probability of anchor \(i\) being a foreground anchor.

\(p_i^*\): the ground truth label, is 0 or 1.

\(t_i\): a vector representing the transformation from a foreground anchor to the predicted bounding box.

\[
t_i = (t_x, t_y, t_w, t_h)
\]

(4)

\(t_i^*\): represents the transformation from a foreground anchor to the ground truth bounding box.

\(L_{cls}\): log loss of the two classes (foreground anchor and background anchor).

\[
L_{cls}(p_i, p_i^*) = -[p_i^* \log p_i + (1 - p_i^*) \log(1 - p_i)]
\]

(5)

\(L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)\)

\(R(\cdot)\): robust loss function (smooth\(_L_1\)):

\[
\text{smooth\(_L_1\)}(x) = \begin{cases} 
0.5x^2 & \text{if } |x| < 1 \\
|x| - 0.5 & \text{otherwise}
\end{cases}
\]

(6)

N\(_{cls}\): batch size.

N\(_{reg}\): the number of anchors in a batch.

\(\lambda\): balancing parameter for adjusting the weights of the two losses, we set \(\lambda = 10\) by default.

For the final classification and regression network, it is needed to replace the \(L_{cls}(p_i, p_i^*)\) with the cross entropy loss \(H(p_i, p_i^*)\). \(p_i^*\) is one-hot vector of ground truth label; \(p_i\) is predicted vector.
\[ H(p_i, p_t) = -p_i \log p_t \] (7)

3. Experiments and Analysis

The experimental platform adopts Windows Server 2012 operating system, Keras deep learning framework, 128GB of computer memory and Nvidia GeForce GTX1080Ti GPU.

3.1. Experimental Data Set

![Figure 8](image1.png)

**Figure 8.** Samples before and after the bounding box of boots and footwear is corrected. The first row shows the images with original annotations from ModaNet dataset. The second shows the revised annotations.

ModaNet street fashion dataset is used for the experiments, which has annotations of bounding boxes and polygons of 13 clothing categories. The original dataset contains 52377 annotated images, most of which contain more than one piece of clothing. The bounding boxes of footwear and boots in the original dataset have many errors, which will affect the training of the network. Therefore, we first correct the position of the bounding boxes (as shown in figure 8) according to the polygon annotations, and then eliminate the images that failed to correct. Finally, 52254 images with correct annotations are obtained. The number of images and bounding boxes for each category are shown in figure 9.

![Figure 9](image2.png)

**Figure 9.** The number of images and instances per category
3.2. Experimental Results and Analysis

There are many evaluation indexes for the performance of neural network models. In equation (8) and (9), \( P \) is precision. \( R \) is recall. \( TP \) (true positive) is the number of positive samples that are predicted to be positive. \( FP \) (false positive) is the number of negative samples that are predicted to be positive. \( FN \) (false negative) is the number of positive samples that are predicted to be negative. In equation (10), \( AP \) is the average precision of a certain category. \( P_n \) and \( R_n \) are the values of \( P \) and \( R \) at \( n \)th threshold respectively. In equation (11), \( C \) is the number of all categories and \( mAP \) is the mean average precision of all categories.

\[
P = \frac{TP}{(TP + FP)} = \frac{TP}{all\ detections}
\]

\[
R = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths}
\]

\[
AP = \sum_n (R_n - R_{n-1})P_n
\]

\[
mAP = \frac{1}{C} \sum AP
\]

To realize clothing image detection and recognition, the modified ModaNet dataset is adopted to train the network that extract features with pre-trained VGG16 model or ResNet50 model. To get more accurate classification and regression results, fine tuning of the pre-trained models is conducted. When using VGG16, the pre-trained model weights are loaded into feature extraction network. The weights are updated in the first 150 epochs and are frozen during the last 50 epochs. When using ResNet50, the pre-trained model weights are loaded into feature extraction network and the final classification and regression network. The weights are updated in the first 80 epochs and are frozen during the last 20 epochs. The new weights are saved only if the training loss is lower than the previous best loss. Finally, we obtained two models, one is Faster R-CNN model with VGG16 backbone, the other is Faster R-CNN model with ResNet backbone. The detection results of the model with ResNet50 backbone are shown in figure 10.

---

**Figure 10.** Samples of ground truth and detection results. The left part shows the groundtruth bounding boxes. The right part shows the corresponding detection results of the Faster R-CNN model with ResNet 50 backbone.
In our experiments, two models are evaluated respectively. Table 1 shows the AP for each category. All categories’ mAP reaches 0.79 when using ResNet50 to extract features, and all categories’ mAP reaches 0.78 when using VGG16 to extract features.

**Table 1.** The comparison of AP for each category and mAP for all categories when extracting features with VGG16 or ResNet50, respectively.

| Category | AP (VGG16) | AP (ResNet50) | Category | AP (VGG16) | AP (ResNet50) | Category | AP (VGG16) | AP (ResNet50) |
|----------|------------|---------------|----------|------------|---------------|----------|------------|---------------|
| bag      | 0.76       | 0.77          | dress    | 0.77       | 0.84          | headwear | 0.82       | 0.83          |
| belt     | 0.58       | 0.57          | pants    | 0.91       | 0.92          | scarf/tie | 0.73       | 0.64          |
| boots    | 0.80       | 0.70          | top      | 0.74       | 0.81          | sunglasses | 0.79   | 0.82          |
| footwear | 0.78       | 0.75          | shorts   | 0.85       | 0.91          |          |            |               |
| outer    | 0.87       | 0.91          | skirt    | 0.78       | 0.83          |          |            |               |

It can be found that belt has more occlusion, it is not rigid and its instance size is small, so it has the lowest AP. Pants have larger instance size and less occlusion, so they have the highest AP. The sunglasses are small, but they are rigid, so they have a relatively high AP. The smaller the instance size, the smaller its corresponding proposal regions in feature maps, which contain less valuable information. Therefore, it is difficult for the classifier to classify small proposal regions.

**4. Conclusion**

In this paper, improved faster R-CNN is proposed in order to accomplish the task of clothing image detection and recognition. Before experiment, the improper bounding boxes of footwear and boots in ModaNet dataset are corrected. VGG16 and ResNet50 are adopted as the backbone network, and the network structure, activation function and loss function are changed accordingly. Finally, two models are obtained, it is found that the performance difference between the models is not significant.

In future work, fashion illustration dataset will be prepared. The neural network will be designed to implement the project of clothing image detection and retrieval based on fashion illustration.

**Acknowledgments**

This work is supported by Research Center of Beijing Digitalization Project in Fashion Industry and Beijing Institute of Fashion Technology (Grant No. NHFZ20190153/013).

**References**

[1] Ren, S, He, K, Girshick, R, & Sun, J 2015 Faster r-cnn: Towards real-time object detection with region proposal networks In Advances in neural information processing systems pp 91-99
[2] Zheng, S, Yang, F, Kiapour, M H, & Piramuthu, R 2018 Modanet: A large-scale street fashion dataset with polygon annotations In 2018 ACM Multimedia Conference on Multimedia Conference pp 1670-1678
[3] Simonyan, K, & Zisserman, A 2014 Very deep convolutional networks for large-scale image recognition arXiv preprint arXiv:1409.1556.
[4] He, K, Zhang, X, Ren, S, & Sun, J 2016 Deep residual learning for image recognition In Proceedings of the IEEE conference on computer vision and pattern recognition pp 770-778
[5] Xu, P, Yin, Q, Huang, Y, Song, Y Z, Ma, Z, Wang, L, ... & Guo, J 2018 Cross-modal subspace learning for fine-grained sketch-based image retrieval Neurocomputing 278, 75-86
[6] Long, J, Shelhamer, E, & Darrell, T 2015 Fully convolutional networks for semantic segmentation In Proceedings of the IEEE conference on computer vision and pattern recognition pp 3431-3440
[7] Neubeck, A, & Van Gool, L 2006 Efficient non-maximum suppression In 18th International Conference on Pattern Recognition (ICPR’06) (Vol 3) pp 850-855