Pedestrians Detection of Offline Retail Stores Based on Computer Vision

Mingxu Wang*
International School, Beijing University of Posts and Telecommunications, Beijing 100876, China
*Corresponding author e-mail: wangmingxu@bupt.edu.cn

Abstract. Pedestrian Detection is a sub-task in the object detection due to it plays an important role in video surveillance. This project train model algorithm by using offline retail data and intelligent identification based on the offline retail brand store scene and focuses on the two core personnel of the retail scene—shopping guides and customers. This project detects pedestrian’s localization and their key points by using deep learning and computer vision technologies.

1. Introduction
This paper is based on the scene of offline retail store, intelligent identification and analysis of the two core personnel, shopping guide and customers. By using computer vision and deep learning technologies to locate and detect pedestrians’ localization and their key points.

This paper achieved a two-branch network by using pose estimation. On the one hand, I use finetune the pre-trained object detection model [1] to locate pedestrian and classification model to predict the pedestrian’s bounding box. On the other hand, I use pose estimation to detect the key points of each pedestrian.

At present, there are already many new retail solutions on the market, but most of them focus on applications such as passenger flow analysis statistics and user accurate portraits. In response to the industry’s pain points and innovative application of blue ocean, improve the operation efficiency of brand stores and store management efficiency, optimize operational marketing strategies, improve sales conversion rate, improve service efficiency and consumer experience, etc. to help upgrade the retail industry.

2. Related Work
2.1. Foreign and domestic research situation
The current object detection models are mainly divided into two types, one-stage and two-stage. On the one hand, one-stage object detection models, which has easier network structure and progress faster, but usually with less accuracy, directly predict the bounding boxes of target and the classification of detected object. On the other hand, two-stage models first predicts the region of interest (ROI) and classifies what the object is the foreground or background, and then refinedly predicted bounding box and classifies the target. Thus, two-stage models usually have better accuracy but less compute efficiency. In the consideration of accuracy, I finally chose the second-order network.
2.2. Deep Neural Network
With the development of deep learning, some well-known basic network in the deep neural network, which can well extract high-level Semantics feature, can be used for the pedestrian attribute recognition task.

ResNet (Residual Neural Network) [2] was proposed by four Chinese from Kaiming He of Microsoft Research Institute. Through the use of ResNet Unit, the 152-layer neural network was successfully trained and won the championship in ILSVRC2015. The error rate on top5 was 3.57%. At the same time, the parameter quantity is lower than VGGNet[3], and the effect is very outstanding. The structure of ResNet can accelerate the training of neural networks very quickly, and the accuracy of the model is also greatly improved.

The main idea of ResNet is to add a direct connection channel, as shown in Figure 2, to the network, which is the idea of the Highway Network. The previous network structure was a non-linear transformation of the performance inputs, while the Highway Network allowed a certain percentage of the output of the previous network layer to be preserved. The idea of ResNet is very similar to that of the Highway Network, allowing raw input to pass directly to subsequent layers.

![Figure 1. A building block of Residual Network](image)

2.3. Transfer Learning
The most common example of image recognition is training a neural network. To identify different breeds of cats, if you start training from scratch, you need millions of levels of labeled data, massive graphics resources. If you use transfer learning, you can use Google's network of mature items such as Inception or VGG16, only train the last softmax layer, you only need thousands of pictures, you can use ordinary CPU, and the model is accurate. Sexuality is not bad. When using transfer learning, it should be noted that the neural network that is pre-trained should have little difference from the current task, otherwise the effect of transfer learning will be poor.

Compared with traditional methods, transfer learning can help to achieve the learning of multi-task objectives. Traditional machine learning needs to train multiple different models when facing different types of tasks.[1] While, with the development of transfer learning, we can first implement simple tasks, and apply the knowledge gained in simple tasks to more difficult problems in order to solve the problem what having fewer high-quality training data.

3. Methods

3.1. Network Infrastructure
This paper use the Faster R-CNN model as the base network, and add multi-task loss on the base network, so as to achieve the positioning task and multi-attribute classification task for pedestrians in the retail scene. This paper use the basic network such as VGG-net and fine-turning on the BOT data in order to get useful feature information. Therefore, this paper modifies the classification loss, which is the two-class loss for "Pedestrian" and "Background", on the second stage of the Faster R-CNN, and uses VGG-net as the basic feature extraction network to extract high-level feature information for my task.
The final network is implemented as Table 1. This paper name the first five conv layers as head model. I fixed the weights of conv1 and conv2 layers, because the first two convolution layers of network usually learn some edge, color and corner information. The goal of supervised learning is to extract as much of the feature information as possible according to the target task. And the higher relevant information is extracted, the higher performance of the model will be get (Rasmus et al., 2015a) [9]. And the weights of conv1 and conv2 layers can extract these features well for the different task. Besides, stacked 3x3 filters can capture larger receptive field and use less computing resources. And the next three convolutional layers are fine-tune when training in order to extract task-relevant features.

R-CNN module is implemented by two stacked fully connection layers (fc6 and c7) which both contain 4096 neurons and followed by dropout layer with 0.5 keep probability in order to accelerate training progress and prevent overfitting. And also this layer is mainly trained so that my model can adapt my task very well. After the two stacked fully connected layers, there is two parallel small fully connection layers (fc_class and fc_bbox). The fc_class predict the probabilities of every class in the category and the fc_bbox will give the bounding box (x, y, w, h) of every possible class.

| Module name | Layer name | Layer hyperparameter |
|-------------|------------|----------------------|
| Head        | Conv1      | [3x3 64] x 2         |
|             | Pool1      | 2x2 max pool, padding='SAME' |
|             | Conv2      | [3x3 128] x 2        |
|             | Pool2      | 2x2 max pool, padding='SAME' |
|             | Conv3      | [3x3 256] x 3        |
|             | Pool3      | 2x2 max pool, padding='SAME' |
|             | Conv4      | [3x3 512] x 3        |
|             | Pool4      | 2x2 max pool, padding='SAME' |
|             | Conv5      | [3x3 512] x 3        |
| RPN         | rpn_conv   | 3x3 512              |
|             | rpn_cls_score | 1x1 4*9             |
|             | rpn_bbox_pred | 1x1 2*9             |
| R-CNN       | fc6        | 4096                 |
|             | dropout6   | keep_prob=0.5        |
|             | fc7        | 4096                 |
|             | dropout7   | keep_prob=0.5        |
|             | fc_class   | 2                    |
|             | fc_bbox    | 2*4                  |

3.2. Training
In the development environment, I use Python as a programming language and all models are trained on 4 NVIDIA GTX1080 GPUs. In terms of network models, I will try faster R-CNN, yolo v3 [5], SSD[6] and other target detection models and also Resnet, DenseNet [7] and other image classification models.

Strictly, this paper use approximate joint training train the model to update RPN and the Fast R-CNN independently. Because the ROI pooling layer is not derivable, the gradient of RPN loss will be
propagated through RPN layer and the head layer that is the feature extracting network (VGG-net) and update their weights. And the Fast R-CNN loss will be back propagation through the Fast R-CNN layer and ROI pooling layer.

I fixed the weights of conv1 and conv2 layers, because the first two convolution layers of network usually learn some edge, color and corner information. The goal of supervised learning is to extract as much of the feature information as possible according to the target task. And the weights of conv1 and conv2 layers can extract these features well for the different task. In the config.py, I set the hyper-parameter which are used in training. Where, I set the max iteration number equal 40000, the batch-size is 64 due to the memory restriction, the momentum of the operator is 0.9, the attenuation of learning rate is 0.1 and the other hyper-parameter in the region proposal network and NMS are set by default.

4. Results

In terms of data set, it is offline retail real scene data, including video data and picture data, pictures in PNG or JPG format, pedestrian resolution is not less than 50x50, video resolution is not less than 720P. The data set contains 5000 training samples. The training label includes all pedestrian locations, attributes (guide or customer, male or female), posture (standing or sitting), behavior (playing with or without mobile phones) in the current retail scenario. A demo of the result on test set is shown as Figure 3.

![Figure 2: the demo of pedestrian attributes recognition](image)

Extra experiment on pose estimation in order to help my model to predict the attributes that are stand, sit and play with phone. I use a top-down multi-person pose estimation which combine the OpenPose [8] and the yolo v3 [5]. The demo of pose estimation on my BOT dataset is shown in Figure 4.
5. Conclusion
This paper prefers the former to model accuracy and model size. Therefore, the second-order object detection network is more suitable for my requirement than the first-order object detection network, such as the faster R-CNN. After the object detection model, I combine the OpenPose model to predict the key points of detected pedestrians.

References
[1] Pan S J , Yang Q, (2010) A Survey on Transfer Learning[J]. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345-1359.
[2] K. He, X. Zhang, S. Ren, and J. Sun, (2016) “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
[3] K. Simonyan and A. Zisserman, (2014) “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556.
[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. International Conference on Neural Information Processing Systems.
[5] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
[8] Redmon J , Farhadi A . (2018) YOLOv3: An Incremental Improvement[J].
[9] Rasmus, Antti, et al. (2015) "Semi-supervised learning with ladder networks." Advances in neural information processing systems.