South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Xueyu Zhou
Shiyao Sun
Haoxuan Li

Supervisor:
Mingkui Tan

Student ID:
201930384264
201930382376
201930381263

Grade:
2019 Software Engineering Class 1 and 2

December 5, 2021

Face Detection Based on Neural Network

***Abstract*—This experiment requires us to understand and master the basic theoretical knowledge of face detection method based on neural network, and understand the basic process of MTCNN face detection for practice.**

I. Introduction

THIS experiment uses the MTCNN method of face recognition to train Pnet, Rnet and Onet in turn. This experimental report will focus on the experimental principle, implementation method and final results.

II. Methods and Theory

In this section, we will focus on the relevant knowledge and formula derivation of MTCNN.

A. Summary

In order to give consideration to both performance and accuracy and avoid the huge performance consumption brought by traditional ideas such as sliding window and classifier, MTCNN first uses a small model to generate the candidate box of the target region with certain possibility, and then uses a more complex model to carry out fine classification and higher-precision region box regression, and makes this step recursive. This idea forms a three-layer network, namely Pnet Rnet and Onet to realize fast and efficient face detection. In the input layer, the image pyramid is used for the scale transformation of the initial image, and Pnet is used to generate a large number of candidate target area boxes. Then Rnet is used for the first selection and border regression of these target area boxes, excluding most negative examples, and then o-net, a more complex and more accurate network, is used to distinguish and return the remaining target area boxes.

B. CNN Architectures

In the framework of MTCNN, the author puts the original 55 filter changed to 33 filter to reduce calculation and increase depth for better performance, and use PReLU as the activation function.

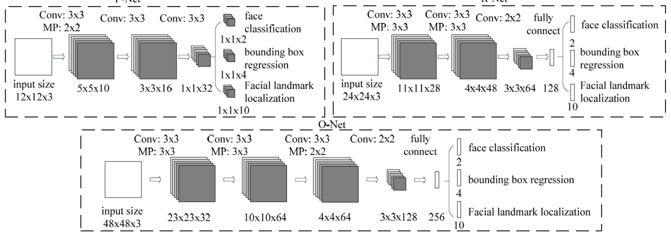

Fig. 1. Structure of MTCNN

1) Pnet: The full name is proposed network, and its basic structure is a full convolution network. The input is $12\times12\times3$. After convolution and maximum pooling, the final feature map is $1\times1\times32$, and then a full convolution is connected to obtain three outputs, which are $1\times1\times2$, $1\times1\times4$ and $1\times1\times10$ Here, the output of Pnet is also a feature map, and the dimensions are (N, W, H, C).

2) Rnet: The full name is refine network. Its basic structure is a convolutional neural network. Compared with the p-net of the first layer, a full connection layer is added, so the filtering of input data will be more strict. The input is $24 \times 24 \times 3$. After convolution and maximum pooling, finally connect the full connection layer to obtain 128 dimensional vectors, and then obtain three outputs. The dimensions are (N, C), and C is 16 channels, including 2 channels of face classification, 4 channels of bounding box offset and 10 channels of landmark offset.

3) Onet: The full name is output network. Its basic structure is a more complex convolutional neural network, which has one more convolution layer compared with R-Net. The input is $48\times48\times3$. After convolution and maximum pooling, finally connect the full connection layer to obtain 256 dimensional vectors, and then obtain three outputs. The dimensions are (N, C). Here, C is the same as Rnet.

C. Training

In the training of MTCNN, we mainly have three tasks: face detection, border regression and face feature point location.

1) Face Detection: The learning objective is formulated as a binary classification problem. For each sample x_i, we use the cross entropy loss function:

$$L_i^{det} = -(y_i^{det} log(p_i) + (1 - y_i^{det})(1 - log(p_i)))$$

p_i is the probability that the sample x_i is a face predicted by neural network. y_i^{det} stands for ground truth, and $y_i^{det} \in \{0, 1\}$

2) Border Regression: For each candidate window, we predict the offset between it and the nearest ground truth (such as the upper left corner coordinate of the regression box and the height and width). The learning objective is formulated as a regression problem, and the loss function is a square loss function:

$$L_i^{box} = ||\hat{y}_i^{box} - y_i^{box}||_2^2$$

$\hat{y}_i^{box}$ is the regression target obtained from the network, y_i^{box} is the ground truth coordinate, which is four-dimensional, including the upper left coordinate, height and width.

3) Face Feature Point Location: Similar to border regression, the loss function is:

$$L_i^{landmark} = ||\hat{y}_i^{landmark} - yi_{landmark}||_2^2$$

Similarly, $\hat{y}_i^{landmark}$ is the coordinate of feature points obtained from the network, and $y_i^{landmark}$ is the coordinate of ground truth. There are five coordinates, two eyes, two corners of mouth and one nose.

III. Experiments

A. Dataset

In order to train Pnet and Rnet, WiderFace is used for face classification and face bounding box regression; In order to train Onet, WiderFace is used for face classification and face bounding box regression, and training dataset is used for face feature point regression. The training dataset contains 5590 LFW pictures and 7976 other pictures; The test set includes 1521 BioId charts, 781 LFPW training charts and 249 LFPW test charts. WiderFace contains 32203 images and marks 393703 faces with high variability in scale, posture and occlusion. The dataset is organized based on 61 event classes. For each event category, 40%/10%/50% data were randomly selected as training, verification and test sets.

1) Multi-source Training: Because there are different tasks in each CNN, there will be different training data sets in the learning process. When training a specific task, the loss value of other tasks should be zero. Therefore, a loss function integrating all tasks is as follows:

$$min\sum_{i=1}^{N}\sum_{j\in\{det,box,landmark\}}\alpha_j\beta_i^j L_i^j$$

Where N represents the number of training samples and a_j represents the importance of each task. In Onet and Rnet,$\alpha_{det}=1,\alpha_{box}=0.5,\alpha_{landmark}=0.5$. However, in Onet, for higher accuracy of facial feature point coordinates, the parameter is set to $\alpha_{det}=1,\alpha_{box}=0.5,\alpha_{landmark}=1$is an indicator of the sample type. In this case, it is natural to use random gradient descent to train these CNN.

B. Implementation

Pnet, Rnet and Onet were trained successively. Finally, the trained model and the given model are tested and compared on the same test set.

For parameter initialization, see the following table:

TABLE I
PARAMETER INITIALIZATION

Train batch size	128
Train epoch	10
Rate of learning	0.01

Figure 2 shows the comparison of some test results selected by us. It can be seen that the effect of the training model is very close to that of the tracking model.

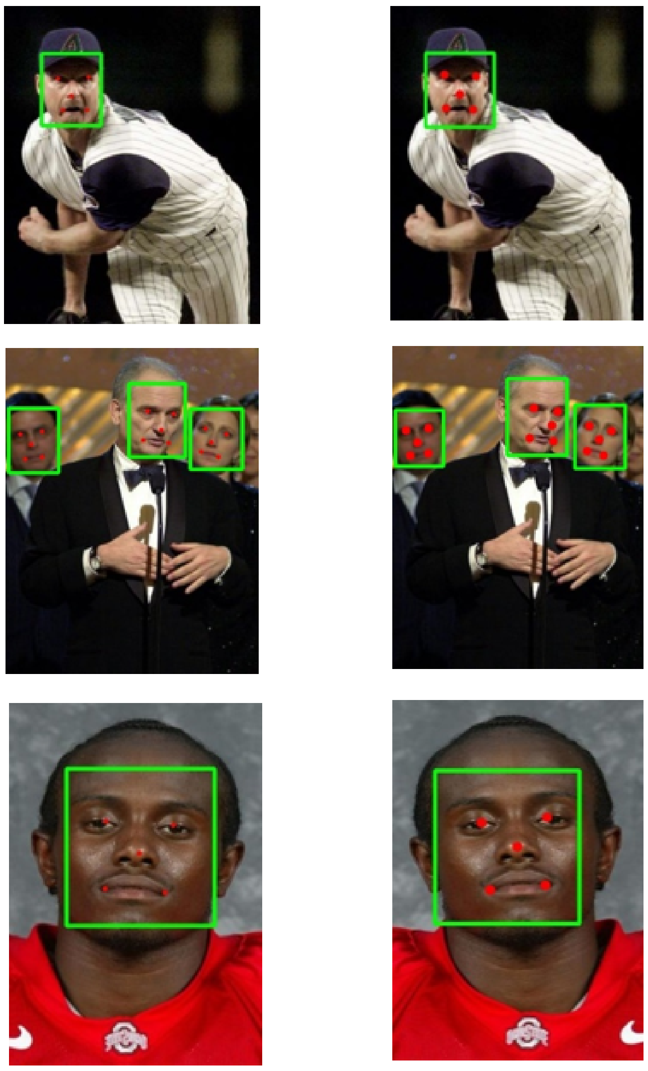

Fig. 2. Trained Model (Left) Given Model (Right)

IV. Conclusion

In this experiment, the classic MTCNN is used for face detection and face key point detection. The basic process is to process through Pnet, Rnet and Onet, and finally output the results of face detection and key point detection. MTCNN adopts the cascade idea to disassemble a large network into three small networks, so that the overall parameters are reduced and the operation is faster.

Through experiments, I understand the basic process of mtcnn face detection, and then deepen my understanding of neural network.