華南理工大學

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Xiaoxuan Peng

Supervisor:
Mingkui Tan

Student ID：
201730683321

Grade:
Undergraduate

November 10, 2019

Face Detection Based on Neural Network

Abstract—The experiment intends to use a given model to test the function about face detection based on neural network.

I. Introduction

There are many methods to implement face detection model, in this experiment, we use MTCNN(Multi-task Convolutional Nerual Network) to implement it. It contains three layers network structure, including P-Net(Proposal Network), R-Net(Refine Network) and O-Net(Output Network).

II. METHODS AND THEORY

A. P-Net(Proposal Network)

It is called Proposal Network, and its basic structure is a fully connected Network. For the image pyramid constructed in the previous step, an FCN is used for preliminary feature extraction and calibration of Bounding Box Regression window and NMS for filtering most of the Windows.

B. R-Net(Refine Network)

It is called Refine Network and it is basically constructed as a Convolutional Neural Network(CNN). Compared with the p-net in the first layer, a full connection layer is added, so the filtering of input data will be more rigorous. After p-net, a lot of prediction Windows will be left behind, and we will send all the prediction Windows into r-net. This network will filter out a large number of candidate boxes with relatively poor effects, and finally conduct Bounding Box Regression and NMS for the selected candidate boxes to further optimize the prediction results.

C. O-Net(Output Network)

Fully known as Output Network, the basic structure is a relatively complex Convolutional Neural Network(CNN) with one more convolutional layer than R-net. The difference between O-net and R-net is that this layer will recognize the facial area through more supervision, and it will return the facial feature points of people, and finally output the facial feature points of five people.

III. Experiment

A. Dataset

The dataset is WIDER FACE. The dataset contains 32,203 images and 393,703 labeled faces with a high degree of variability in scale, pose and occlusion as depicted in the sample images. WIDER FACE dataset is organized based on 61 event classes. For each event class, we randomly select 40%/10%/50% data as training, validation and testing sets.

B. Implementation

(1) Initialization

P-Net:

```
self.pre_layer = nn.Sequential(
    nn.Conv2d(3, 10, kernel_size=3, stride=1),  # conv1
    nn.PReLU(),  # PReLU1
    nn.MaxPool2d(kernel_size=2, stride=2),  # pool1
    nn.Conv2d(10, 16, kernel_size=3, stride=1),  # conv2
    nn.PReLU(),  # PReLU2
    nn.Conv2d(16, 32, kernel_size=3, stride=1),  # conv3
    nn.PReLU()  # PReLU3
)

# detection
self.conv4_1 = nn.Conv2d(32, 1, kernel_size=1, stride=1)
# bounding box regresion
self.conv4_2 = nn.Conv2d(32, 4, kernel_size=1, stride=1)
```

R-Net:

```
self.pre_layer = nn.Sequential(
    nn.Conv2d(3, 28, kernel_size=3, stride=1),  # conv1
    nn.PReLU(),  # prelu1
    nn.MaxPool2d(kernel_size=3, stride=2),  # pool1
    nn.Conv2d(28, 48, kernel_size=3, stride=1),  # conv2
    nn.PReLU(),  # prelu2
    nn.MaxPool2d(kernel_size=3, stride=2),  # pool2
    nn.Conv2d(48, 64, kernel_size=2, stride=1),  # conv3
    nn.PReLU()  # prelu3

)
self.fc = nn.Linear(64*2*2, 128)
self.prelu4 = nn.PReLU()  # prelu4
# detection
self.conv5_1 = nn.Linear(128, 1)
# bounding box regression
self.conv5_2 = nn.Linear(128, 4)
```

O-Net:

```
self.pre_layer = nn.Sequential(
    nn.Conv2d(3, 32, kernel_size=3, stride=1),  # conv1
    nn.PReLU(),  # prelu1
    nn.MaxPool2d(kernel_size=3, stride=2),  # pool1
    nn.Conv2d(32, 64, kernel_size=3, stride=1),  # conv2
    nn.PReLU(),  # prelu2
    nn.MaxPool2d(kernel_size=3, stride=2),  # pool2
    nn.Conv2d(64, 64, kernel_size=3, stride=1),  # conv3
    nn.PReLU(), # prelu3
    nn.MaxPool2d(kernel_size=2,stride=2), # pool3
    nn.Conv2d(64,128,kernel_size=2,stride=1), # conv4
    nn.PReLU() # prelu4
    )
    self.fc = nn.Linear(128*2*2, 256)
    self.prelu5 = nn.PReLU()  # prelu5
    # detection
    self.conv6_1 = nn.Linear(256, 1)
    # bounding box regression
    self.conv6_2 = nn.Linear(256, 4)
    # lanbmark localization
    self.conv6_3 = nn.Linear(256, 10)
```

(2) Parameters

Table 1 shown below listed some important parameters used in this experiment.

Table 1-Some Parameters in MTCNN_pytorch

Parameters	Values
Train_BATCH_SIZE	128
Train_LR	0.01
END_EPOCH	10

(3) Results

Take an example here. Figure 1 shows the initial image that will be used in face detection model and Figure 2 shows the final image that has been used in face detection model.

Figure 1-Initial image

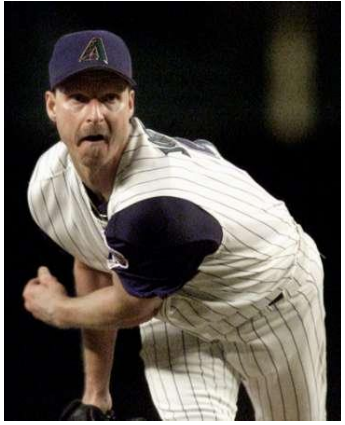
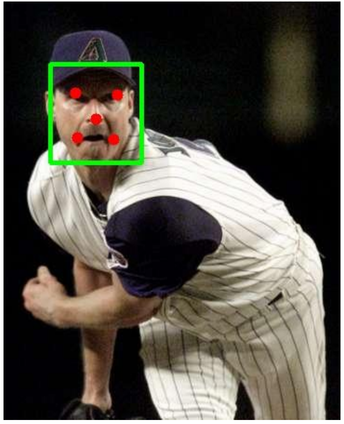


Figure 2-Result image



IV. CONCLUSION

In this experiment, I have learned that MTCNN is a powerful network structure used in face detection field. It puts face area detection and face key point detection together, and its theme framework is similar to cascade. The whole network can be divided into three layers: P-Net, R-Net and O-Net.