CS385 machine learning project 1

Face classification and detection

He Yunfan 516370910010

2019.6.21

1. Introduction
2. Dataset

The dataset we use is FDDB: face detection data set and benchmark. The first 8 folders are chosen as train set, and the last 2 folders are chosen as test set. There are 3618 pictures as train set and there are 1035 pictures as test set. We can generate eight negative pictures through one picture.

1. HOG

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image.

1. SVM

In machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

1. Methodology
2. Face classification

I extend the scale of the bounding box by 1/3.

Width = (axis\*2)\*4/3

Height = (axis\*2)\*4/3

Notes: There are two folders train and test. Bound box picture is in the “rawclip” folder, and the list of bound box picture is in “bblist.txt”. Positive picture is in the “positive” folder, and the list of bound box picture is in “poslist.txt”. Negative picture is in the “negative” folder, and the list of bound box picture is in “neglist.txt”.

  Bound box picture and positive picture

 8 negative pictures

1. Generate HOG feature for SVM

I use hog module in skimage.feature. I put all hog features in “hog\_neg.npy”, “hog\_pos.npy”.

 

Original HOG

1. SVM

First, I test a linear SVM for classification.

Linear

Accuracy 0.9311862587224906

precision recall f1-score support

-1.0 0.93 0.99 0.96 8280

1.0 0.89 0.43 0.58 1035

avg / total 0.93 0.93 0.92 9315

Second, I test a RBF SVM for classification.

Accuracy 0.8888888888888888

precision recall f1-score support

-1.0 0.89 1.00 0.94 8280

1.0 0.00 0.00 0.00 1035

avg / total 0.79 0.89 0.84 9315

Third, I test a ploy SVM for classification.

poly

Accuracy 0.8888888888888888

precision recall f1-score support

-1.0 0.89 1.00 0.94 8280

1.0 0.00 0.00 0.00 1035

avg / total 0.79 0.89 0.84 9315

1. CNN

The model summary is as followed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d\_1 (Conv2D) (None, 70, 70, 32) 896

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 68, 68, 32) 9248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 34, 34, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 34, 34, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 32, 32, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 30, 30, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 14400) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 256) 3686656

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 2) 514

=================================================================

Total params: 3,752,738

Trainable params: 3,752,738

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

The model is modified based on https://blog.csdn.net/qq\_35082030/article/details/79144374

32562/32562 [==============================] - 526s 16ms/step - loss: 1.8018 - acc: 0.8881

9315/9315 [==============================] - 41s 4ms/step

Score: 1.7908994983231836

Accuracy: 0.8888888888888888

1. Conclusion

The accuracy of SVM is 93%, 89%, 89%. And the CNN classification accuracy is 89%. All of the accuracy is acceptable.