Machine Learning Project1

Zhou Yiyuan 516021910270

May 2019

1 Dataset

FDDB(Face Detection Data Set and Benchmark) is used as the dataset. According to the requirement, all positive samples and negative samples are generated from FDDB, and both of them are divided into training sets and test sets. The distribution of the dataset is shown as the Table 1, that is positive:negative=1:4 and testing:training=1:4. The visualization of bounding boxes of positive and

	positive	negative	total
training	4136	16569	20705
testing	1034	4142	5176
total	5170	20711	25881

Table 1: The distribution of the dataset.

negative samples on images is showned as Figure 1. The blue ovals are original annotations, and the blue bounding boxes are positive samples, and the green bounding boxes are negative samples.

2 HOG

I use skimage package in python to extract HOG features, and the visualization is shown as Figure 2.

3 Logistic Model

I train logistic models for classification, optimizing the model via SGD and Langevin dynamics. The results are shown as Figure 3. The accuracy of SGD is slightly higher than Langevin dynamics but Langevin dynamics has more randomness thus is more likely to jump from local solutions.

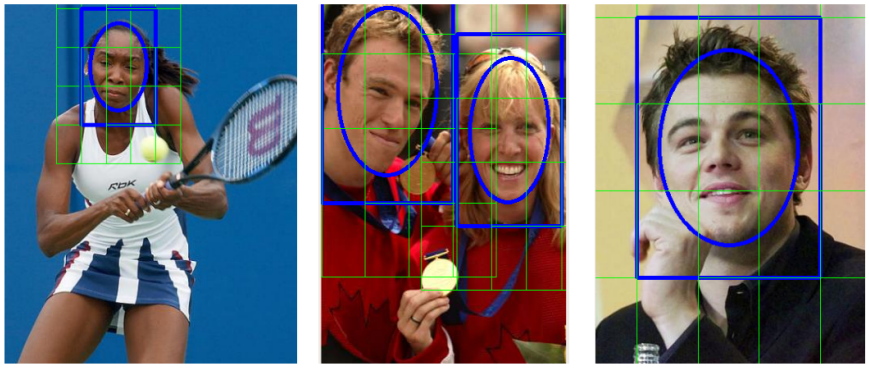

Figure 1: The visualization of bounding boxes.

4 Fisher Model

The last result of Fisher Model is shown as Table 2

accuracy in the training set	98.00%
accuracy in the testing set	97.80%
intra-class variance	0.0012668
inter-class variance	1.60481e-06

Table 2: The result of the Fisher Model

5 SVM

I use the scikit-learn package in python to train SVM model with linear kernel, RBF kernel and poly kernel. A text file containing file paths of all samples of support vectors will be created after training. The result of SVM models is shown as Table 3. Intuitively, the simpler kernel obtain higher accuracy and less support vectors in this task, which may be caused by overfitting.

kernel	accuracy	n_support_vectors(positive)	n_support_vectors(negative)
linear	98.13%	558	670
RBF	96.77%	1657	1644
sigmoid	96.54%	2082	2093

Table 3: The results of SVM models

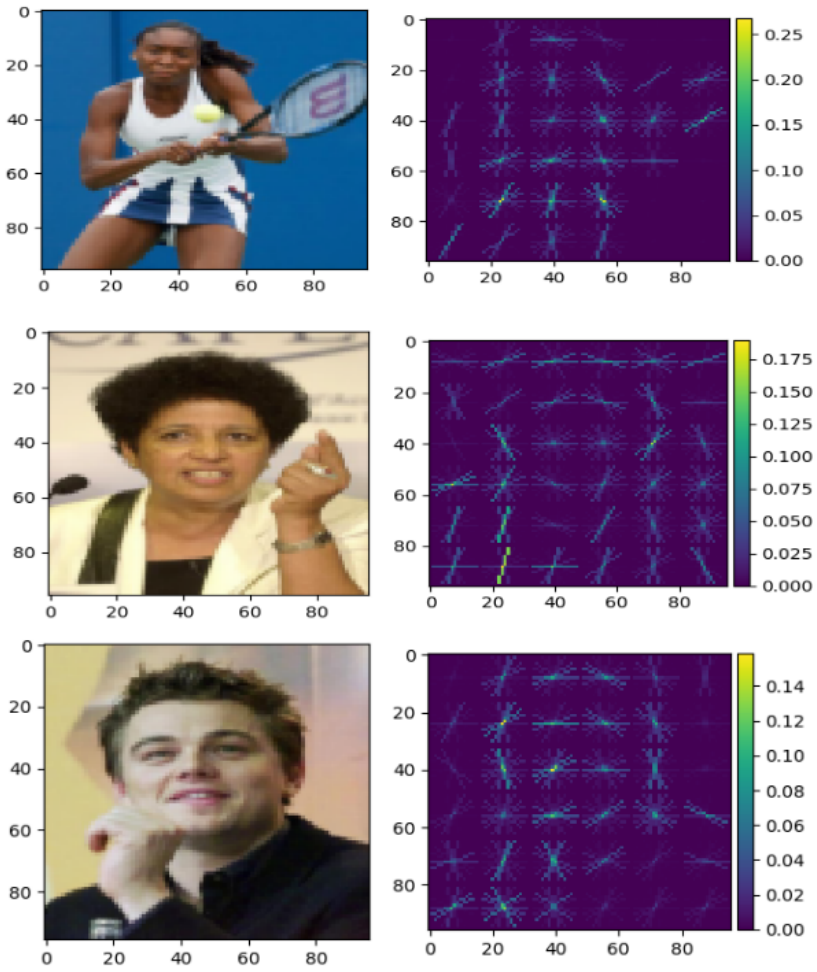

Figure 2: The visualization of HOG features.

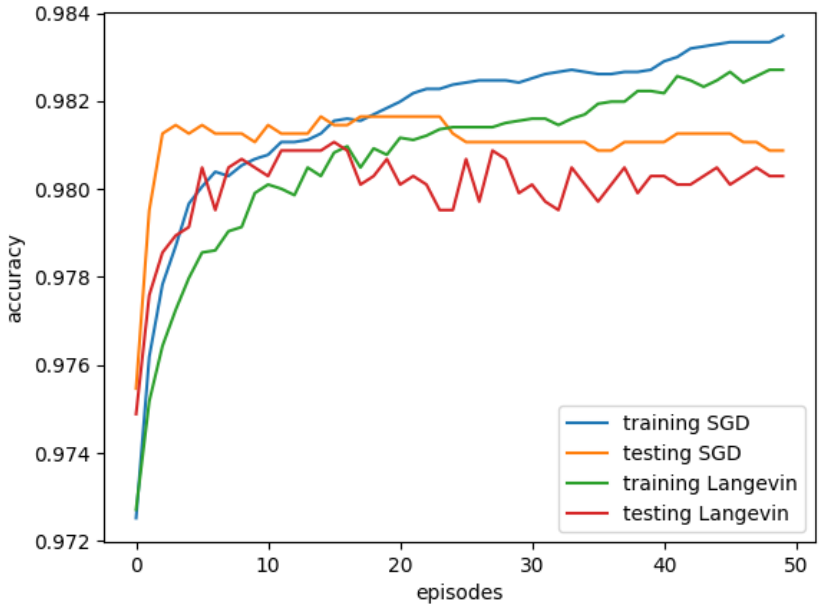

Figure 3: Logistic Model optimized by SGD and Langevin dynamics.

6 CNN

I use pytorch to build the CNN model. The structure of my model is shown as Figure 4. The loss and accuracy along with epoches is shown as Figure 7. After about 30 epochs, the training loss decreases while the testing loss increases, which means the overfitting occurs. The last accuracy in testing set can reach about 98.0%.

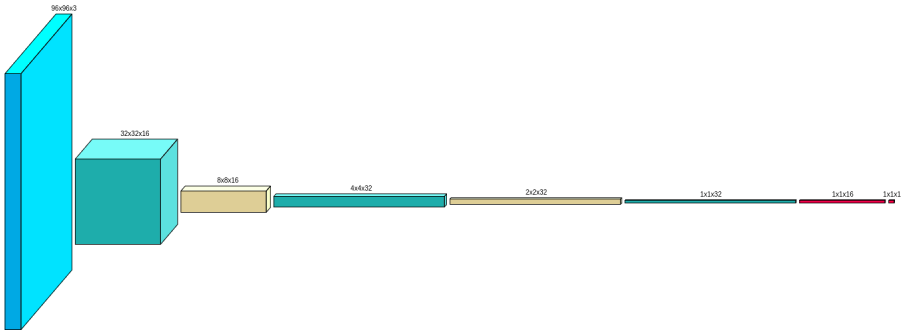

Figure 4: The structure of my model

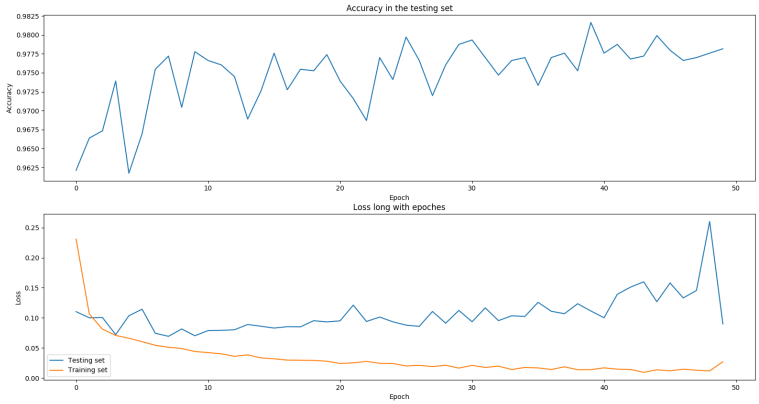

Figure 5: The results of CNN

7 Detect

7.1 Sliding Windows

Sliding windows are generated according to the following rules:

- 30 < window.width < img.width
- 30 < widow.height < img.heigh
- widow.ratio = min(widow.width/widow.height, window.heigh/window.width) and widow.ratio > 0.6
- window.width and widow.height increase by 1.33 times, which means 30, 40, 52, ...
- widow.ratio increases by 0.1 each time, which means 0.6, 0.7, 0.8, 1.0
- the stride of sliding windows is 1/3 of window.width or window.height

7.2 Non maximum suppression

NMS is used to remove overlap windows and I set the threshold IoU as 0.4.

7.3 Results

Figure 6: Result of logistic model

Figure 7: Result of cnn model

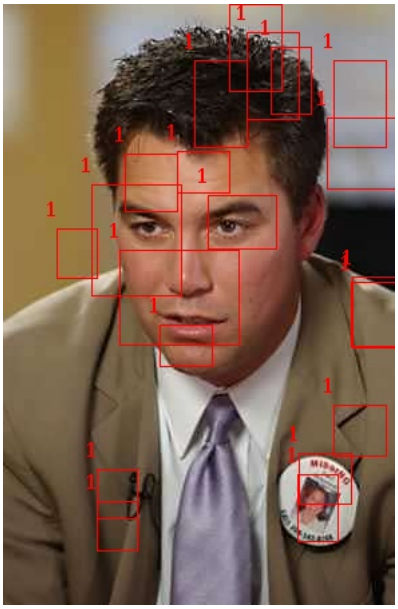

Figure 8: Result of linear SVM model

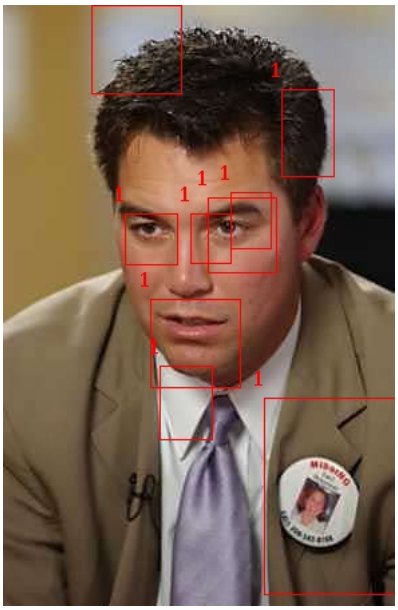

Figure 9: Result of RBF SVM model

7.4 Analysis

The result is awesome because the negative samples in training set is too less.