Machine Learning Project 1

Xiangyu Lin

May 2019

1 Introduction

This project aims to help us get to know about some machine learning methods by using a face detection dataset. The objective is to have a deeper insight of current models, including the intuition about the feature and the model, and the discrimination power of different models.

Here are some of the key environments for the project:

- Dataset: FDDB(Face Detection Dataset and Benchmark)
- Programming Environments: python3.6.7 and corresponding version of numpy, opencv and so on.

2 Data Processing

The target of data processing is to get the positive samples and negative samples from the dataset, and extract their hog features.

Corresponding files:

- get_positive.py
- get_negative.py
- get_hog.py
- Dataset.py

To get positive samples, we use the files in folder "FDDB-folds" to find the face from the image. And we expand the bounding box by 1/3 as requested. For face near the bound, we use padding method to pad the outer space.

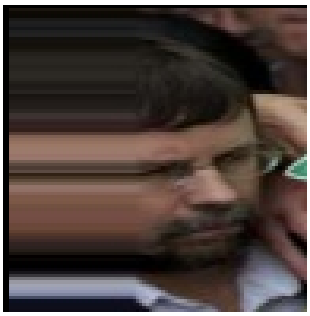

Figure 1: Positive sample using padding

To get negative samples, we use the method talked about in the 9th page of ppt, generating 8 negative samples for each face. By running the file visualize.py, we can visualize the positive and negative samples on the image.

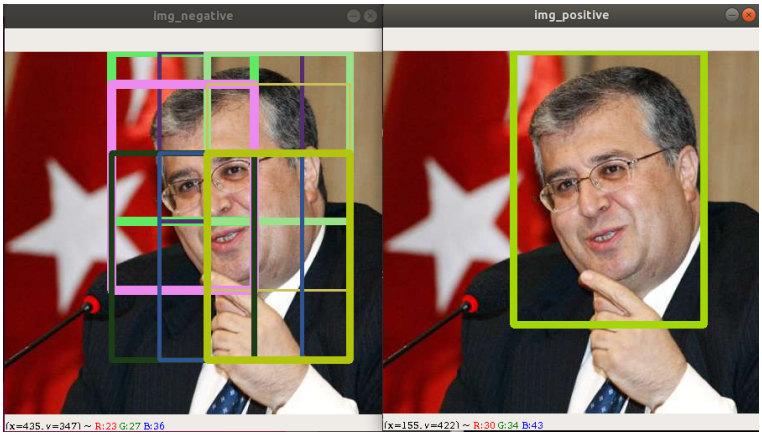

Figure 2: Negative and positive samples

After getting positive and negative samples, we resize the samples to (96,96) and save them in folders.Then we use get_hog.py to extract their hog features. For convenience, we use skimage to extract hog features, finally we get a 900-d feature vector for each sample.

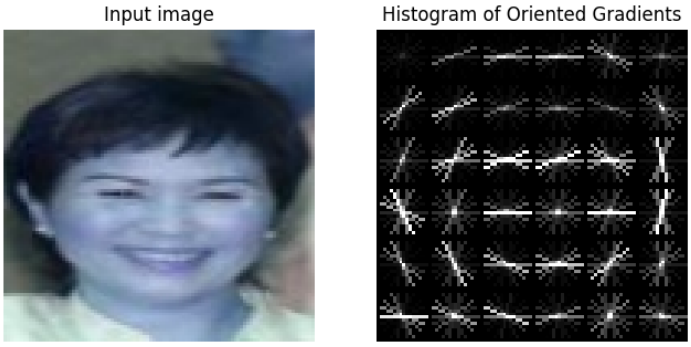

Figure 3: Visualization of hog feature

Dataset.py is used to arrange the hog feature vectors and generate labels.

3 Logistic model

File: logistic.py

First I build a logistic model without off-the-shelf codes. With labels of 1 and -1, the loss function for logistic model is as follows:

$$\text{Logistic loss} = \log\left(1+\exp(-y_i X_i^\top \beta)\right)$$

I first use SGD, each step is shown as follows:

$$\theta_{t+1} = \theta_t - \eta_t L_i'(\theta_t),$$

By calculation the partial derivative of logistic loss over to β, we can proceed the gradient descent process.

For langevin dynamics:

$$X_{t+\Delta t} = X_t - \frac{1}{2}U'(X_t)\Delta t + \sqrt{\Delta t}\varepsilon_t,$$

I consider Δ as learn rate, so langevin is considered as SGD with a gaussian noise ϵ.

The accuracy of SGD and langevin:

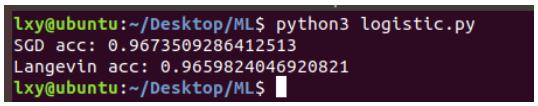

Figure 4: Accuracy of logistic model

4 Fisher model

File: fisher.py

Fisher linear discriminant is also called linear discriminant analysis(LDA).

Fisher model aims to maximizes the inter-class variance and minimizes the intra-class variance when doing dimension reduction. I use this method to get the answer directly, which can be viewed as reducing the feature over to the label space.

The formula we use is shown as follows:

$$\beta \propto S_W^{-1}(\mu^+ - \mu^-)$$

Where

$$S_W = n_{\text{pos}}\Sigma^+ + n_{\text{neg}}\Sigma^-$$

The result of fisher model is shown as follows:

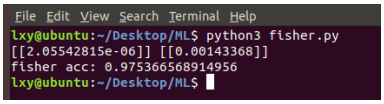

Figure 5: Result of fisher model

The first term is inter-class variance and the second term is intra-class variance.

5 SVM

SVM is a classifier which maximizes the margin between two classes. I use sklearn to build the SVM model.

I use three different kernels: linear, RBF and sigmoid. Their accuracy is shown as follows:

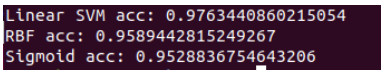

Figure 6: Result of SVM model

Linear kernel works better than RBF and sigmoid kernerl, and is much faster. For linear kernel, here are some of the support vectors:

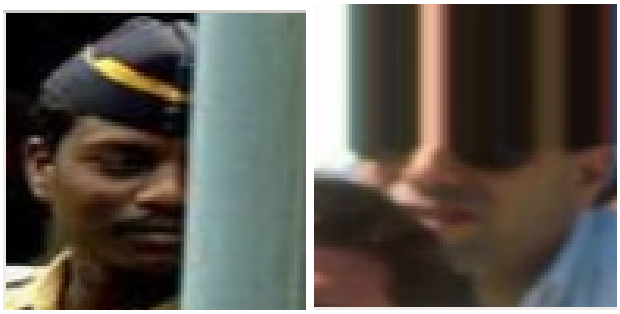

Figure 7: Positive support vectors

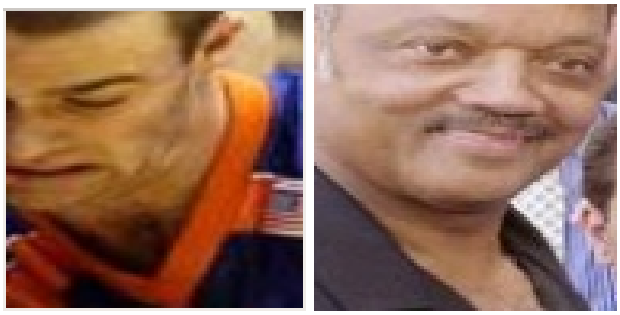

Figure 8: Negative support vectors

6 CNN

I use keras as a tool to build a CNN model. The model is shown as follows:

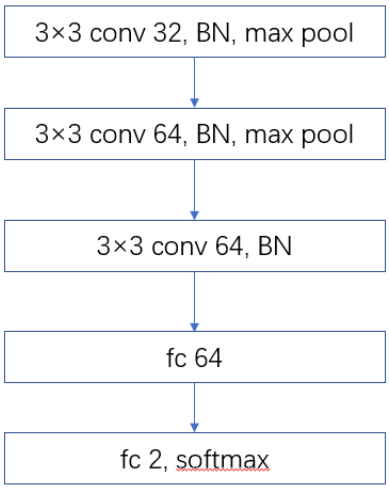

Figure 9: CNN model

And here is the result of CNN model:

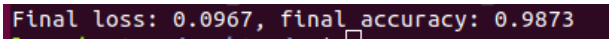

Figure 10: CNN model

7 Evaluation

Evaluate by accuracy, the performance is: CNN > linear SVM > fisher > SGD > langevin > RBF SVM > sigmoid SVM.

However, there's parameter tuning work for CNN model, while none for others. Linear SVM and fisher model are good alternatives.

8 Face detection

All the models perform bad in face detection as they may recognize other irrelevant things as faces, and there's little difference in their performance as they are all with similar accuracy. The reason may be that the negative samples are too naive. But they are still able to find the face if bounding box placed properly.

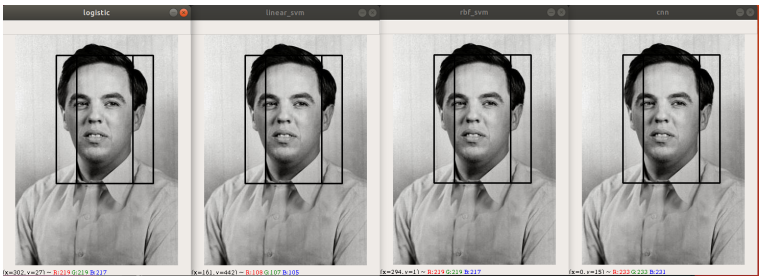

Figure 11: Face Detection

Besides, enumerating all the locations and sizes takes too much time and space, so I just enumerate some of the cases.

9 Feature Visualization

Here's the visualization of hog feature:

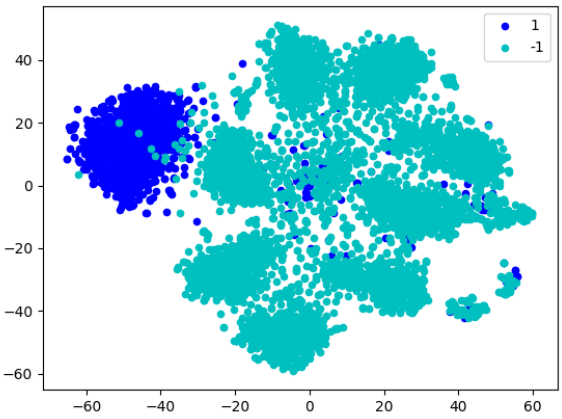

Figure 12: Hog feature distribution visualization

We can see that positive and negative features are distinct and generally linearly seperable in this figure. This is the reason why simple models like logistic and SVM work well in this project.

10 Summary

In this project, I tried several machine learning models on the dataset, and they all achieve good results. However, due to the naiveness of negative sample

formation, the performance of face detection is not as good as expected. But in general, this is a successful project for me and I learnt a lot from it.