# CS385 PROJECT I REPORT

**June 18, 2019**

Zixuan Chen
516030910545
Shanghai Jiao Tong University
School of Electronic Information and Electrical Engineering

# Contents

0.1 Preprocessing . . . . . 2
0.1.1 Image Sample Extration . . . . . 2
0.1.2 HOG Feature Extration . . . . . 2
0.2 Classification . . . . . 3
0.2.1 Logistic Model . . . . . 3
0.2.1.1 Model via SGD . . . . . 4
0.2.1.2 Model via Langevin . . . . . 4
0.2.2 Fisher Model . . . . . 4
0.2.3 SVM . . . . . 5
0.2.4 Neural Networks . . . . . 6
0.3 Detection . . . . . 6
0.3.1 detection strategy . . . . . 6
0.3.2 result of detection . . . . . 7
0.4 Visualization . . . . . 7
0.4.1 PCA . . . . . 7
0.4.2 t-SNE . . . . . 7

## 0.1 Preprocessing

### 0.1.1 Image Sample Extration

For the positive example, we generate one $96 \times 96$ cropped image for each face.For the positive example, we generate 9 $96 \times 96$ cropped image for each face.We extracted 12665 sample images for training and 3059 sample images for validation

**Figure 1:** positive samples

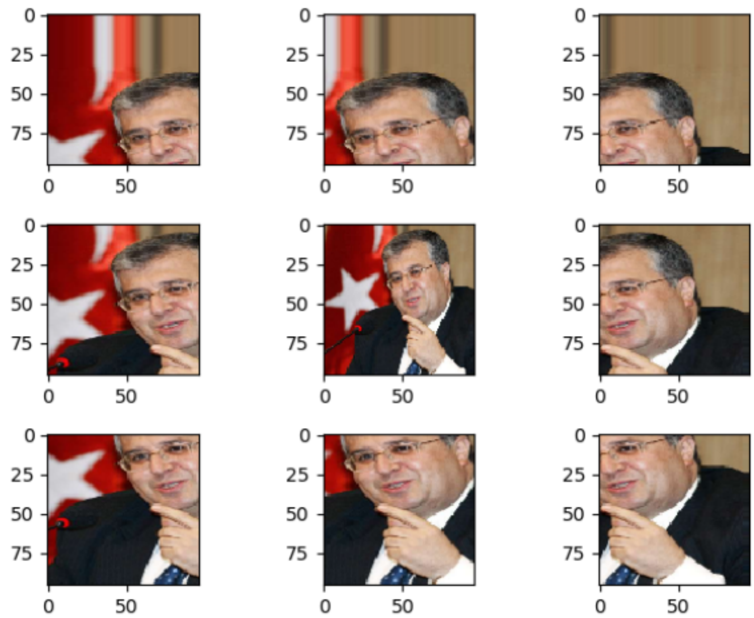


**Figure 2:** negative samples

### 0.1.2 HOG Feature Extration

We extract one HOG feature for each image sample, which is of size (1, 900). We can also show the HOG features and compare it with the original image. We extracted 22748 HOG Features for training and 5724 HOG examples for validation

**Figure 3:** original image

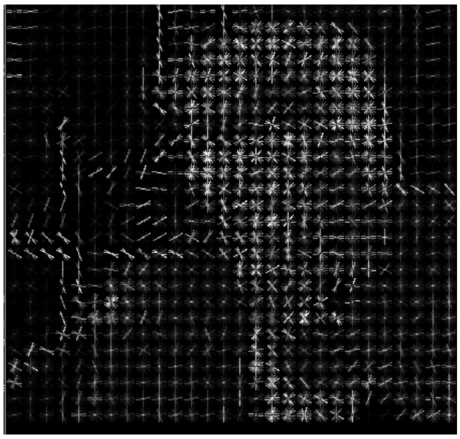

**Figure 4:** hog visualization

**Figure 5:** compare hog with the original image

## 0.2 Classification

We set the label of positive example to 1 and the negative example to -1. So it is a 2-class classification problem, the classification result of all models is shown in the following table

| name | accuracy(%) |
|---|---|
| Logistic(SGD) | 96.6 |
| Logistic(Langevin) | 95.8 |
| Fisher | 96.3 |
| SVM(Linear) | 97.2 |
| SVM(RBF) | 97.8 |
| SVM(Poly) | 98.1 |
| CNN | 98.0 |

**Table 1:** classification result

### 0.2.1 Logistic Model

For Logistic Model, we have

$$Logit_i = X_i W + b$$

$$Pr(y_i) = \frac{1}{1 + exp(-y_i \cdot Logit_i)}$$

and the loss function with regularization

$$L = \sum_{i=1}^{n} log(1 + exp(-y_i \cdot Logit_i)) + r||W||^2$$

where $r$ is the regularization rate. Since we use the back propagation to update the parameters, we need to calculate the derivative first.

$$\frac{\partial L}{\partial W^T} = -\sum_{i=1}^{n} \frac{y_i X_i exp(-y_i \cdot Logit_i)}{1 + exp(-y_i \cdot Logit_i)} + 2rW$$

$$\frac{\partial L}{\partial b^T} = -\sum_{i=1}^{n} \frac{y_i exp(-y_i \cdot Logit_i)}{1 + exp(-y_i \cdot Logit_i)}$$

#### 0.2.1.1 Model via SGD

For SGD, we choose a mini-batch to update at each time.

$$W = W - \alpha \frac{\partial L}{\partial W^T}$$

$$b = b - \frac{\partial L}{\partial b^T}$$

#### 0.2.1.2 Model via Langevin

The Langevin is the same as SGD, except that we add a small random walk on each parameter update.

$$W = W - \alpha \frac{\partial L}{\partial W^T} + \sqrt{\alpha}\epsilon_t$$

$$b = b - \frac{\partial L}{\partial b^T} + \sqrt{\alpha}\epsilon_t$$

### 0.2.2 Fisher Model

The Fisher Model, i.e. LDA model project high dimensional features to low dimensional space on a special hyper-plane. In this problem, we project the 900-dimensional hog features to 1-dim space. What we have to train is the hyper-plane we are trying to project on. According to the lecture note, the most key step in this algorithm is to calculate the inverse matrix of $S_w$. Once the $S_w^{-1}$ is calculated, we finish training. But During training steps, I found some times the inverse step failed because we get a singular $S_w$. And we can do PCA of the singular matrix and derivate a non-singular from it.

the inter-class variance $= 1.142 \times 10^{-6}$, the intra-class variance $= 1.229 \times 10^{-3}$

### 0.2.3 SVM

We use Liear Kernel, RBF Kernel and Polynormial Kernels to train the SVM. To show the ground property of the support vectors. We show the histogram of support vectors along some specific axis, eg. axis 1 of 900. The result is shown in the following graph

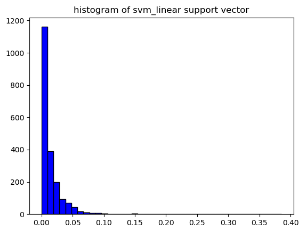


**Figure 6:** linear

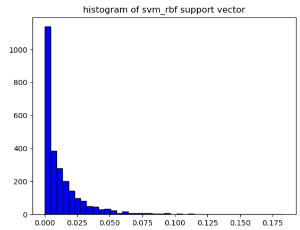


**Figure 7:** rbf

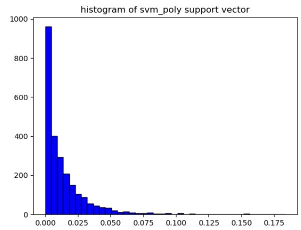


**Figure 8:** poly

**Figure 9:** support vector distribution along axis 1

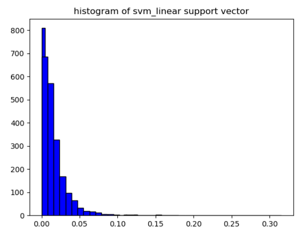


**Figure 10:** linear

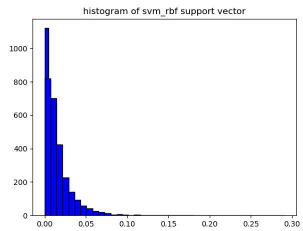


**Figure 11:** rbf

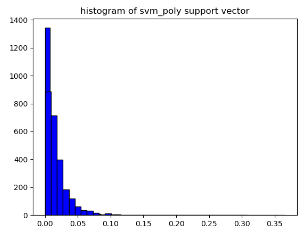


**Figure 12:** poly

**Figure 13:** support vector distribution along axis 450

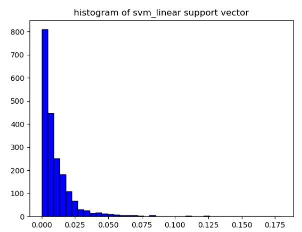


**Figure 14:** linear

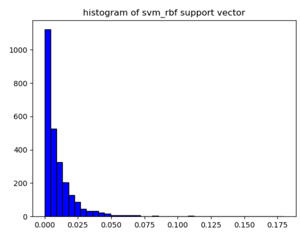


**Figure 15:** rbf

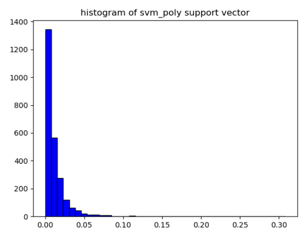


**Figure 16:** poly

**Figure 17:** support vector distribution along axis 89

we can see from the support vector distributions on these graph. What we find is that all kinds of SVMs have similar distributions along the same axis, and for a specific SVM, its feature distributions on all of its axis have some similar pattern, (e.g. the feature distribution of RBF SVM is more discrete than other SVMs.)

### 0.2.4 Neural Networks

We derive our network structure from AlexNet, Where we change its MaxPooling Layer to AdaptiveAvgPool2d Layer and the result is good.

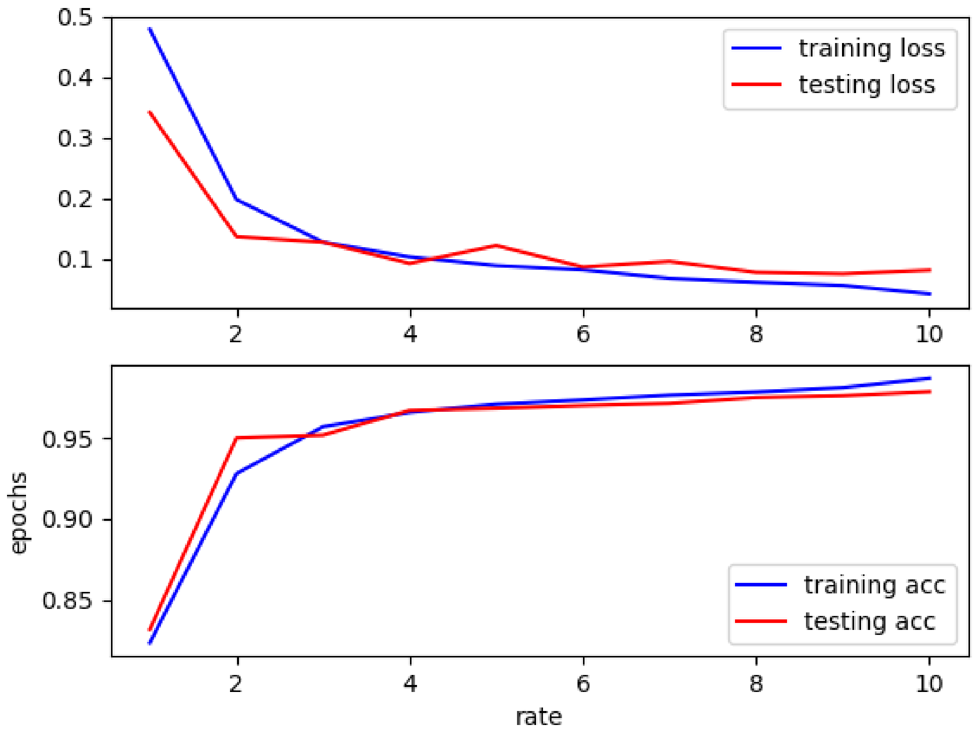


**Figure 18:** cnn

## 0.3 Detection

### 0.3.1 detection strategy

We locate the bounding box by a (leftup_x, leftup_y, height, width) tuple. We use the normal for-loop to enumerate all kinds of bounding box. So Theoretically we step sufficiently small intervals, we can catch the appropriate detection result. We set the loop step to be $step_w = \frac{1}{20} weight$, $step_h = \frac{1}{20} height$.

### 0.3.2 result of detection

From Feature 19, We see that all model can recognize the face in most images. However, there is inevitably many kinds of misjudgement. Some fails to have to many overlaps, others recognize areas which are total not have relation with face.

## 0.4 Visualization

We Visualize the HOG Feature using t-SNE and PCA.

### 0.4.1 PCA

The following graph shows a 3D-PCA. In this graph we label the positive image samples as 1 and the negative example as 0. We can see that most of the image samples labelled 1 are cluster together.

### 0.4.2 t-SNE

from the t-SNE, We can see that most of the image samples labelled 1 are cluster together. We can see more clearly from the 3D t-SNE

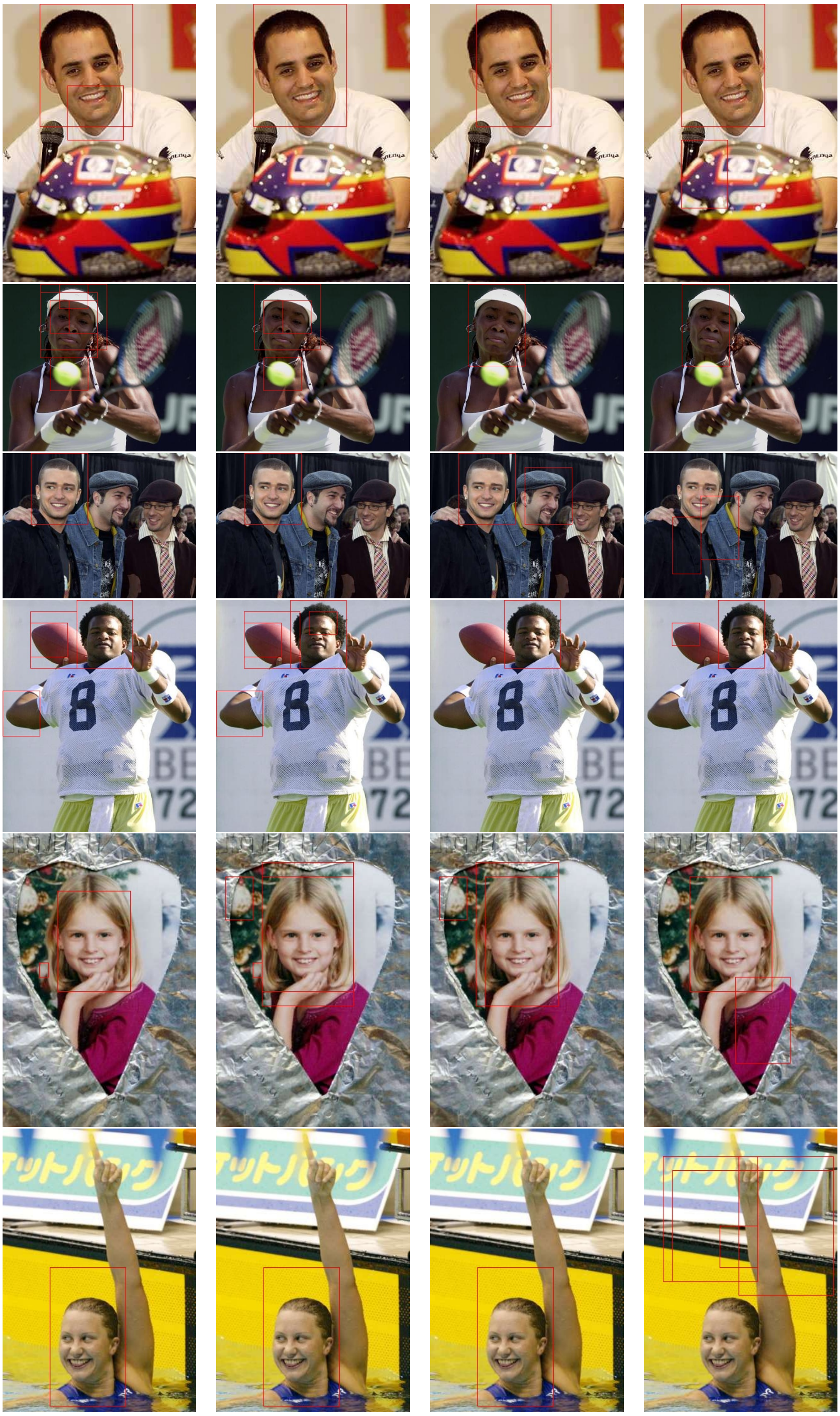

**Figure 19:** detection result, models from left to right: Logistic, Linear_SVM, RBF_SVM, CNN

**Figure 20:** global view

**Figure 21:** dark area

**Figure 22:** PCA

**Figure 23:** global view

**Figure 24:** dark area

**Figure 25:** t-SNE 2D

**Figure 26:** t-SNE 3D

## ACKNOWLEDGEMENT

Thanks for Prof. Zhang and all the TAs. The course lets me learn a lot deep insight and different perspectives to see machine learning. Without your help, I canâĂŹt ïňĄnish all these projects and applications. Thank you.