**Famous48 Face Recognize Report**

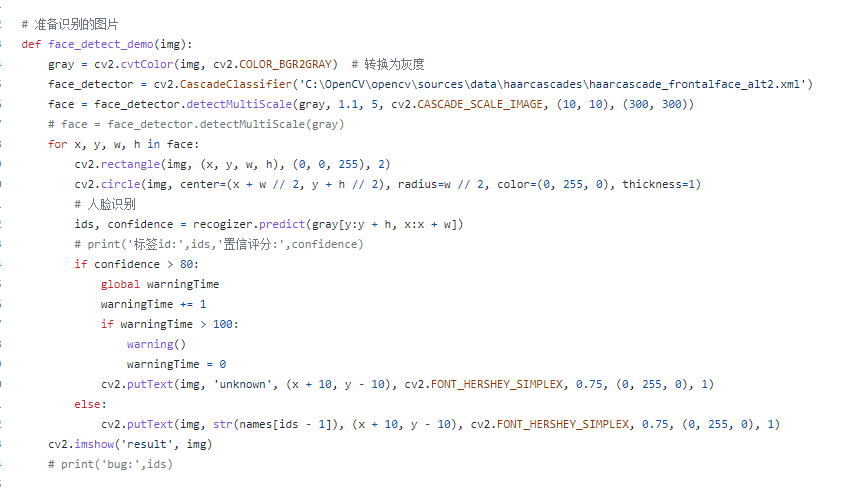
**--write feature & design classifier algorithm manually (no library)**

**Yifei Liu s188026**

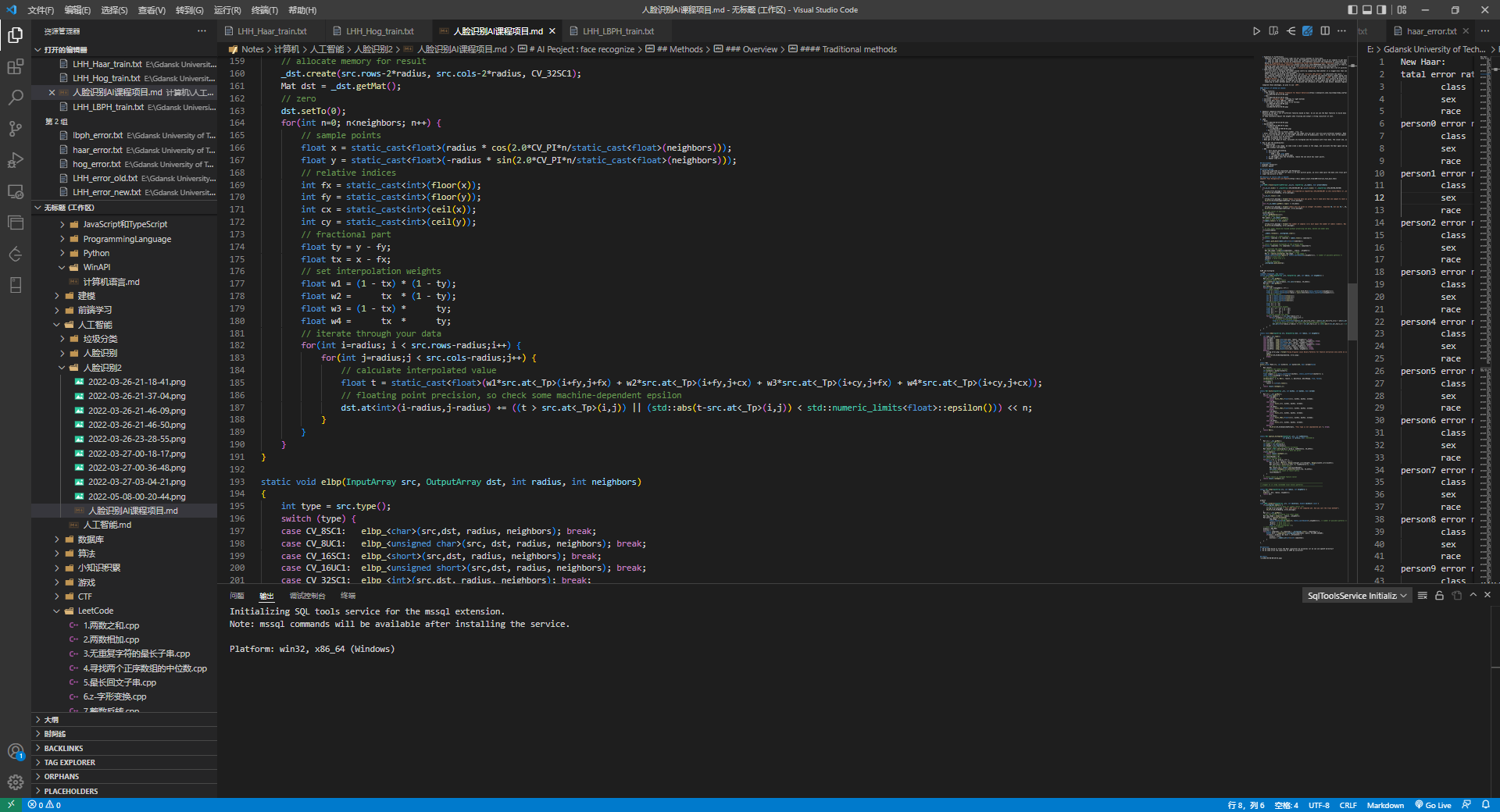
**Kacper Guzewicz s184339**

**Description:**

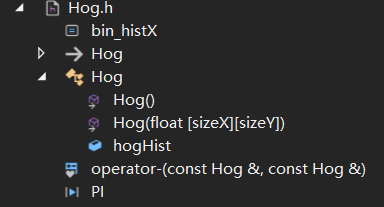
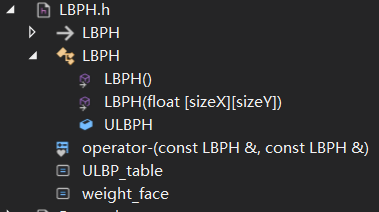
1. Firstly, try to use python + OpenCV to see how the face recognize work (<https://github.com/DuGuYifei/FaceRecognitionDemo>). Below shows how OpenCV read picture/video/camera to do face detection firstly then use model from library to do face recognition.

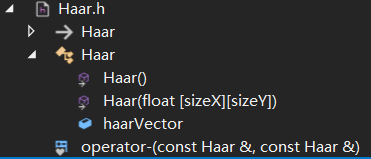
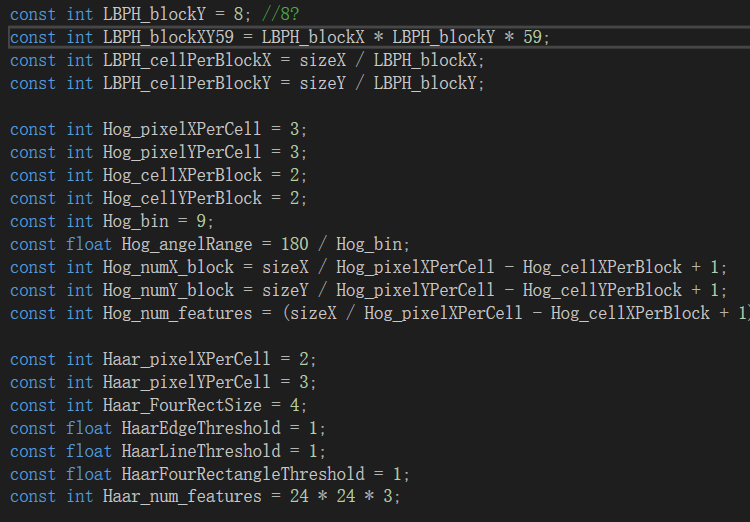
**** 

1. Analyse the source code of OpenCV.



1. Write three different classifiers manually in **C++**:

For ***LBP***, we choose **ULBP (LBPH.h)** which can decrease the dimension of histogram into 59. After training with different parameters we choose **8\*8 blocks** and using weight of different block in face to calculate **chi-square distance**.

For ***Hog* (Hog.h)**, after trying different parameter we choose **3\*3 (pixels)** **cell**, **2\*2 (cells) blocks** and **9 bins**. Using slide window in the face to get the histogram.

For ***Haar* (Haar.h)** we try two different ways: one is using ‘bool’ value as flag of is pixels in one Haar feature; one is storing the feature value directly. And as we wish, the direct value will better than previous one, because bool will decrease the difference between similar person.

1. Design whole classifier:
2. Our group wanted to use **Adaboost**, but when in practise, we find that, with such low quality picture (24\*24) and so many classes (48 classes), it’s hard to find a threshold to classify different people no matter we choose **multi-classes** adaboost or **two-classes** adaboost.
3. Based on different sub-classifier will have different error rate for different people, we use **error rate (e)** in weight of each sub-classifier as **1-e**. Then we choose **top 5 closed person from each sub-classifier**. Then they use weight to vote these people, choose the person who has the highest votes.
4. But for such pictures with low quality (24\*24), some people is hard to recognize in any sub-classifier. So we can make a threshold of the votes weight (v): **if v < threshold, let v = 1-v+adjustment**. So we need train for **threshold** and **adjustment value**. It will increase the votes weight of people who hard to recognize but not influence the votes result of simple person.

Name it as **LHH** (LBPH + Hog + Haar) algorithm.

**Workflow**

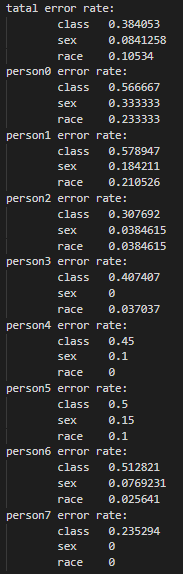
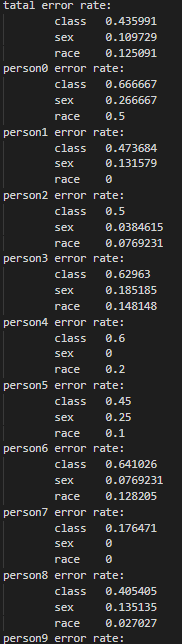
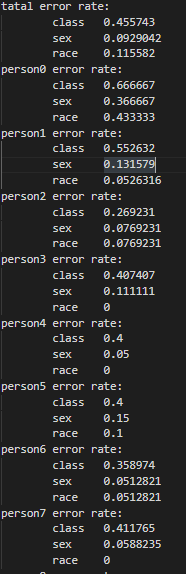
1. **Train 3 sub-classifiers: LBPH, Hog, Haar and divide data into 3 parts.**
   1. Get histogram of each picture and mark its labels.
   2. Each 5 pictures: 1-3 using train sub-classifiers, 4 train LHH, 5 test model.
2. **Train LHH (LBPH + Hog + Haar) classifier:**
   1. Get the error rate of each sub-classifier.
   2. 1-e is the weight of votes.
3. **Train weights of LHH:**

Let threshold 0.5~0.0, adjustment 0.0~1.0, train model get the lowest error rate.

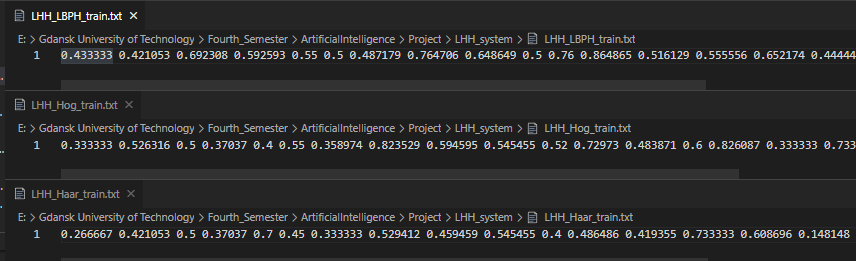
**Result Display**

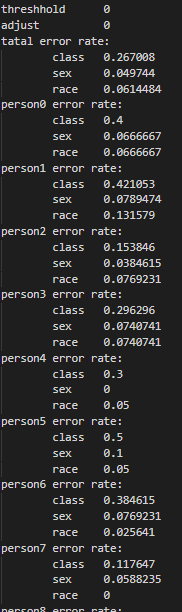
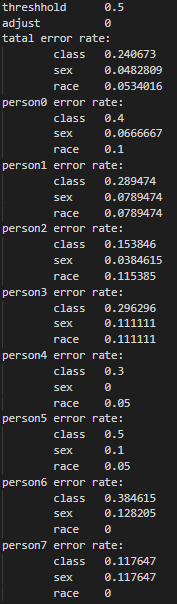
Here we also get the error rate of sex and race, because we want to know if it is wrong whether it is because these two persons are similar (in same sex or same race).

1. LBPH, Hog, Haar:

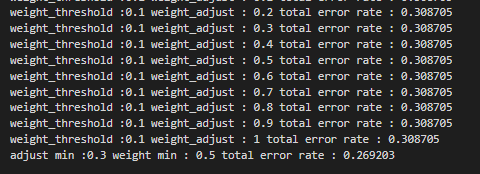
  

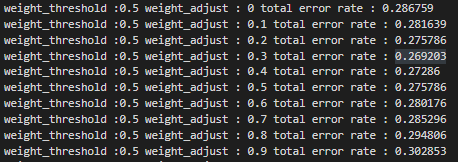
1. LHH train result and test result without train threshold + adjustment value ([0,0] and [0.5,0]):



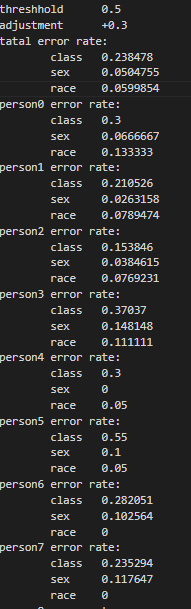
1. Train votes rate:





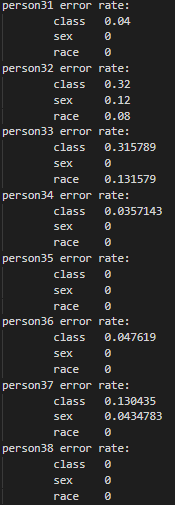
The train result arount minimum value is similar with parabola.

1. The final result:

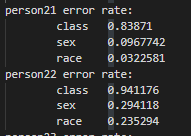


**Conclusion**

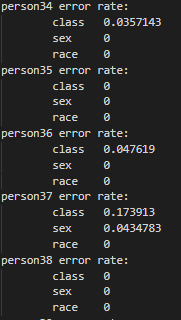
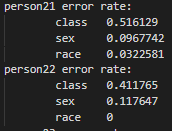
Final error rate is 0.23 which is relative really low with these pictures. Also if we kick some person class out it will get more low error rate. For example, some people we can **have the error rate even 0** like thisif we don’t train LHH and use it directly:



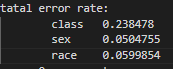
While some people **error rate is almost 100%** like this:



After train LHH, we can make it better:



And totally the result is:

****

which means that we have 0.23 error rate of person class, but peroson who are recognized wrong is because they are similar in sex or race (the error rate of sex and race is low).

The classifier is training for 48 classes, not for the sex and race which are smaller than 48 classes, otherwise the error rate of them should be lower.