Famous48 Face Recognize Report

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

Yifei Liu **s188026**
Kacper Guzewicz **s184339**

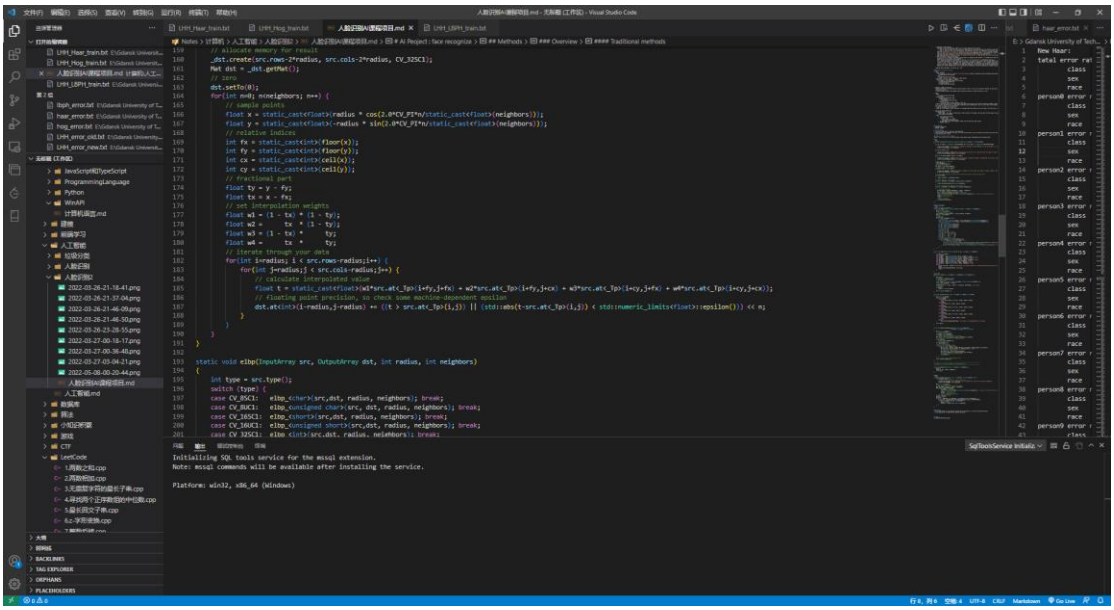
Description:

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.

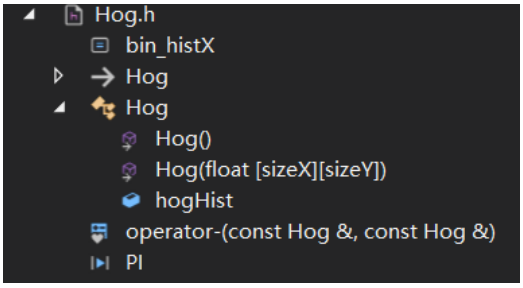
```python
# 准备识别的图片
def face_detect_demo(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # 转换为灰度
    face_detector = cv2.CascadeClassifier('C:\OpenCV\opencv\sources\data\haarcascades\haarcascade_frontalface_alt2.xml')
    face = face_detector.detectMultiScale(gray, 1.1, 5, cv2.CASCADE_SCALE_IMAGE, (10, 10), (300, 300))
    # face = face_detector.detectMultiScale(gray)
    for x, y, w, h in face:
        cv2.rectangle(img, (x, y, w, h), (0, 0, 255), 2)
        cv2.circle(img, center=(x + w // 2, y + h // 2), radius=w // 2, color=(0, 255, 0), thickness=1)
        # 人脸识别
        ids, confidence = recogizer.predict(gray[y:y + h, x:x + w])
        # print('标签id:',ids,'置信评分:',confidence)
        if confidence > 80:
            global warningTime
            warningTime += 1
            if warningTime > 100:
                warning()
                warningTime = 0
            cv2.putText(img, 'unknown', (x + 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 255, 0), 1)
        else:
            cv2.putText(img, str(names[ids - 1]), (x + 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 255, 0), 1)
    cv2.imshow('result', img)
    # print('bug:',ids)
```

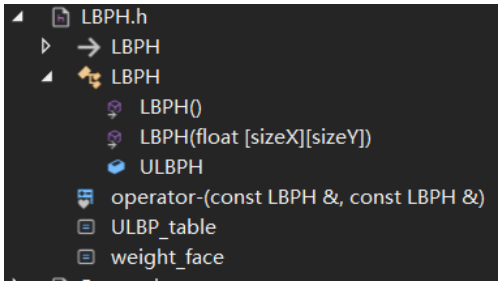
```python
# 读取摄像头
cap = cv.VideoCapture(0)
# cap = cv.VideoCapture('E:\视频\\test.flv')
"""
等待输入视频的名字 test被我删除了
"""
# 等待
while True:
    flag, frame = cap.read()
    if not flag:
        break
    face_detect_demo(frame)
    if cv.waitKey(1) & 0xFF == ord('q'):
        break
# 释放内存
cv.destroyAllWindows()
# 释放摄像头
cap.release()
```

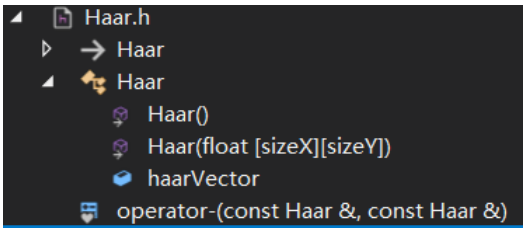
2. Analyse the source code of OpenCV.

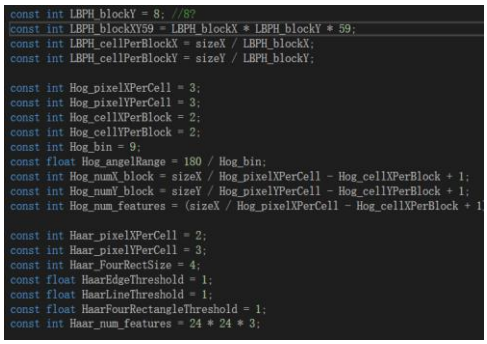


3. Write three different classifiers manually in **C++**:









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.

4. Design whole classifier:
   a) 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.
   b) 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.
   c) 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.**
   a) Get histogram of each picture and mark its labels.

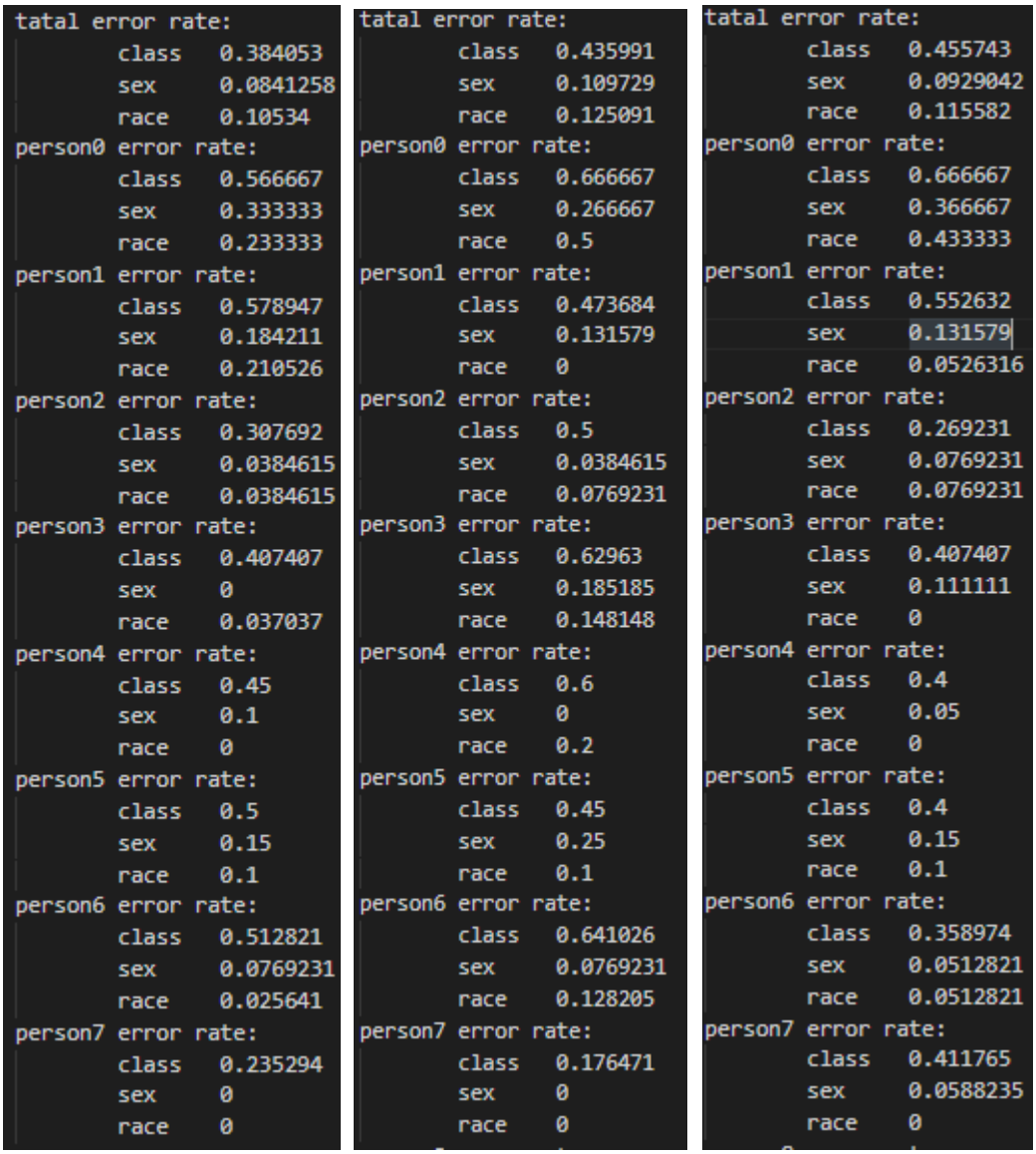
b) Each 5 pictures: 1-3 using train sub-classifiers, 4 train LHH, 5 test model.

2. **Train LHH (LBPH + Hog + Haar) classifier:**
   a) Get the error rate of each sub-classifier.
   b) 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:

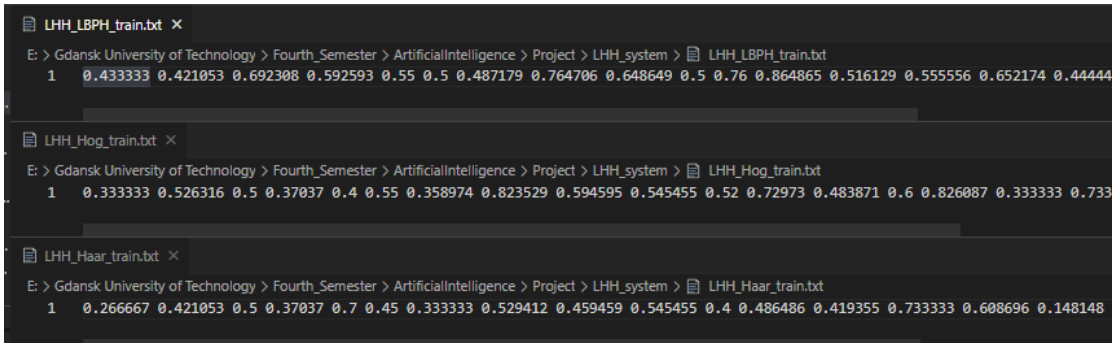


```
tatal error rate:
        class   0.384053
        sex     0.0841258
        race    0.10534
person0 error rate:
        class   0.566667
        sex     0.333333
        race    0.233333
person1 error rate:
        class   0.578947
        sex     0.184211
        race    0.210526
person2 error rate:
        class   0.307692
        sex     0.0384615
        race    0.0384615
person3 error rate:
        class   0.407407
        sex     0
        race    0.037037
person4 error rate:
        class   0.45
        sex     0.1
        race    0
person5 error rate:
        class   0.5
        sex     0.15
        race    0.1
person6 error rate:
        class   0.512821
        sex     0.0769231
        race    0.025641
person7 error rate:
        class   0.235294
        sex     0
        race    0
```

```
tatal error rate:
        class   0.435991
        sex     0.109729
        race    0.125091
person0 error rate:
        class   0.666667
        sex     0.266667
        race    0.5
person1 error rate:
        class   0.473684
        sex     0.131579
        race    0
person2 error rate:
        class   0.5
        sex     0.0384615
        race    0.0769231
person3 error rate:
        class   0.62963
        sex     0.185185
        race    0.148148
person4 error rate:
        class   0.6
        sex     0
        race    0.2
person5 error rate:
        class   0.45
        sex     0.25
        race    0.1
person6 error rate:
        class   0.641026
        sex     0.0769231
        race    0.128205
person7 error rate:
        class   0.176471
        sex     0
        race    0
```

```
tatal error rate:
        class   0.455743
        sex     0.0929042
        race    0.115582
person0 error rate:
        class   0.666667
        sex     0.366667
        race    0.433333
person1 error rate:
        class   0.552632
        sex     0.131579
        race    0.0526316
person2 error rate:
        class   0.269231
        sex     0.0769231
        race    0.0769231
person3 error rate:
        class   0.407407
        sex     0.111111
        race    0
person4 error rate:
        class   0.4
        sex     0.05
        race    0
person5 error rate:
        class   0.4
        sex     0.15
        race    0.1
person6 error rate:
        class   0.358974
        sex     0.0512821
        race    0.0512821
person7 error rate:
        class   0.411765
        sex     0.0588235
        race    0
```

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




```
threshhold      0
adjust          0
tatal error rate:
        class   0.267008
        sex     0.049744
        race    0.0614484
person0 error rate:
        class   0.4
        sex     0.0666667
        race    0.0666667
person1 error rate:
        class   0.421053
        sex     0.0789474
        race    0.131579
person2 error rate:
        class   0.153846
        sex     0.0384615
        race    0.0769231
person3 error rate:
        class   0.296296
        sex     0.0740741
        race    0.0740741
person4 error rate:
        class   0.3
        sex     0
        race    0.05
person5 error rate:
        class   0.5
        sex     0.1
        race    0.05
person6 error rate:
        class   0.384615
        sex     0.0769231
        race    0.025641
person7 error rate:
        class   0.117647
        sex     0.0588235
        race    0
```

```
threshhold      0.5
adjust          0
tatal error rate:
        class   0.240673
        sex     0.0482809
        race    0.0534016
person0 error rate:
        class   0.4
        sex     0.0666667
        race    0.1
person1 error rate:
        class   0.289474
        sex     0.0789474
        race    0.0789474
person2 error rate:
        class   0.153846
        sex     0.0384615
        race    0.115385
person3 error rate:
        class   0.296296
        sex     0.111111
        race    0.111111
person4 error rate:
        class   0.3
        sex     0
        race    0.05
person5 error rate:
        class   0.5
        sex     0.1
        race    0.05
person6 error rate:
        class   0.384615
        sex     0.128205
        race    0
person7 error rate:
        class   0.117647
        sex     0.117647
        race    0
```

3. Train votes rate:

```
weight_threshold :0.1 weight_adjust : 0.2 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.3 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.4 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.5 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.6 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.7 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.8 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 0.9 total error rate : 0.308705
weight_threshold :0.1 weight_adjust : 1 total error rate : 0.308705
adjust min :0.3 weight min : 0.5 total error rate : 0.269203
```

```
weight_threshold :0.5 weight_adjust : 0 total error rate : 0.286759
weight_threshold :0.5 weight_adjust : 0.1 total error rate : 0.281639
weight_threshold :0.5 weight_adjust : 0.2 total error rate : 0.275786
weight_threshold :0.5 weight_adjust : 0.3 total error rate : 0.269203
weight_threshold :0.5 weight_adjust : 0.4 total error rate : 0.27286
weight_threshold :0.5 weight_adjust : 0.5 total error rate : 0.275786
weight_threshold :0.5 weight_adjust : 0.6 total error rate : 0.280176
weight_threshold :0.5 weight_adjust : 0.7 total error rate : 0.285296
weight_threshold :0.5 weight_adjust : 0.8 total error rate : 0.294806
weight_threshold :0.5 weight_adjust : 0.9 total error rate : 0.302853
```

The train result arount minimum value is similar with parabola.

4. The final result:

```
threshhold      0.5
adjustment      +0.3
tatal error rate:
        class   0.238478
        sex     0.0504755
        race    0.0599854
person0 error rate:
        class   0.3
        sex     0.0666667
        race    0.133333
person1 error rate:
        class   0.210526
        sex     0.0263158
        race    0.0789474
person2 error rate:
        class   0.153846
        sex     0.0384615
        race    0.0769231
person3 error rate:
        class   0.37037
        sex     0.148148
        race    0.111111
person4 error rate:
        class   0.3
        sex     0
        race    0.05
person5 error rate:
        class   0.55
        sex     0.1
        race    0.05
person6 error rate:
        class   0.282051
        sex     0.102564
        race    0
person7 error rate:
        class   0.235294
```

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 this if we don't train LHH and use it directly:

```
person34 error rate:
        class   0.0357143
        sex     0
        race    0
person35 error rate:
        class   0
        sex     0
        race    0
person36 error rate:
        class   0.047619
        sex     0
        race    0
person37 error rate:
        class   0.130435
        sex     0.0434783
        race    0
person38 error rate:
        class   0
        sex     0
        race    0
```

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

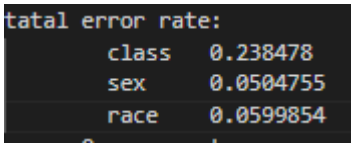
```
person21 error rate:
        class   0.83871
        sex     0.0967742
        race    0.0322581
person22 error rate:
        class   0.941176
        sex     0.294118
        race    0.235294
```

After train LHH, we can make it better:

```
person21 error rate:
        class   0.516129
        sex     0.0967742
        race    0.0322581
person22 error rate:
        class   0.411765
        sex     0.117647
        race    0
```

```
person34 error rate:
        class   0.0357143
        sex     0
        race    0
person35 error rate:
        class   0
        sex     0
        race    0
person36 error rate:
        class   0.047619
        sex     0
        race    0
person37 error rate:
        class   0.173913
        sex     0.0434783
        race    0
person38 error rate:
        class   0
        sex     0
        race    0
```

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.