

# 《计算机视觉》实验报告

姓名：冯俊佳 学号：23122721

## 实验 5 目标检测

### 一. 任务 1

#### a) 核心代码:

```
1.import cv2
2.import os
3.import numpy as np
4.
5.# 加载人脸分类器
6.face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_de
    fault.xml")
7.
8.# 加载 AdaBoost 分类器
9.boost = cv2.ml.Boost_create()
10.
11.# 提取 Haar 特征
12.def extract_haar_features(image, is_roi=False):
13.    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
14.    if not is_roi:
15.        faces = face_cascade.detectMultiScale(gray, scaleFacto
            r=1.1, minNeighbors=5)
16.        if len(faces) > 0:
17.            x, y, w, h = faces[0]
18.            gray = gray[y:y+h, x:x+w]
19.            # 如果没有检测到人脸,就保留整图 gray,用于负样本特征提取
20.            roi = cv2.resize(gray, (24, 24))
21.            hog = cv2.HOGDescriptor((24, 24), (8, 8), (4, 4), (8, 8),
                9)
22.            feature = hog.compute(roi)
23.            return True, feature.flatten()
24.# 提取正样本和负样本的特征
25.pos_features = []
26.neg_features = []
27.
28.# 准备正样本特征
```

```

29. pos_samples_folder = '/Users/feng/Desktop/ 计 算 机 视 觉
    /Week5/W5_Experiment/yale_face'
30. for filename in os.listdir(pos_samples_folder):
31.     if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG', 'bmp
        ')):
32.         image_path = os.path.join(pos_samples_folder, filename)
33.         image = cv2.imread(image_path)
34.         has_face, feature = extract_haar_features(image, is_ro
            i=False)
35.         if has_face:
36.             pos_features.append(feature)
37.
38. # 准备负样本特征
39. neg_samples_folder = '/Users/feng/Desktop/ 计 算 机 视 觉
    /Week5/W5_Experiment/non_face'
40. for filename in os.listdir(neg_samples_folder):
41.     if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG')):
42.         image_path = os.path.join(neg_samples_folder, filename)
43.         image = cv2.imread(image_path)
44.         has_face, feature = extract_haar_features(image, is_ro
            i=False)
45.         neg_features.append(feature)
46.
47. # 准备训练数据
48. features = np.vstack((np.vstack(pos_features), np.vstack(neg_f
        eatures)))
49. labels = np.hstack((np.ones(len(pos_features)), np.zeros(len(n
        eg_features)))).astype(np.int32)
50.
51. # 训练 AdaBoost 模型
52. boost.train(features, cv2.ml.ROW_SAMPLE, labels)
53.
54. folder_path = '/Users/feng/Desktop/ 计 算 机 视 觉
    /Week5/W5_Experiment/test_photo'
55. for filename in os.listdir(folder_path):
56.     if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG')):
57.         # 读取图像
58.         image_path = os.path.join(folder_path, filename)
59.         frame = cv2.imread(image_path)
60.
61.         size = frame.shape[:2]
62.         minSize_1 = (size[1] // 10, size[0] // 10) #计算最小尺寸

```

```

63.         face_rects = face_cascade.detectMultiScale(frame, scaleFactor=1.05, minNeighbors=2, minSize=minSize_1)
64.         # 在图像上绘制检测到的人脸
65.         for (x, y, w, h) in face_rects:
66.             # 提取当前人脸区域的特征
67.             roi = frame[y:y+h, x:x+w]
68.             has_face, feature = extract_haar_features(roi, is_roi=True)
69.             # 使用 AdaBoost 模型进行预测
70.             _, result = boost.predict(feature.reshape(1, -1))
71.             if result == 1: # 1 表示检测到人脸
72.                 cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 0), 2)
73.                 cv2.imshow('detection', frame)
74.                 cv2.waitKey(0)
75.
76. cv2.destroyAllWindows()

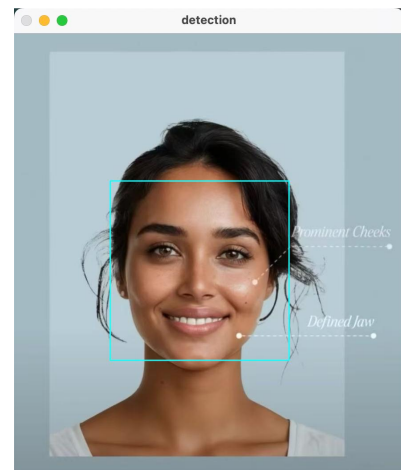
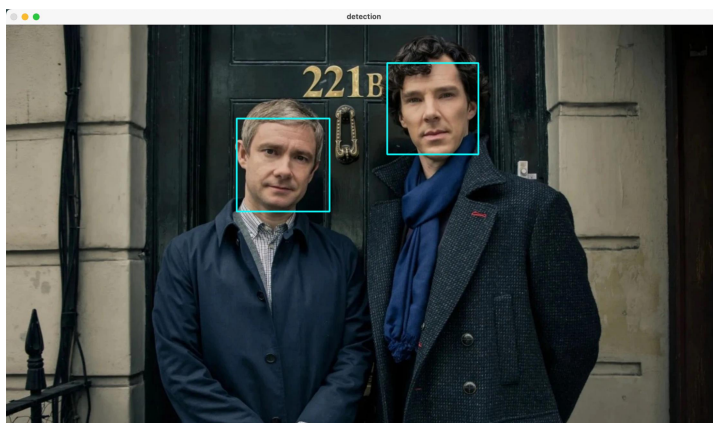
```

## b) 实验结果截图

测试图片集为自选图片。

训练的数据集：人脸图片集作为正样本，选取 yale 人脸数据集。

非人脸图片集选用电表图片作为负样本。





### c) 实验小结

本实验在没有进行训练 AdaBoost 模型的时候, 测试的时候会有同一个人两个框和非人类被框起来的情况, 但是在 AdaBoost 算法进行训练后, 测试的结果有明显提升。

本实验在处理作为正样本和负样本的图像 (也就是人像和非人像) 的时候, 一开始由于二者维度不匹配的情况, 导致特征长度为 0。后来进行修改, 在提取特征时, 将正样本和负样本的特征调整为相同的大小, 可成功运行并检测人脸。