

《计算机视觉》实验报告

姓名：孔馨怡 学号：22122128

实验 5 目标检测

一. 任务 1

a) 核心代码：

```
import cv2
import os
import numpy as np

# 加载人脸分类器
face_cascade =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

# 加载 AdaBoost 分类器
boost = cv2.ml.Boost_create()

# 提取 Haar 特征
def extract_haar_features(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    haar_cascade =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    faces = haar_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5)
    if len(faces) > 0:
        # 只取第一个检测到的人脸区域的特征
        x, y, w, h = faces[0]
        roi = gray[y:y+h, x:x+w]
        # 使用 cv2.resize() 函数将图像大小调整为指定大小
        roi = cv2.resize(roi, (24, 24))
        # 使用 cv2.HOGDescriptor() 对象的 compute 方法提取特征
        hog = cv2.HOGDescriptor((24, 24), (8, 8), (4, 4), (8, 8), 9)
        feature = hog.compute(roi)
        return True, feature.flatten() # 将特征展平成一维数组
    else:
        # 如果没有检测到人脸，则返回一个填充的特征数组
        return False, np.zeros((576,)) # 24x24 的灰度图像，特征向量大小为 576
```

```

# 提取正样本和负样本的特征
pos_features = []
neg_features = []

# 准备正样本特征
pos_samples_folder = 'faces'
for filename in os.listdir(pos_samples_folder):
    if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG')):
        image_path = os.path.join(pos_samples_folder, filename)
        image = cv2.imread(image_path)
        has_face, feature = extract_haar_features(image)
        if has_face:
            pos_features.append(feature)

# 准备负样本特征
neg_samples_folder = 'non_faces'
for filename in os.listdir(neg_samples_folder):
    if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG')):
        image_path = os.path.join(neg_samples_folder, filename)
        image = cv2.imread(image_path)
        has_face, feature = extract_haar_features(image)
        if not has_face:
            neg_features.append(feature)

# 准备训练数据
features = np.vstack((np.vstack(pos_features), np.vstack(neg_features)))
labels = np.hstack((np.ones(len(pos_features)),
np.zeros(len(neg_features))))

# 训练 AdaBoost 模型
boost.train(features, cv2.ml.ROW_SAMPLE, labels)

folder_path = 'photo'
for filename in os.listdir(folder_path):
    if filename.endswith(('.jpg', '.jpeg', '.png', 'JPG')):
        # 读取图像
        image_path = os.path.join(folder_path, filename)
        frame = cv2.imread(image_path)

        size = frame.shape[:2]
        minSize_1 = (size[1] // 10, size[0] // 10) # 计算最小尺寸
        face_rects = face_cascade.detectMultiScale(frame, scaleFactor=1.05,
minNeighbors=2,

```

```

minSize=minSize_1)

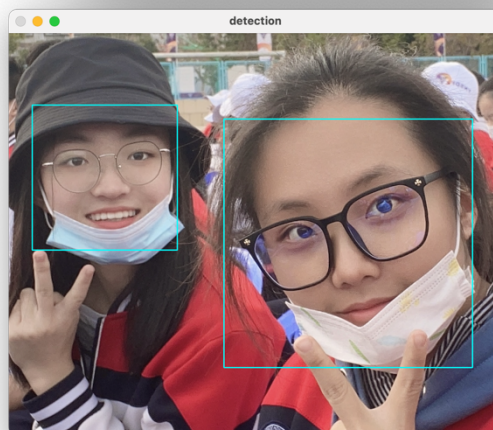
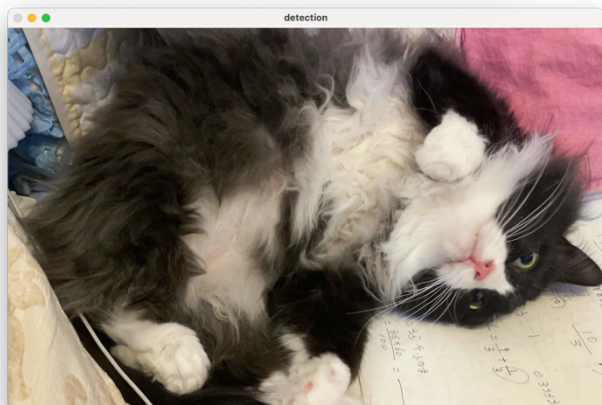
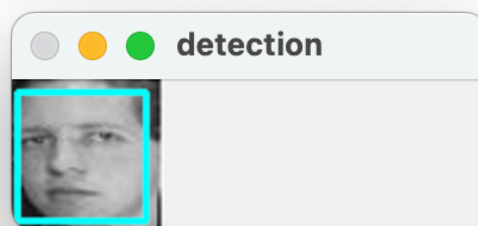
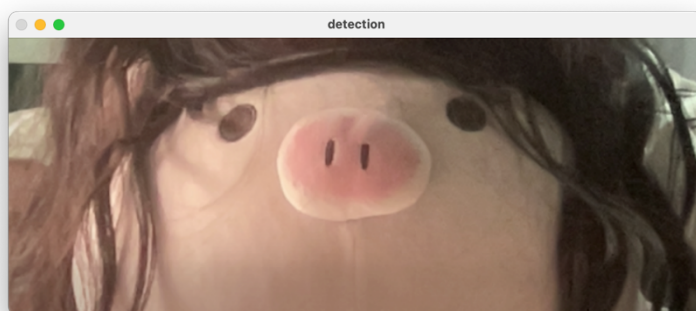
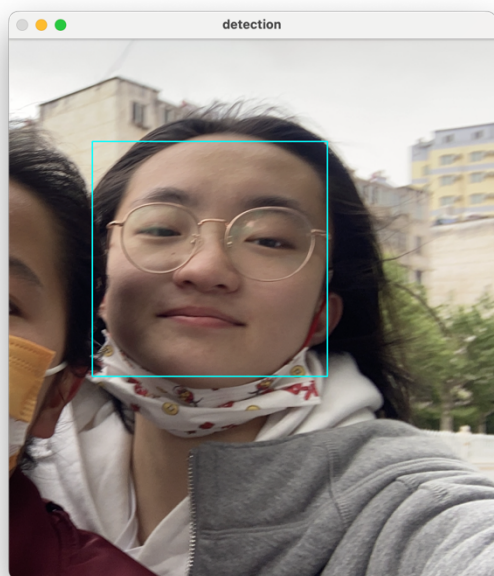
# 在图像上绘制检测到的人脸
for (x, y, w, h) in face_rects:
    # 提取当前人脸区域的特征
    roi = frame[y:y+h, x:x+w]
    has_face, feature = extract_haar_features(roi)
    # 使用 AdaBoost 模型进行预测
    _, result = boost.predict(feature.reshape(1, -1))
    if result == 1: # 1 表示检测到人脸
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 0),
2)

cv2.imshow('detection', frame)
cv2.waitKey(0)

cv2.destroyAllWindows()

```

b) 实验结果截图



TIPS:

测试图片集为自选;

训练的图片集:

人脸图片集作为正样本, 选取 yale 数据集

非人脸图片集—地标建筑作为负样本, 选取

<https://purl.stanford.edu/rb470rw0983> 网站的 lansmarks 中的 Query 为数据集。

c) 实验小结

下面是一些实验中踩的坑和心得:

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, 测试的结果好很多。

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, 可能会出现维度不匹配的情况, 可能是因为提取的特征尺寸不同, 或者某些图像中没有检测到人脸, 导致特征长度为 0。这种情况下可以尝试在提取特征时, 将正样本和负样本的特征调整为相同的大小, 并确保在没有检测到人脸时返回一个固定大小的特征向量。