

1. 任务一：人脸模糊处理

1.1 调用电脑自带摄像头

使用 Python 编程语言和 OpenCV 库调用电脑的摄像头。

将捕获的帧转换为图像，以便进行后续处理。

1.2 提出人脸部分

使用 OpenCV 自带的人脸检测算法，通过加载官方的 haarcascade_frontalface_default.xml 权重文件来检测图像中的人脸。

使用检测到的人脸的边界框来提取人脸区域。

1.3 在人脸部分进行模糊处理（只对人脸做处理）

对提取的人脸图像应用图像模糊（如高斯模糊）以减少细节和面部特征。

将处理后的人脸图像与原始图像中的相应位置进行替换。

1.4 代码

```
import cv2 as cv
import cv2
import numpy as np

frameWidth = 640
frameHeight = 480
#0 默认的是电脑自带的摄像头

cap = cv2.VideoCapture(0)
#cap.set 函数中的 3 为帧的宽度、4 为高度、10 为亮度
cap.set(3, frameWidth)
cap.set(4, frameHeight)
cap.set(10,150)

faceCascade= cv2.CascadeClassifier("face.xml")
```

```

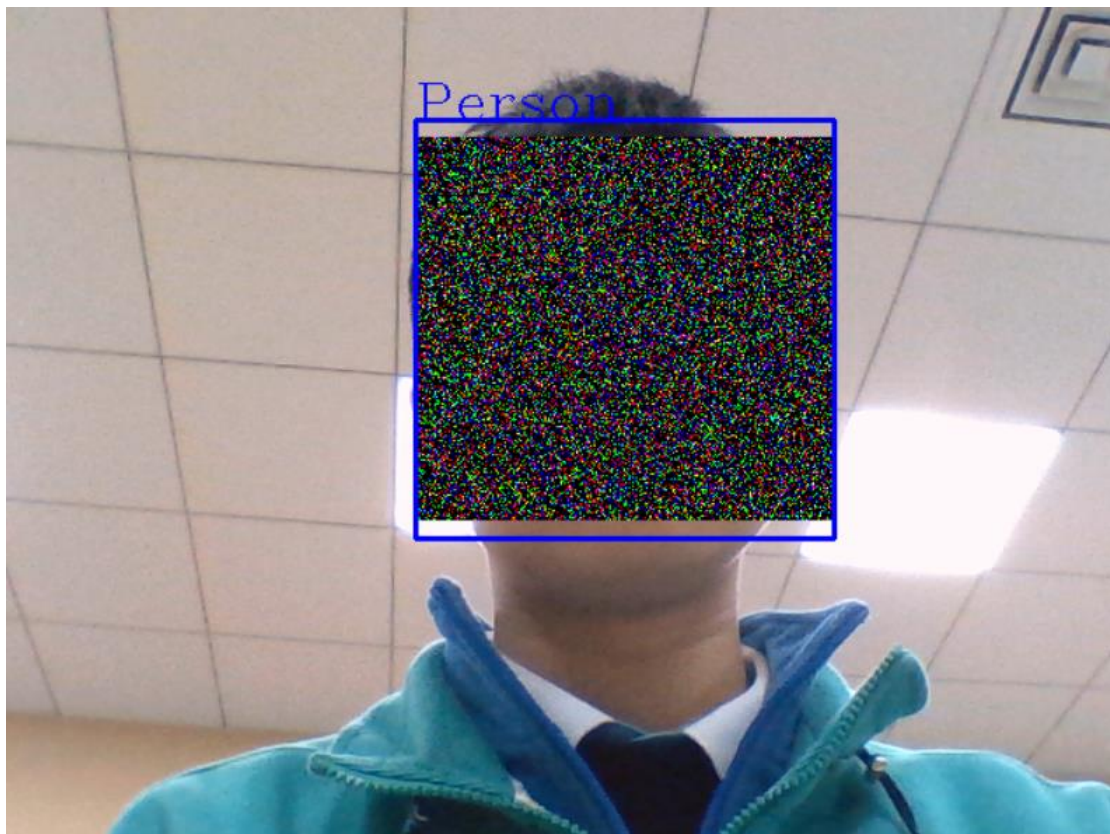
while True:
    success, img = cap.read()
    imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(imgGray,1.1,4)
    for (x,y,w,h) in faces:
        # 打码：使用高斯噪声替换识别出来的人眼所对应的像素值
        img[y+10:y+h-10,x:x+w,0]=np.random.normal(size=(h-20,w))
        img[y+10:y+h-10,x:x+w,1]=np.random.normal(size=(h-20,w))
        img[y+10:y+h-10,x:x+w,2]=np.random.normal(size=(h-20,w))

        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        text = "{}".format('Person')
        cv.putText(img, text, (x, y), cv.FONT_HERSHEY_COMPLEX,
color=(255,0,0), fontScale=1)

    cv2.imshow("Result", img)
#按下 键后break
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cv2.destroyAllWindows()

```

1.3 实验结果截图



2. 任务二：Flash 游戏辅助工具

2.1 打开黑色跳跳球游戏，使用 python 提取游戏画面

为了提取游戏画面，需要使用 Python 中的一些库，比如，可以使用 OpenCV 库来处理图像

2.2 使用图像匹配方法，识别黑色球，星星，终点三个物体

接下来，我们需要使用图像匹配的方法来识别黑色球，星星和终点。其中最简单的方法是使用模板匹配技术，也就是将要识别的物体作为模板，在游戏画面中搜索匹配的区域。

2.3 在三个目标物体的周围画框并标上坐标

读取物体的坐标位置，通过设置文本内容的详细，讲每个物体的信息画在区域中。

2.3 实验代码

```
import cv2 as cv
import cv2
import numpy as np
import os
import time
from windowcapture import WindowCapture
import pyautogui

# Change the working directory to the folder this script is in.
# Doing this because I'll be putting the files from each video in their
# own folder on GitHub
os.chdir(os.path.dirname(os.path.abspath(__file__)))

def pressSpaceButton():
    time.sleep(0.15)
    pyautogui.keyDown('space')
    pyautogui.keyUp('space')

def match(haystack_img, needle_img, threshold, name):
```

```

    result = cv.matchTemplate(haystack_img, needle_img,
cv.TM_SQDIFF_NORMED)

# I've inverted the threshold and where comparison to work with
TM_SQDIFF_NORMED
    # threshold = threshold
# The np.where() return value will look like this:
# (array([482, 483, 483, 483, 484], dtype=int32), array([514, 513, 514,
515, 514], dtype=int32))
    locations = np.where(result <= threshold)
# We can zip those up into a list of (x, y) position tuples
    locations = list(zip(*locations[::-1]))
    print(locations)
    print(len(locations))
    # time.sleep(10)

    if locations:
        print('Found needle.')

        needle_w = needle_img.shape[1]
        needle_h = needle_img.shape[0]
        line_color = (0, 255, 0)
        line_type = cv.LINE_4

        # Loop over all the locations and draw their rectangle
        for loc in locations:
            # Determine the box positions
            top_left = loc
            bottom_right = (top_left[0] + needle_w, top_left[1] +
needle_h)

            # print(top_left)
            # print(bottom_right)

            # Draw the box
            cv.rectangle(haystack_img, top_left, bottom_right,
line_color, line_type)

            # print(top_left, bottom_right)
            # time.sleep(20)
            text = "{}".format(name)
            cv.putText(haystack_img, text, top_left,
cv.FONT_HERSHEY_COMPLEX, color=(255,0,0), fontScale=1.2)
            return haystack_img
        #cv.imwrite('result.jpg', haystack_img)

```

```

        else:
            return haystack_img

# initialize the WindowCapture class
wincap = WindowCapture('黑色跳跳球')

needle_img = cv.imread('ball.png', cv.IMREAD_COLOR)
star_img = cv.imread('star.png', cv.IMREAD_COLOR)
final_img = cv.imread('final.png', cv.IMREAD_COLOR)

loop_time = time.time()
while(True):

    # get an updated image of the game
    screenshot = wincap.get_screenshot()
    img = match(screenshot, needle_img, 0.3, name='ball')
    img = match(img, star_img, 0.05, name='star')
    img = match(img, final_img, 0.1, name='final')

    # img = match(screenshot, needle_img)
    cv.imshow('Computer Vision', img)
    #pressSpaceButton()
    # debug the loop rate
    print('FPS {}'.format(1 / (time.time() - loop_time)))
    loop_time = time.time()

    if cv.waitKey(1) == ord('q'):
        cv.destroyAllWindows()
        break

print('Done.')
```

2.3 实验结果截图

