1. Introduction

The aim of the laboratory was to familiarize with OpenCV in Python – open source real-time optimized Computer Vision library. First part was to track faces and second to recognize car plates.

What are the difference between Image Processing and Computer Vision?

Image Processing is focused on processing raw input image to enhance them or prepare for further operations, while Computer Vision aims to extract information from input to better understand them (for e.g. recognition). Image Processing is often necessary first step begore Computer Vision. Examples of Image Processing are rescaling, blurring, filtering while Computer Vision – object detection, face recognition.

Where Computer Vision systems are used nowadays?

Computer Vision systems are widely used for example in:

- Transport – smart traffic lights, self-driving cars
- Healthcare – image recognition for X-rays, diagnosis
- Manufacturing – samples inspection, warehouse automatization
- Agriculture – crop monitoring
- Security – movement, face detection

2. Results

2.1. Face Tracking

Script was mostly finished and running properly, task required to add coordinates. It was achieved with cv2 function

```
cv2.putText(img, str(x) + '; ' + str(y), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0,0,0), 5)
```

As seen below, face was properly recognized, no matter if on another display or not. Also program does not have problem with recognizing multiple faces at once. It is worth noting that sometimes ear was recognized as a face for a split second.

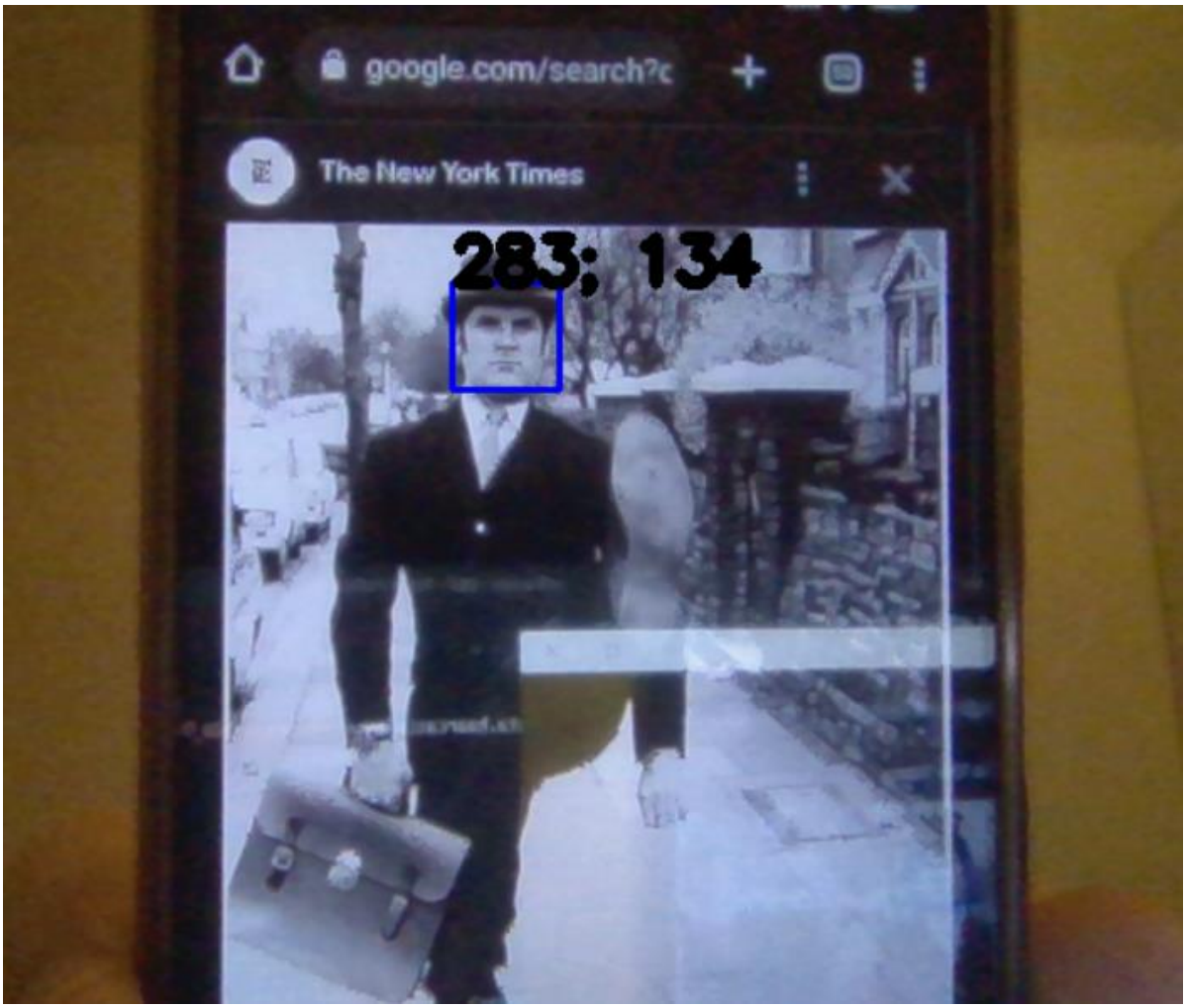

Figure 1 Face recognition with added coordinates

2.3. Plate recognition

Script again was ready and running, the task was to visualize steps in plate recognition.

Firstly, we read and resize image

Figure 2 resized image

Next, we convert it to grayscale to simplify process (RGB values are not needed to find contours)

Figure 3 Image in grayscale

Then, image is blurred with Bilateral Filter to reduce noise

Figure 4 Blurred image

Next, Canny algorithm detects edges

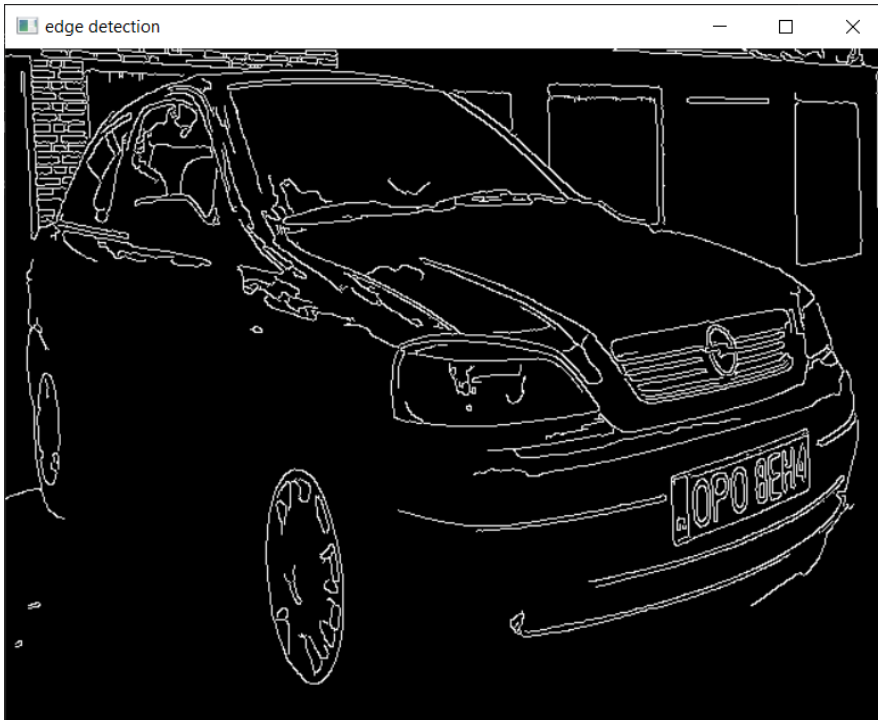

Figure 5 Edge detection

Lastly, we look for 4 contours (rectangle) assuming that the largest one can be our plate

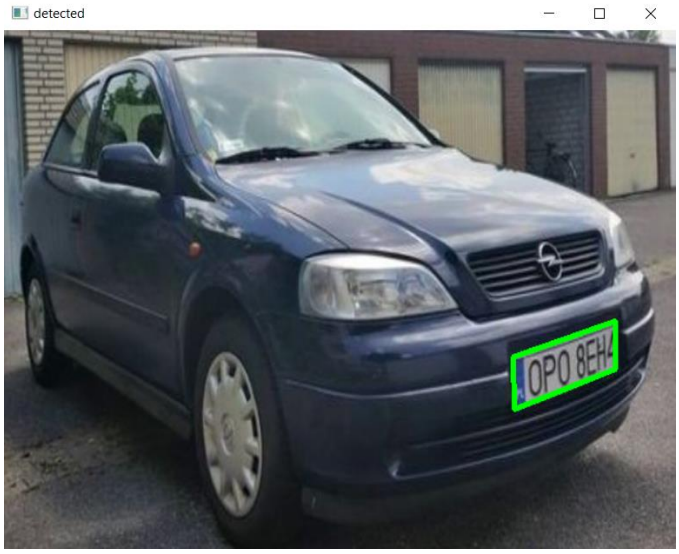

Figure 6 Plate detected

Additionally, we can crop image

Figure 7 Cropped image and result for Car1

Lastly parameters were tuned to try and achieve similar results for rest of cars (for blur, Canny, shapes etc). Unfortunately for Car2 and Car3 results were not satisfactory – plate was not recognized, probably angle of photo is the cause. Plate was recognized for Car4 – it required looking for more than 4 point contour.

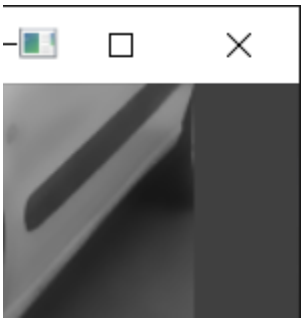

Figure 8 Not recognized for Car2

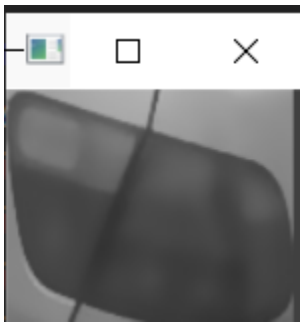

Figure 9 Not recognized for Car3

Figure 10 Recognition for Car4