WEB CAMERA VIDEO PROCESSING

JANUARY 11, 2022

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
Sas-Miresan Andrei

Web camera video processing

Table of Contents

1. Introduction

1.1. Context

The goal of this project is to design a program that identifies the position of moving objects in the frames of the web camera live feed video.

1.2. Specification

The program will be created in Python 3.7.8. It will be able to start the web camera of the device it runs on and process the video the web camera is capturing

1.3. Objectives

The program should detect the movements of the objects present in the camera view and should highlight it on the camera feed for the user to notice. The area of the moving object should be traced by a green rectangle and the moments of the movement should be recorded in a separate file.

2. Bibliographic study

With the proliferation of low-cost digital cameras able to shoot video, it is possible to use the output of such a camera to detect motion in its field of view using software. This solution is particularly attractive when the intent is to record video triggered by motion detection, as no hardware beyond the camera and computer is needed. Since the observed field may be normally illuminated, this may be considered another passive technology. However, it can also be used together with near-infrared illumination to detect motion in the dark, that is, with the illumination at a wavelength undetectable by a human eye.

The background of the video stream is largely static and unchanging over consecutive frames of a video. Therefore, if we model the background, we monitor it for substantial changes. If there is a substantial change, it can be detected — this change normally corresponds to motion on the video.

3. Analysis

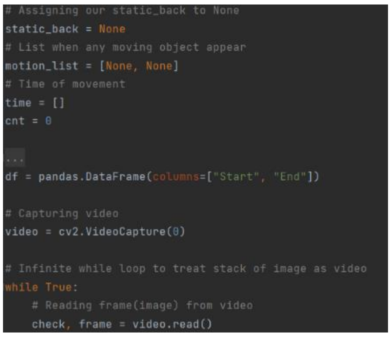
By analysing the camera detection movement we see that in order to design the program we need to save different frames and compare them between each other to detect the differences between the frames to see if anything changed.

Also we need to change the frames recorded by the camera to grayscale to reduce the errors that could appear from the colours of objects in the frame and the light levels of the video captured.

4. Design & Implementation

To create the code we mostly use the functions provided by the libraries OpenCV (import cv2) and Pandas (import pandas).

The cv2 library offers the function VideoCapture() which lets us record with the web camera. After the video is started to be captured we can transform it using read() function in frames that we can process them as we need to detect the motion.



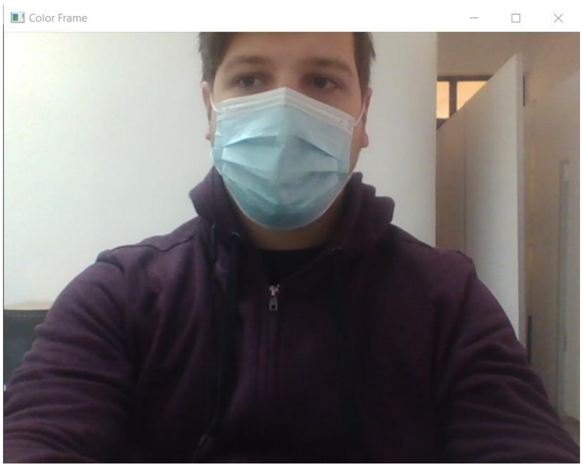
```
# Assigning our static_back to None
static_back = None
# List when any moving object appear
motion_list = [None, None]
# Time of movement
time = []
cnt = 0

...
df = pandas.DataFrame(columns=["Start", "End"])

# Capturing video
video = cv2.VideoCapture(0)

# Infinite while loop to treat stack of image as video
while True:
    # Reading frame(image) from video
    check, frame = video.read()
```

We take than the frames, transform them to grayscale using 2 cv2 functions (cvtColor(), GaussianBlur()) and at the beginning and every 100 frames we save the current frame as the static background frame (static_back) to which we compare the other frames. After that we compare the frame with the static background using a difference frame which using the function absdiff().

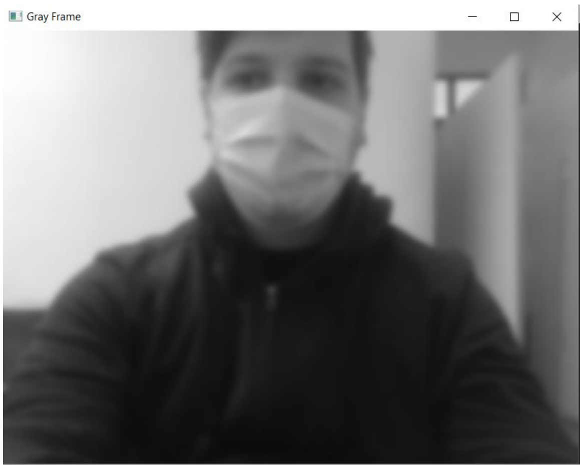
Using the data from the difference frame we than try to detect how big the differences between the frames are by creating a threshold frame which is strictly binary black and white and transforms area of white and grey which pass the threshold size area set into more clear and bigger areas that than can be detected and fit into rectangles using chain approximations. We than print the rectangles on the video where the motion is detected and we record the moment of the motion in a file to be set as a record of the movement recording.

5. Testing and validation

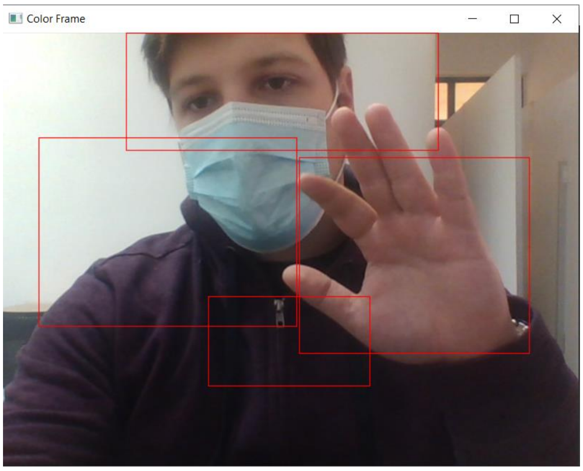
What the camera usually sees:



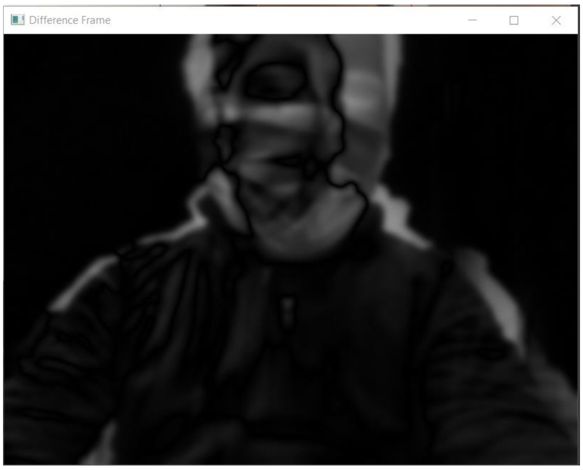
The frame from the camera transformed to grayscale:



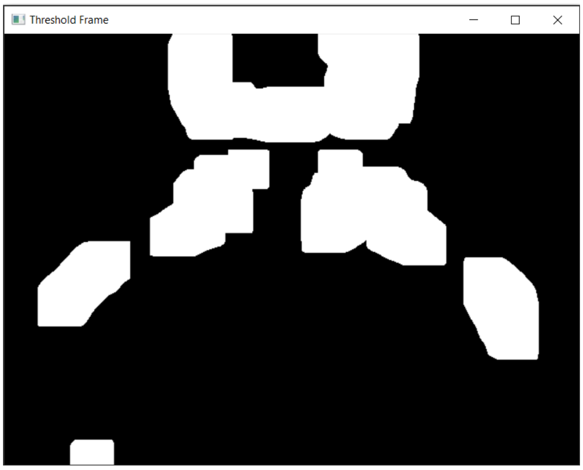
The detection of movement:



The difference frame between the background frame and the current frame:



The final threshold frame transforming the small grey difference areas from the difference frame into clear white zones for the motion areas:



6. Conclusions

The project was challenging but also helped me learn a lot of new things. It helped a lot for me to develop my skills with python. It also let me work with two libraries that I would rarely use and so I discovered some of their functionalities.

7. Bibliography

[1] https://en.wikipedia.org/wiki/Motion_detector

[2] https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/