

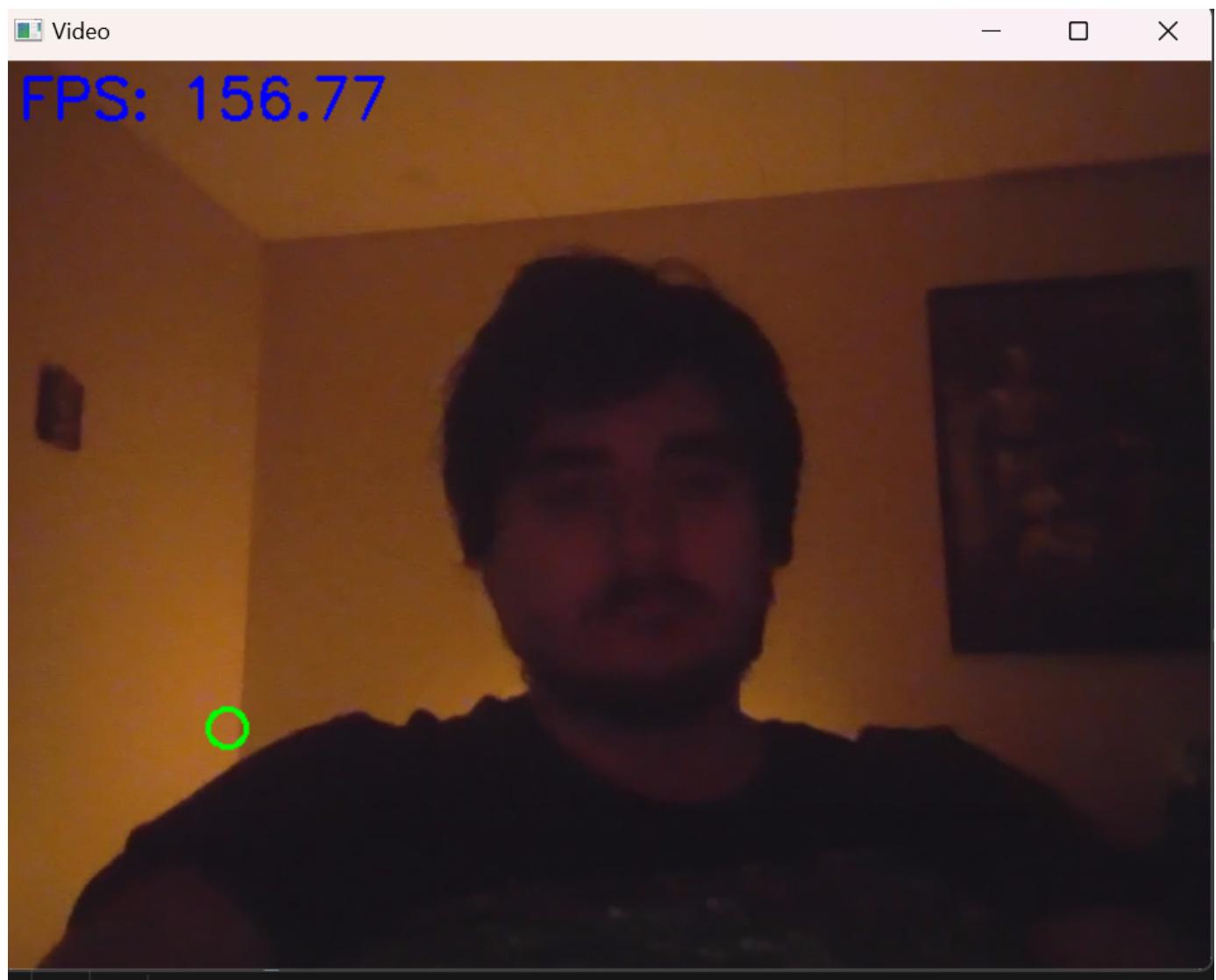
ass1

Assignment 1 Vilhjálmur

Code is available at: <https://github.com/vilhjalmurt19/cvass1>

Demonstration video: <https://youtu.be/JGkByjFsEag>

Part 1



Screenshot of an image capture

Part 2

See demonstration video

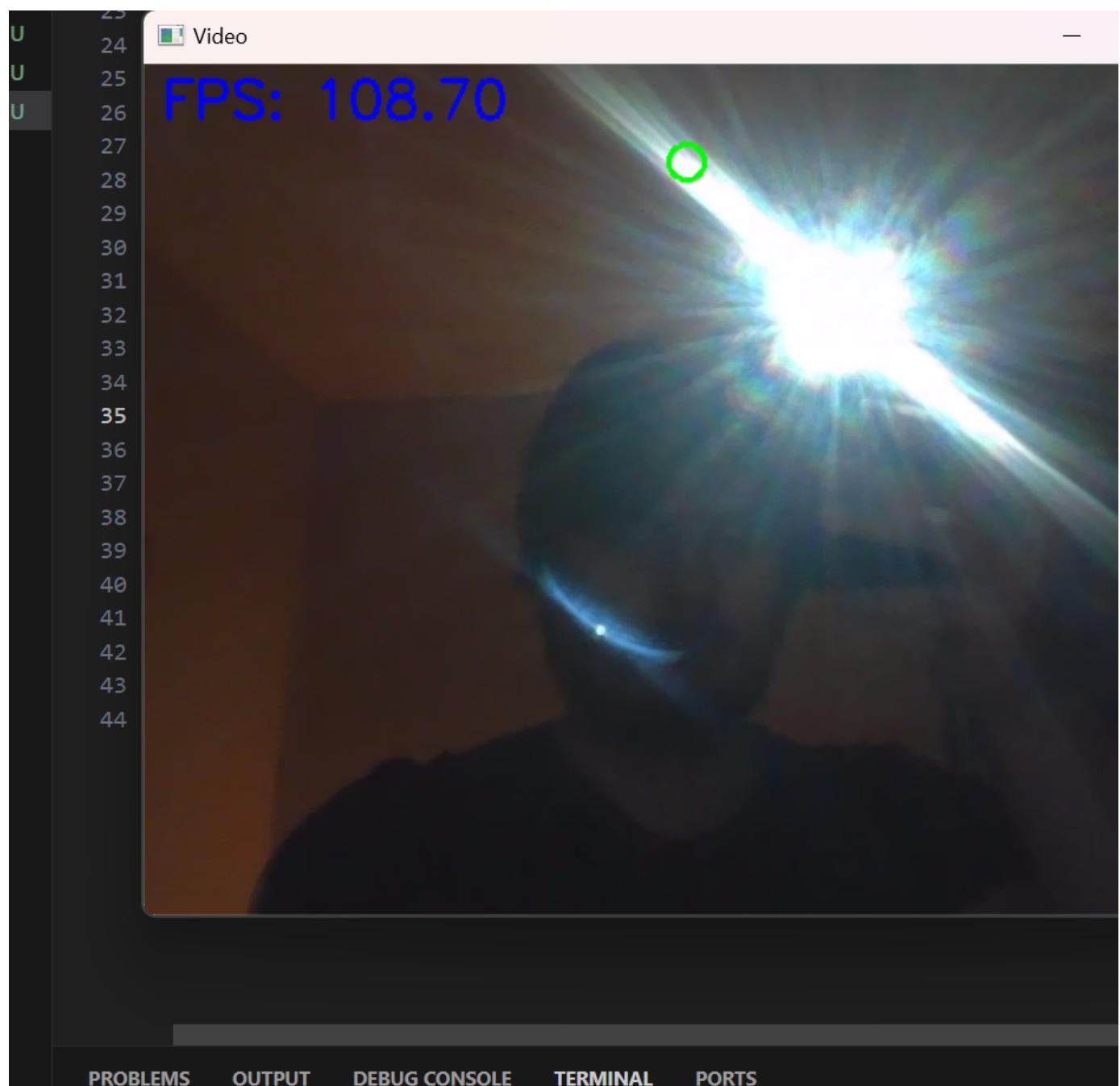
Part 3

See github code for specific implementation

```
Processing time for one frame: 0.0020 seconds
Processing time for one frame: 0.0758 seconds
Processing time for one frame: 0.0015 seconds
Processing time for one frame: 0.0014 seconds
Average processing time: 0.0263 seconds
```

Results of realtime processing time and average processing time printed out to terminal

Part 4



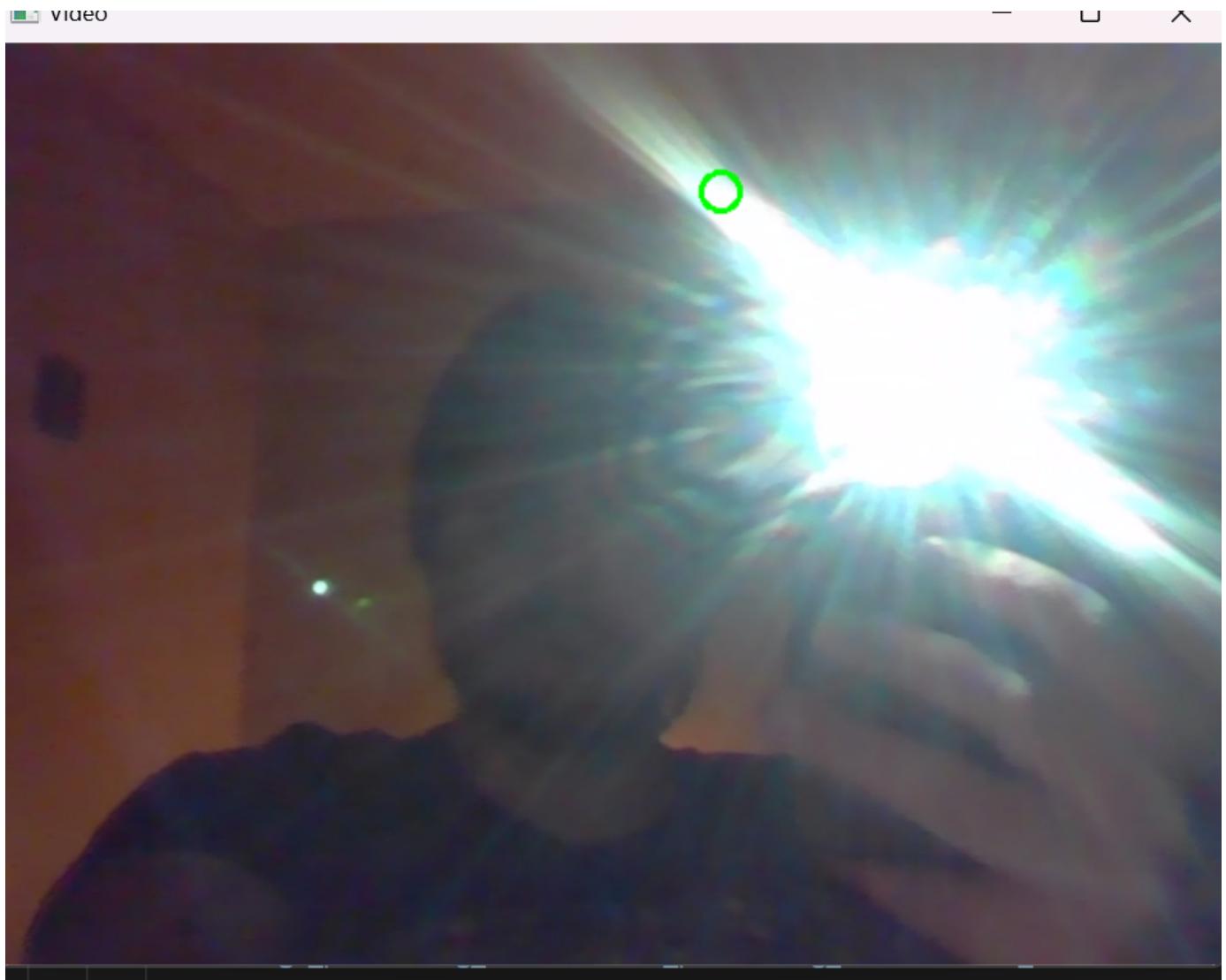
Bright spot located marked with a green circle

Part 5

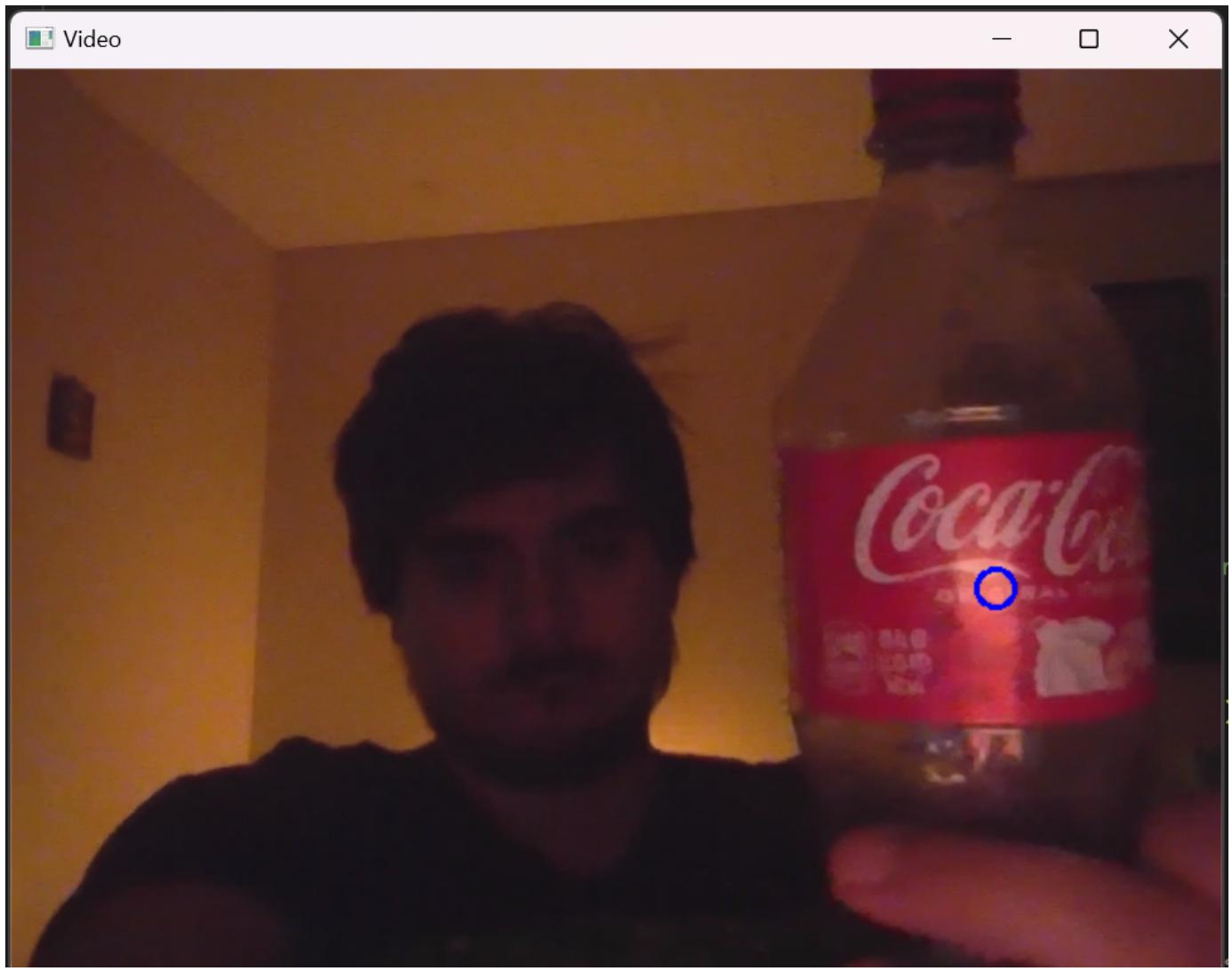


Reddest spot located a cocacola bottle was used. The reddest spot in the image was defined as the pixel location with the highest intensity value in the red channel of an RGB image. which is clear as the frame also has a reddish light source in the background yet the cocacola bottle label was marked

Part 6



Brightest spot loop implementation



Reddest spot loop implementation

Part 7

See demonstration video.

the app droidcam was used and the ip and port was simply parsed as a http url as the camera index

Questions

The processing time for one video frame or image.

[PROBLEMS](#) [OUTPUT](#) [DEBUG CONSOLE](#) [TERMINAL](#) [PORTS](#)

```
Processing time for one frame: 0.0014 seconds
Processing time for one frame: 0.0029 seconds
Processing time for one frame: 0.1197 seconds
Processing time for one frame: 0.0020 seconds
Average processing time: 0.0288 seconds
PS C:\Users\vilhj\OneDrive\Documents\School\cv\cvass1> █
```

The average processing time for one frame was calculated to be 0.0288 seconds

How does the processing time change when you add the bright spot detection?

[PROBLEMS](#) [OUTPUT](#) [DEBUG CONSOLE](#) [TERMINAL](#) [PORTS](#)

```
Processing time for one frame: 0.0027 seconds
Processing time for one frame: 0.0015 seconds
Processing time for one frame: 0.0013 seconds
Processing time for one frame: 0.0746 seconds
Average processing time: 0.0236 seconds
PS C:\Users\vilhj\OneDrive\Documents\School\cv\cvass1> S █
```

the average processing time was calculated to be 0.0236 seconds which is higher than the previous experiment time. This may be an insignificant difference.

Is the processing time identical when you do not display the image?

[PROBLEMS](#) [OUTPUT](#) [DEBUG CONSOLE](#) [TERMINAL](#) [PORTS](#)

```
Processing time for one frame: 0.0022 seconds
Processing time for one frame: 0.0022 seconds
Processing time for one frame: 0.0032 seconds
Processing time for one frame: 0.0701 seconds
Average processing time: 0.0230 seconds
PS C:\Users\vilhj\OneDrive\Documents\School\cv\cvass1> █
```

Ctrl+K to generate a command

when commenting out cv2.imshow a slightly faster processing time is seen

This can generally be attributed to a slight processing overhead of displaying an image but ultimately the processing time comes down to

- The performance of your computer's graphics hardware.
- The resolution of the video frames.

How does your for-loop implementation compare to the built-in function?

```
Processing time for one frame: 0.0789 seconds  
Processing time for one frame: 0.0756 seconds  
Processing time for one frame: 0.0771 seconds  
Processing time for one frame: 0.0774 seconds  
Average processing time: 0.0641 seconds  
PS C:\Users\vilhj\OneDrive\Documents\School\cv\cv
```

Average processing time for the reddest spot loop implementation takes significantly longer by 0.0400ms +

```
Processing time for one frame: 0.1001 seconds  
Processing time for one frame: 0.0394 seconds  
Processing time for one frame: 0.0427 seconds  
Processing time for one frame: 0.0418 seconds  
Average processing time: 0.0533 seconds  
PS C:\Users\vilhj\OneDrive\Documents\School\cv\cvass1>
```

The same can be said for the brightest spot loop implementation. From the figure it shows an average processing time increase of + 300ms

Moving your hand in front of the camera, estimate the latency between image capture and display.

Moving my hand in front of the camera there seems to be no visible latency and is hard to estimate

Is the latency different when capturing from a mobile phone?

The latency was significant the framrate dropped to around 17 fps and the estimated latency was about 2 seconds when using a digital clock to measure