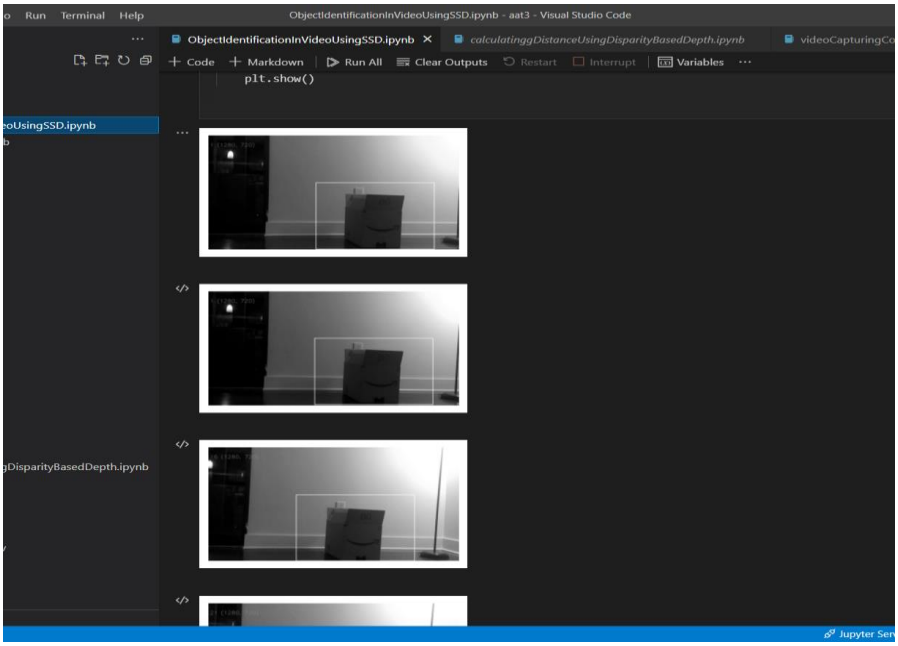
3rd Assignment

1. Capture a 10 sec video footage using the camera. The footage should be taken with the camera in hand and you need to pan the camera slightly from left-right or right-left during the 10 sec duration. Pick any image frame from the 10 sec video footage. Pick a region of interest corresponding to an object in the image. Crop this region from the image. Then use this cropped region to compare with randomly picked 10 images in the dataset of 10 sec video frames, to see if there is a match for the object in the scenes from the 10 images. For comparison use sum of squared differences (SSD) or normalized correlation.

   I have taken the video using the rgbVideo.py code from Depth AI and saved the video. Then an image is taken randomly from the video and have cropped an object(cartoon box) and then compared with rest of the frames in the video to see if the object(cartoon box) is available or not.

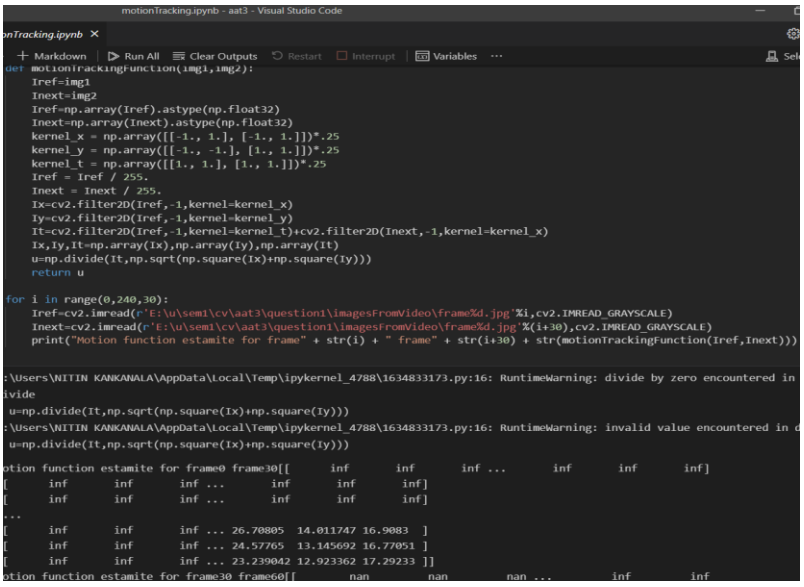
Cropped image in the video (a cartoon box)



Output

The shape of the cartoon is perfectly identified in the rest of the frames in the video. I have shown the output with rectangle identifying the box in other frames.

2. Implement the motion tracking equation from fundamental principles. Select any 2 consecutive frames from the set from problem 1 and compute the motion function estimates. Conduct image registration to realign the frames. Repeat test for all consecutive pairs of frames in the video.

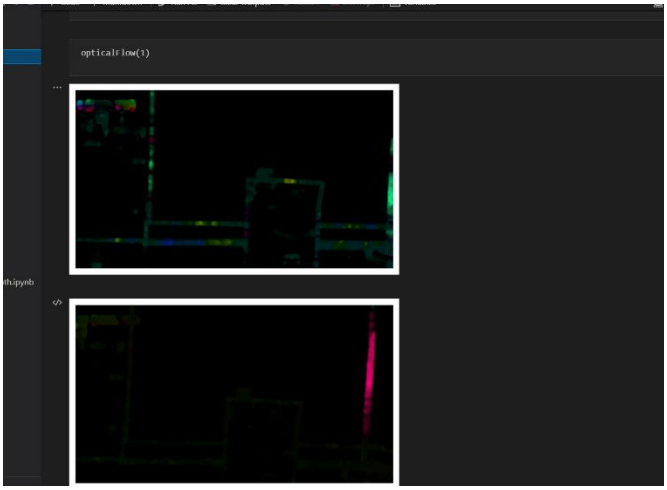


The motion function estimate for frames i, i+30

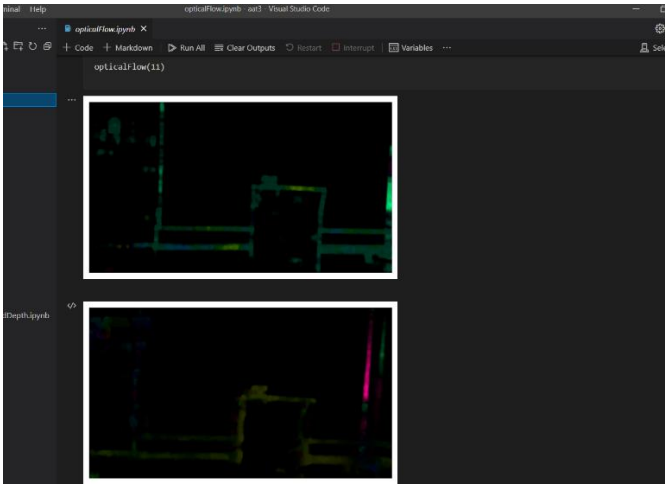
3. For the video (problem 1) you have taken, plot the optical flow vectors on each frame. (i) treating every previous frame as a reference frame (ii) treating every 11th frame as a reference frame (iii) treating every 31st frame as a reference frame

Optical flow for every 1st frame, 11th frame and 31st frame are recorded as outputs which can be seen in the IPYNB file in github which shows the optical movement of objects in the frames in the video captured.

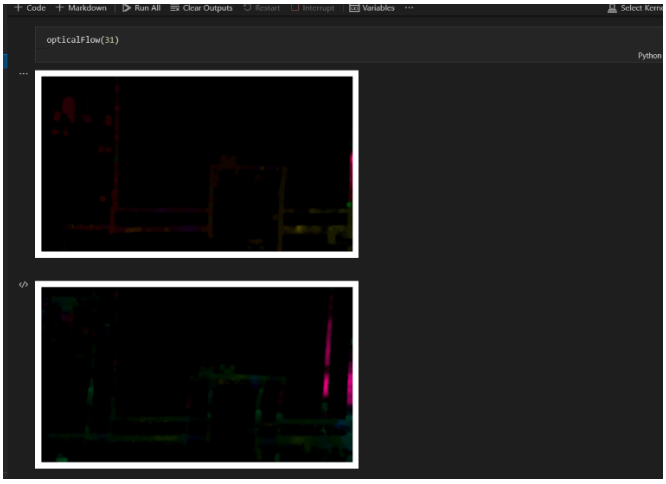
The video taken is a slow video which covers only part of my room and in the video there are two objects a cartoon box and a lamp stand. These two objects are clearly identified and the flow is clear to viualize.



Reference as previous frame

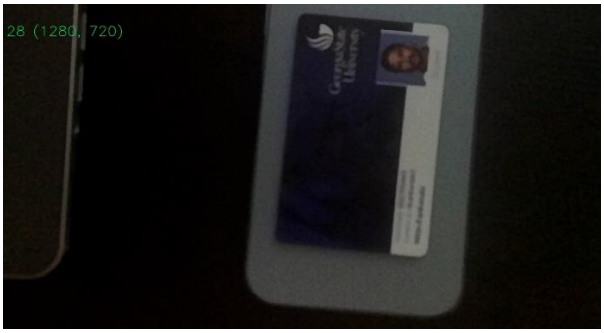


Reference as every previous 11th frame

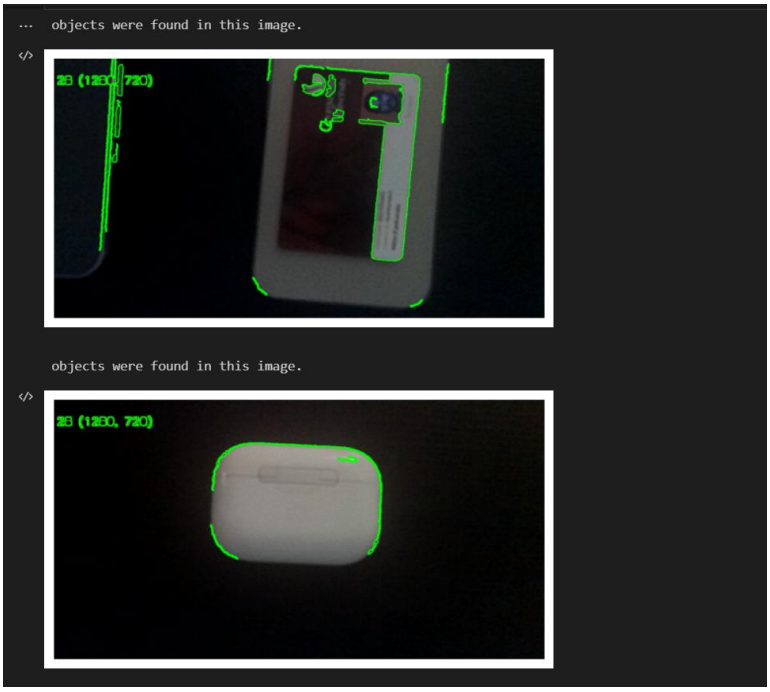


Reference as every previous 31st frame

4. Implement a feature based object detection application (from scratch) for detecting an object of your choice. Test it for at least 2 differently looking objects. Validate your results by testing against built-in object detection functions/code in MATLAB/OpenCV



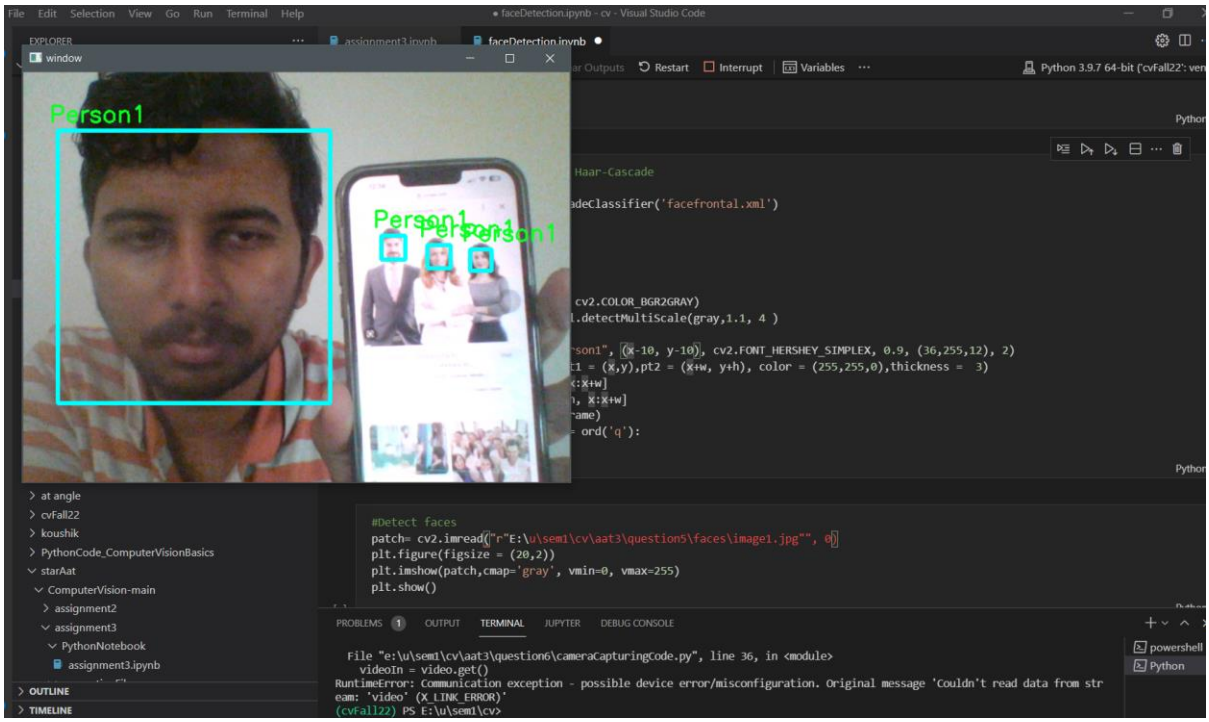
Airpods IDCard



Identified Objects

5. Implement a real-time face tracking application that will detect as many faces there are with a scene, and identify the person's facial region (draw a bounding box) whose is sought for by the user (you must ask for a person in your application and it should show a bounding box over the person of interest). Validate at least 20 times and present the recognition performance metrics (accuracy, precision, recall and Intersection over Union (IoU)).

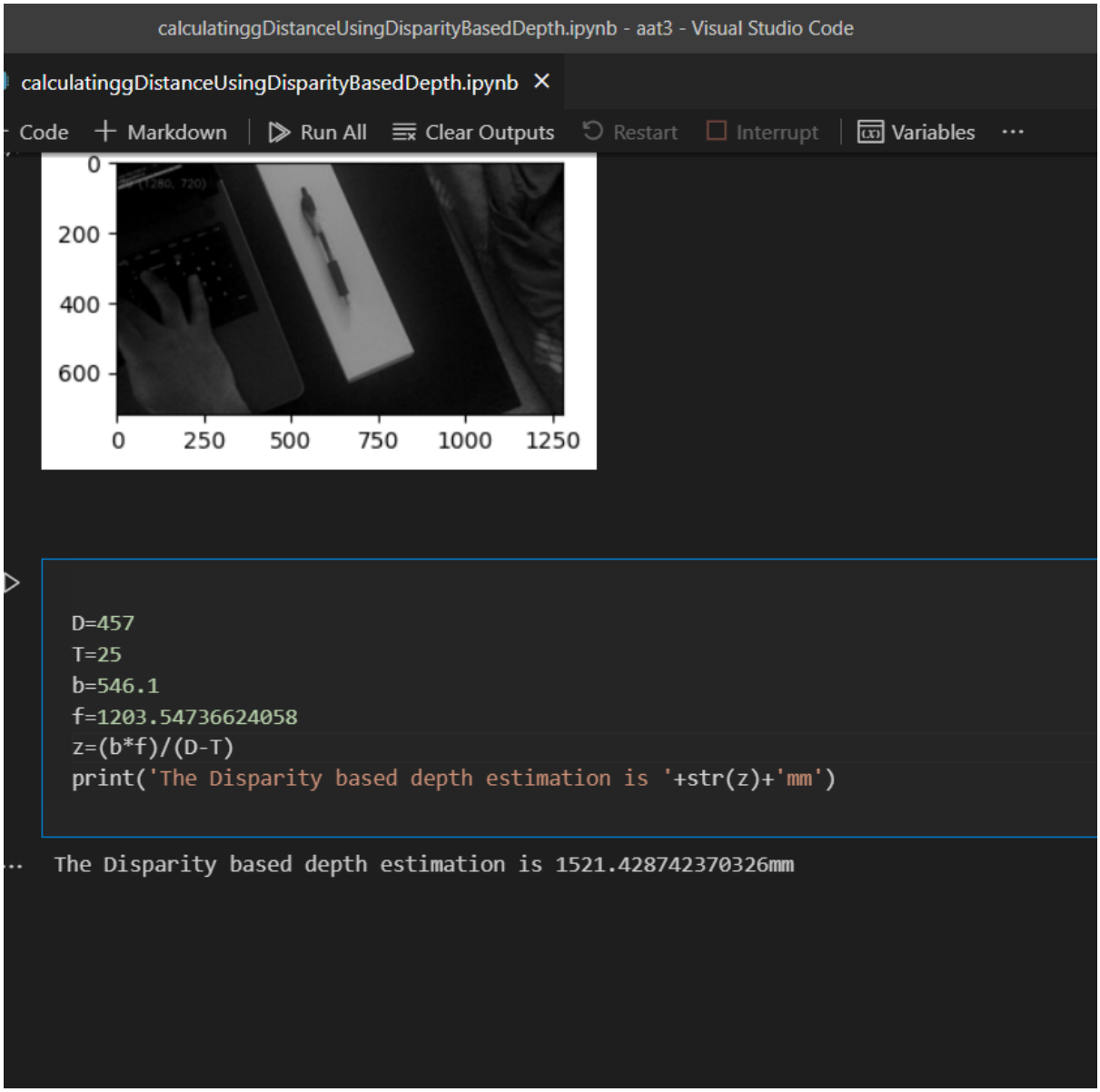
For the 20 images sample taken we have calculated the Accuracy, precision, recall and IOU. The results are as follows.



Multiple face detection sample image 1

6. Fix a marker on a wall or a flat vertical surface. From a distance D, keeping the camera stationed static (not handheld and mounted on a tripod or placed on a flat surface), capture an image such that the marker is registered. Then translate the camera by T units along the axis parallel to the ground (horizontal) and then capture another image, with the marker being registered. Compute D using disparity based depth estimation in stereo-vision theory. (Note: you can pick any value for D and T. Keep in mind that T cannot be large as the marker may get out of view. Of course this

depends on D)



The disparity based depth calculated is 1521mm