CV Intern Assignment – Amrith Coumaran

Problem 1 and 2:

First problem was to detect the faces in the image. face_det.py detects all the faces in an image. It only takes an image as input and isn't programmed for video input. The model used for detection is a pre-trained SSD Caffe Model and the required files for loading the model are attached with the folder. I have also attached some test files under dataset folder.

I have marked the primary face (face at focus) with a green bounding box and the remaining faces (if any) with a red bounding box. Method I used to select the primary face was based on simple distance calculation. I calculated the distance between the center of the input image and center of all bounding boxes that were detected and chose the box with minimal distance as the primary face.

Distance formula is given by,

dist = [(x2-x1)^2 – (y2-y1)^2]^0.5

Problem 3:

The face_landmarks.py detects the facial landmarks of all the faces in the image. I have used a pre-trained model for facial landmarks detection. First the image is passed through a the face detection (from previous problem) process and the resultant face ROI is first preprocessed and converted into a square size(128x128) before passing into a facial landmark detection model. Facial landmarks detected are the outline of the face, eyes, nose and mouth with a total of 68 key points. This only takes in an image as input and outputs the landmarks of all the faces in this image and not just the primary one.

Problem 4 and 5:

This is the final piece where all the three concepts come into place,

1) Face Detection (Just the Primary one)

2) Facial Landmarks Detection (Just main 6 key points for pose estimation)

3) Head pose Estimation

For head pose detection we need 6 key points: Nose tip, Chin, Left eye corner, Right eye corner, Left mouth corner and Right mouth corner. These are 2-D points and we also need 3-D locations of the same points. Since it's hard to get the 3D model, we take 3D locations of a few points in some arbitrary reference frame. Here I used,

1) Tip of the nose : (0.0, 0.0, 0.0)
2) Chin : (0.0, -330.0, -65.0)
3) Left corner of the left eye : (-225.0f, 170.0f, -135.0)
4) Right corner of the right eye : (225.0, 170.0, -135.0)

5) Left corner of the mouth : (-150.0, -150.0, -125.0)

6) Right corner of the mouth : (150.0, -150.0, -125.0)

We also need camera parameters. That is we need focal length of the camera, the optical centre in the image and the radial distortion parameters. We can approximate optical centre by the centre of the image, focal length by the width of the image in pixels and let's assume there are no distortions.

We use opencv function solvePnP to estimate pose to get both rotational as well as translational vectors. We need project a 3D point (0,0,1000) onto the image plane and use this to draw a line sticking out of the nose.

The function takes in live stream as input and not any video or image. The output vector coordinates are printed in the output.

Additional Problems:

1) If the head goes out of focus and comes back in, will your algorithm start tracking again?

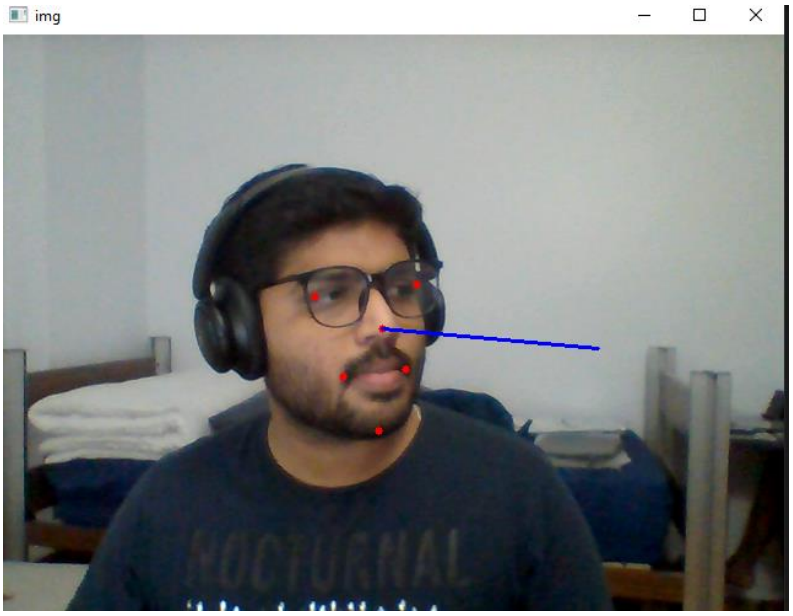
   Yes, I have designed the function in such a way that even if doesn't detect any face in the image, it still keeps running.

2) How are you prioritizing tracking for objects in focus?

   I am just selecting the bounding box of the face closest to the centre of the image in case of multiple images. If there is any only one face in the frame, then the detection process will run on the detected face.

3) Are you able to track with foreign objects like headphones, spectacles, etc?

   Yes I was able to detect with glasses and spectacles.



4) If there is occlusion (example: if someone is drinking from a bottle) will the tracking stop?

While drinking from a bottle, it was still detecting but the detection was not accurate. There was some distortion.