Embody
Computer Vision Engineering Internship Assignment
(Time Period: 3 Days)

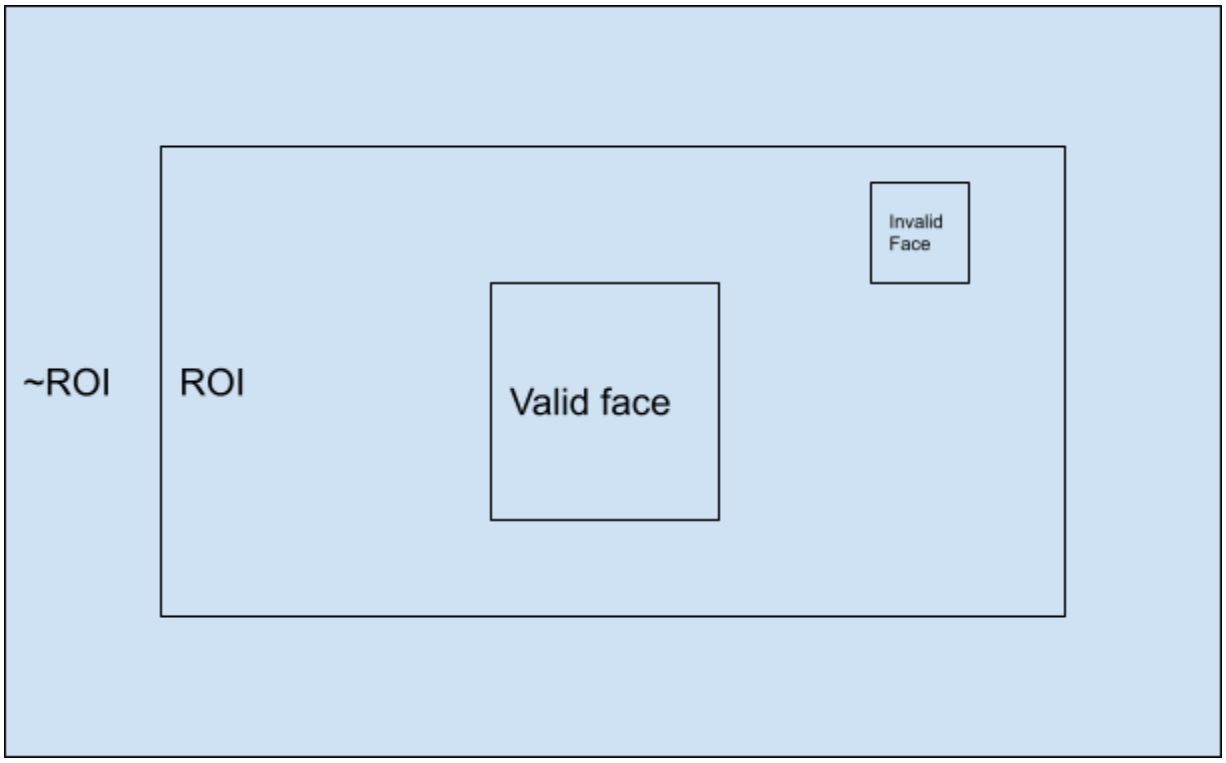
Please solve the following problems. Problem 5 should be your eventual goal, however, try to solve as many intermediary problems as possible.

Some pointers to follow:

- You should implement this solution in either C/C++ or python.
- Create a project out of the solution.
- Reuse as much code as possible.
- Write a README for the project with clear instructions on how to compile and run the program.
- For C/C++, libraries and make files will be required.
- For python, a pip requirements file will be required.

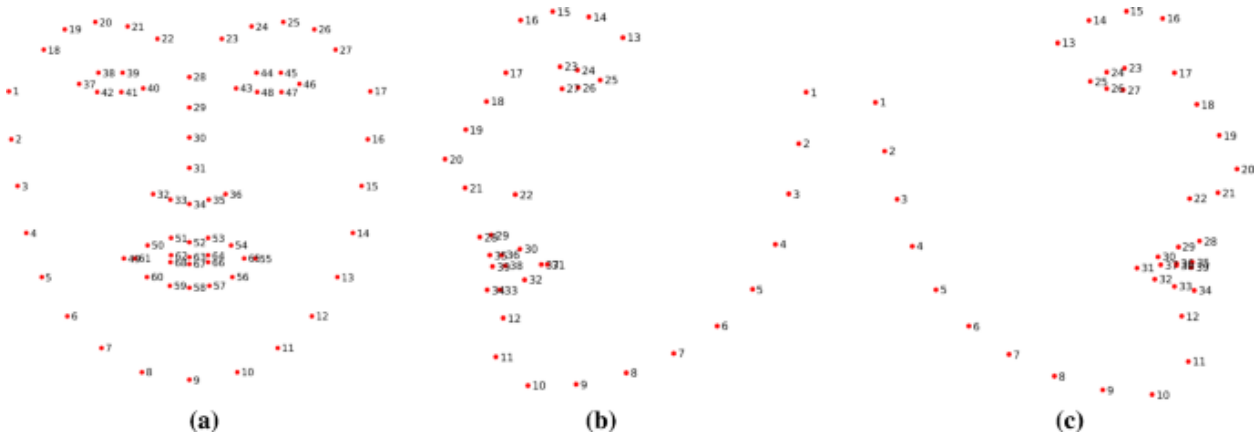
Problem 1:

Implement a script that can detect a face in an image. You can assume that the face will be within 80% of the image dimensions. You can ignore faces that occupy less than 10% area of the ROI (Region of Interest). Example figure shown below.



Problem 2:
Implement a script that can detect multiple faces in an image. Same specifications as problem 1. Mark the face closest to the center of the image as the one under focus. You can mark the person under focus with a green bounding box and other faces with a red bounding box.

Problem 3:
Implement a script that can identify most commonly known facial landmarks in the image. This needs to be done only for the face in focus but you can choose to do it for all faces in the frame. You can come up with your own list of facial landmarks. Elaborate on why you chose those specific landmarks. A sample set of facial landmarks is shown below:



Problem 4:
Implement a script that can identify the head pose either using the landmarks that you have calculated in problem 3 or with any other algorithm. Mark the direction in which the person is looking at with a vector (theta, phi) centered at the tip of the nose of the person.This needs to be done only for the face in focus but you can choose to do it for all faces in the frame.

Problem 5:
Combine all problem solutions listed above into one solution and create a head pose tracker. You can test this with standalone videos. We would ideally like a solution that can be tested in real time with a webcam.

Additional Problems:

- If the head goes out of focus and comes back in, will your algorithm start tracking again?
- How are you prioritizing tracking for objects in focus?
- Are you able to track with foreign objects like headphones, spectacles, etc?
- If there is occlusion (example: if someone is drinking from a bottle) will the tracking stop?
- Anything else that you think will be important for a real-world application but might not have been mentioned above. Explain why it is important.

NOTE:
Please train/test your solutions on your own dataset. You can create one by yourself or choose one/multiple from the web. Please write a report briefly detailing the approach you take for every solution along with the dataset/s. Attach the dataset/s if you create one and/or leave a link/s to it if you use one from the web.