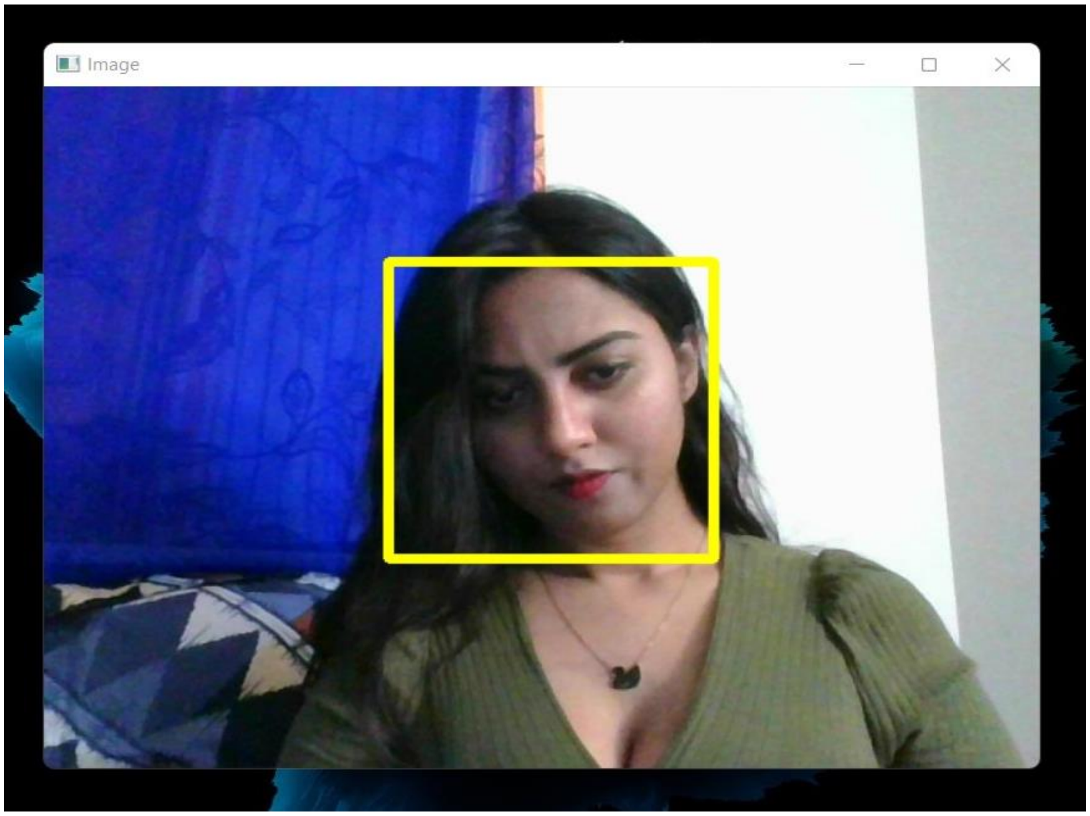
TASK 2: WHAT DID YOU ANALYZE IN THE ABOVE CODE?

DETECTING FACES USING WEBCAM

- Firstly, we install OpenCV.
- Then, we use the Haar cascade classifier to detect the object for which it has been trained.
- Then we configure the screen resolutions and capture the frame using the webcam.
- After capturing the image, we convert the image to RGB, and we try to detect the face in the captured frame using the function detectMutliscale.
- For the faces detected, we draw a rectangle around the face.
- Then, we display the frame around the face.
- We can click the button ‘q’ to break out of the loop.
- Finally, we clean up by releasing the stream and destroying the windows.

Output:



FACE DETECTION USING CNN(DEEP LEARNING MODEL)

- We use CNN to detect faces. Firstly, we load cropped images of a dataset of 17 people.
- Then we use ImageDataGenerator for pre-processing the training images, the hyperparameters generate different images by changing the original images.
- Then we generate the training and testing data, then we print the class labels.
- We create a lookup table for the faces, by giving a numeric tag to each face image, we save the face map for future reference.
- The number of neurons in the output layer is equal to the number of faces.
- Now we create the CNN model using deep learning, we use a sequential classifier.
- The first step is Convolution, then Max Pooling, then flattening, then adding a fully connected Neural Network and compiling the CNN in 10 epochs.
- Finally, we predict the test image as one of the 17 people.

TASK 3: WRITE WHAT COULD BE THE REQUIREMENT, SPECIFICATION, AND ENVIRONMENT FOR THE FACE DETECTION MODEL BY TAKING THE BELOW EXAMPLE.

DETECTING FACES USING WEBCAM

REQ: The model must be able to detect the faces and draw a rectangular frame around the faces.

SPEC: The webcam streams the video accurately, and the haar cascade classifier detects the faces in the webcam and draws a rectangular frame around each face found correctly.

ENV: The webcam provides an accurate video stream; the haar cascade classifier detects the faces accurately; the rectangular frame is accurately drawn around each face that has been detected.

FACE DETECTION USING CNN (DEEP LEARNING MODEL)

REQ: The model must be able to recognize the face from the set of trained faces.

SPEC: The model must recognize the face from the trained faces and predict the output based on the images that it is trained with.

ENV: The model is trained well with the face images; The model recognizes the face in the test image and labels it as one of the trained faces correctly.

TASK 4: WRITE AN ANALYSIS ON WHETHER OUR FACE DETECTION MODEL IS SATISFYING ALL THREE THINGS.

DETECTING FACES USING WEBCAM

Yes, the requirement is feasible, and the specification and environment are correct because the model turns on the webcam and reads the video stream accurately; the classifier detects the face from the webcam and draws a rectangular frame around the face as shown in the output. Hence, the model is satisfying all three things.

FACE DETECTION USING CNN (DEEP LEARNING MODEL)

Yes, the model satisfies all three things in a few cases, but the accuracy rate of the model is quite low, which could lead to wrong predictions as well. The model is trained with the face images; The model recognizes the face in the test image and labels it as Sharukh khan correctly.

TASK 5:

Choose one of the problems such as face detection or vehicle detection. Write what could be the requirement, specifications, and environment for that problem

DETECTING VEHICLE(CARS) USING A WEBCAM

REQ: The model must be able to detect a car and draw a rectangular frame around the car in a video having different vehicles.

SPEC: The video is loaded accurately, and the vehicle detection classifier detects the car in the video and draws a rectangular frame around each car found correctly.

ENV: The input provides an accurate video stream; the vehicle detection classifier detects the cars accurately; the rectangular frame is accurately drawn around each car that has been detected.

VEHICLE DETECTION USING CNN (DEEP LEARNING MODEL)

REQ: The model must be able to recognize the vehicle from the trained set of vehicles.

SPEC: The model must recognize the vehicle from the trained faces and predict the output based on the images that it is trained with.

ENV: The model is trained well with the vehicle images; The model recognizes the vehicle in the test image and labels it as one of the trained vehicles correctly.

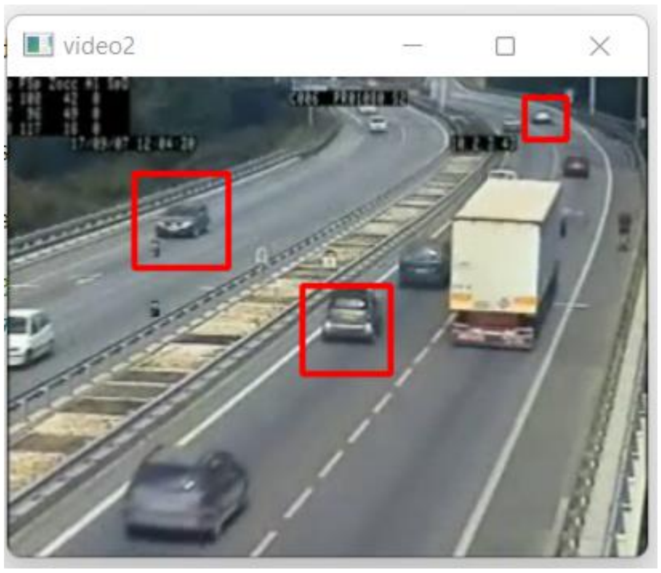
Write an analysis on whether the written requirement is feasible or not, environment and specification are correct or not, etc.

DETECTING VEHICLES USING WEBCAM

- Firstly, we install OpenCV.
- Then, we use the vehicle_detection classifier to detect the cars.
- Then we take a video as an input that has cars.
- We convert the video frames to RGB, and we try to detect the Vehicle(car) in the captured frame using the function detectMutliscale.
- For the Vehicles(car) detected, we draw a rectangle frame around the Vehicle.
- We can click the button 'q' to break out of the loop.
- Finally, we clean up by releasing the stream and destroying the windows.

Yes, the written requirement is feasible, environment and specifications are correct. The model correctly detects the cars in the video having different vehicles and also draws a frame around the detected cars.

Output:



VEHICLE DETECTION USING CNN (DEEP LEARNING MODEL)

- We use CNN to detect Vehicles. Firstly, we load a dataset of 16 different vehicles.
- Then we use ImageDataGenerator for pre-processing the training images, the hyperparameters generate different images by changing the original images.

- Then we generate the training and testing data, then we print the class labels.
- We create a lookup table for the Vehicles, by giving a numeric tag to each Vehicle image, we save the Vehicle map for future reference.
- The number of neurons in the output layer is equal to the number of Vehicles.
- Now we create the CNN model using deep learning, we use a sequential classifier.
- The first step is Convolution, then Max Pooling, then flattening, then adding a fully connected Neural Network and compiling the CNN in 10 epochs.
- Finally, we predict the test image as one of the trained vehicles.

Yes, the written requirement is feasible, environment and specifications are correct, but the accuracy of the model is very low, which could be increased by improving the training data or changing the hyperparameters or by increasing the number of epochs.