**Task\_02:**

**Object Detection and Classification:**

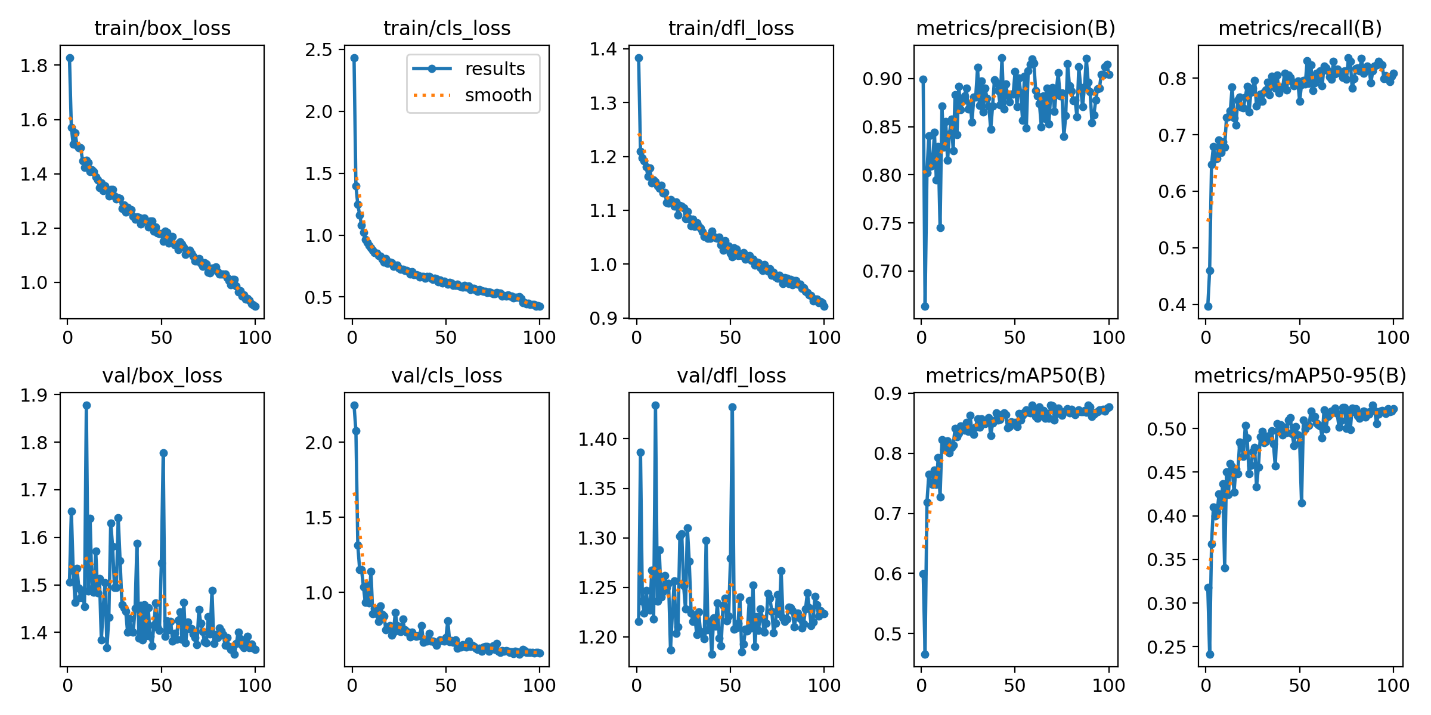
As this task comprises of two components, for the object detection I’ve used the pre-trained YOLO model “yolov8n.pt” and re-trained again on the dataset provided.

The dataset was about the masked and un-masked facial dataset. So, for the object detection it was to identify the whether the person in the image is masked or un-masked.

For this, as I’ve used the yolov8n.pt pre-trained model, for retraining I’ve hyper-parametered with 100 epochs.

After that, I’ve tested the current model with the testing dataset.

The result was as following:



The precision and recall were increasing as the number of epochs were increasing.

The val\_loss was decreasing as the number of epochs were increasing.

Image Classification:

The second thing is classification where the input should come from the previously trained object detection model and image should be cropped with respect to human. So, if there are any other objects like cars, bikes, tables we need to crop only the person image and send it as the input for the image classification model.

So, here what I’ve done is, I’ve given input image for the prediction to the object detection model and the result gets stored in the type of ultralytics.engine.result.Results class.

So, here I’ve created a loop to iterate along all the input images for the testing.

At each result/ image, after detecting the object in that one, I’ve taken the count of the person class detection while ignoring the other classes.

Later based on the count of person detections, I’ve created a looping statement and then took the co-ordinates of each detection.

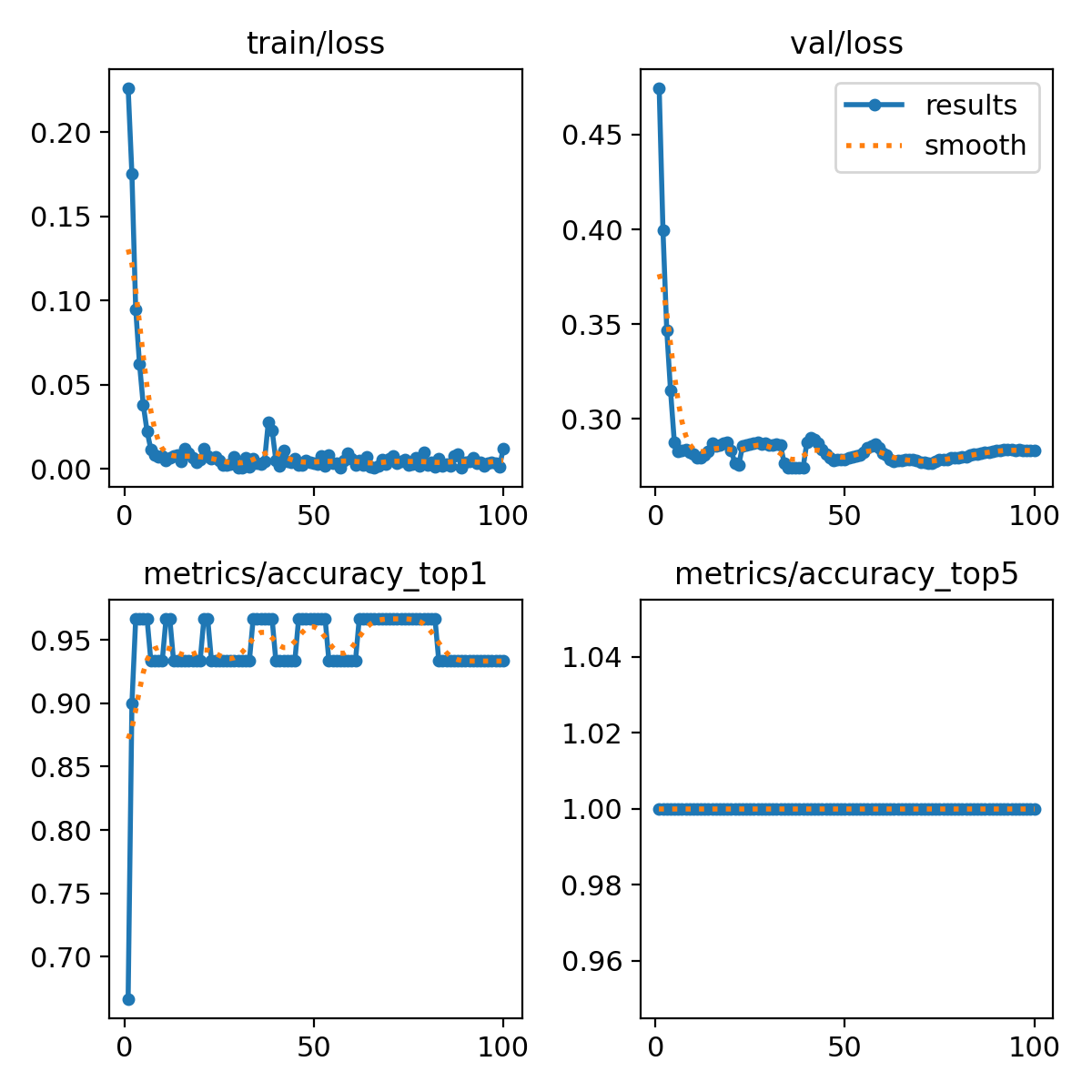
By using these co-ordinates and the original input image, I’ve cropped the image of the person’s from the image that was sent for the prediction.

Before making predictions for object detection, I’ve also took another pre-trained classification “yolov8s-cls.pt” model. And again I’ve trained it with the dataset for [‘happy’, ’sad’, ’neutral’] categories.

So, before going for the predictions of object detection, we have a model for classification as well. As soon as we predict the object detections in the image, we check for the count of detections and take into consideration of person-detections.

Using the co-ordinates of those detections, we crop the image and send it to the classification model for classifying the categories like happy, sad, neutral.

The results for the classification are like following:



As the number of epochs increases the training and validation loss also lowering down.

Here, due to computational limitations I’ve used only small amount of data for the training and validation. Although, I used some Data Augmentation Techniques for the increase in the data still there is lot more dataset required for more particular usage of problems. The model would perform better if the dataset and the platform is well defined.

The major challenged I faced while making this was the computational resources, as the programs were requiring high-end infra resource.