Nighttime Pedestrian Detection Based on a Fusion of Visual Information and Millimeter-Wave Radar

Now-a-days world is progressing towards self-driving vehicles where intelligent sensors only has to take all decision to avoid all possible obstacles occur while driving such as Pedestrians, pothole etc. Sensors were dependent on similarity distance to identify objects but this technique prove to be insufficient. So author of this paper employing deep learning algorithm which are based on images and prove its accuracy in almost all domains like medical image disease detection and classification, road side traffic detection and many more.

So sensors alone are not sufficient for accurate detection on pedestrians at night or day time. To improve detection author of this paper employing following techniques

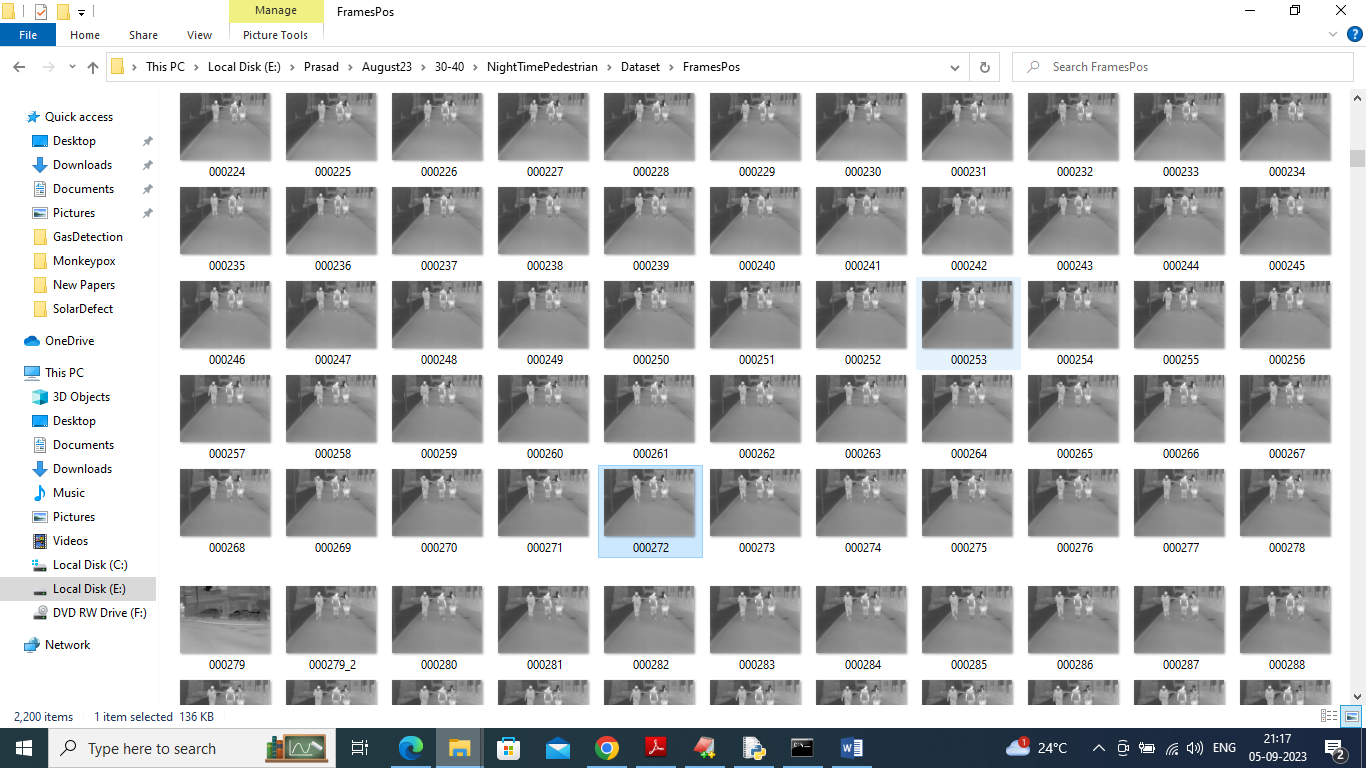
1. Cameras will be installed which are suitable for infrared vision information and millimetre wave (MMW) radar data.
2. Features of images will be extracted and categorized by improved YoloV5 model. YoloV5 model is improvised by adding Squeeze layer called Attention
3. Extended Kalman Filter will be employed to accurately detect or localized pedestrians and this localized image features will be input to improved YoloV5 for detection.

So by employing above features improved YoloV5 detecting pedestrians with high probability and accuracy. Author has compare performance of propose improved YoloV5 with Faster RCNN and existing Yolov5 and many other algorithms. Training all those existing algorithms may take more time so as existing we have used faster RCNN.

To train above existing propose algorithms author has used CVC-09 infrared pedestrians images dataset and this dataset can be downloaded from below link

<http://adas.cvc.uab.es/elektra/enigma-portfolio/item-1/>

This dataset has images for both night and daytime but author has used only for Night time so we have used same images only. Below are the some images from the dataset

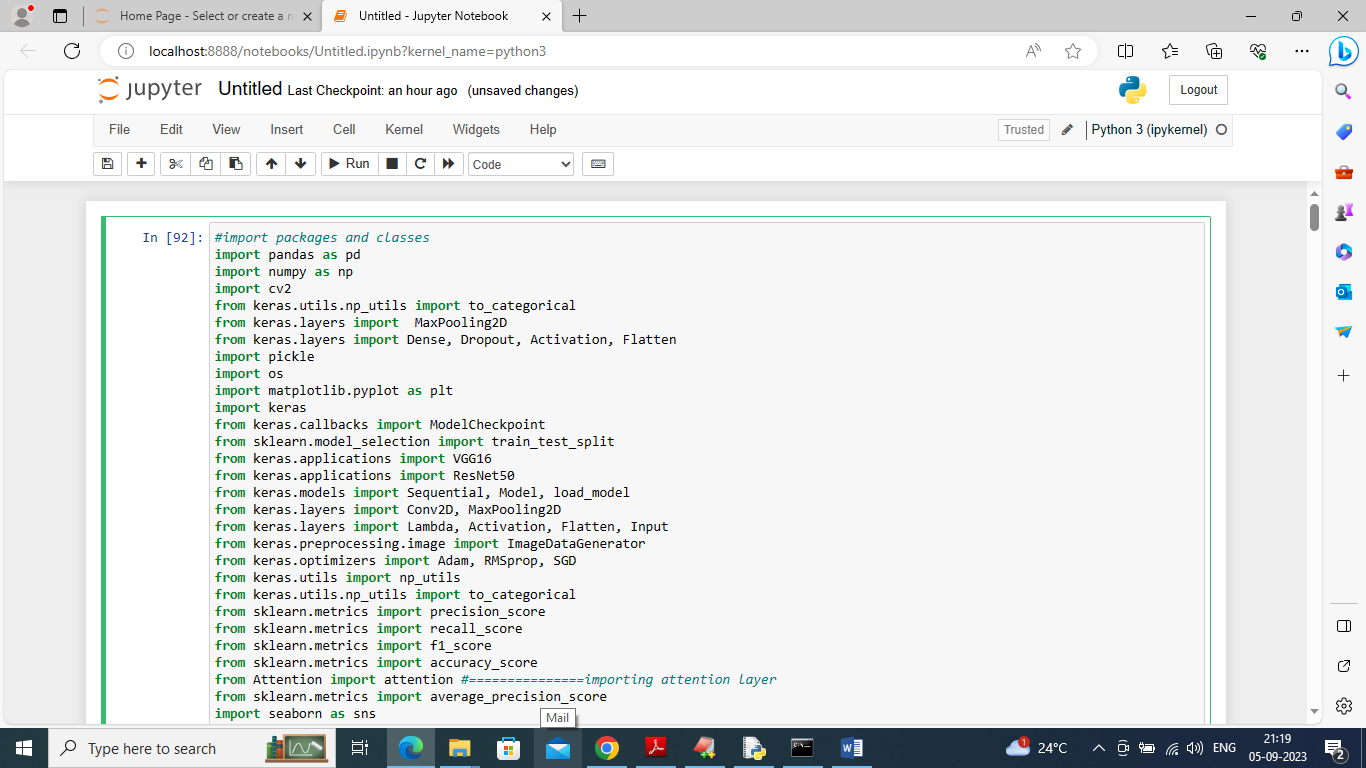


So by using above dataset we will train and test all algorithm performance and each algorithm is evaluated in terms of accuracy, precision, recall and FSCORE

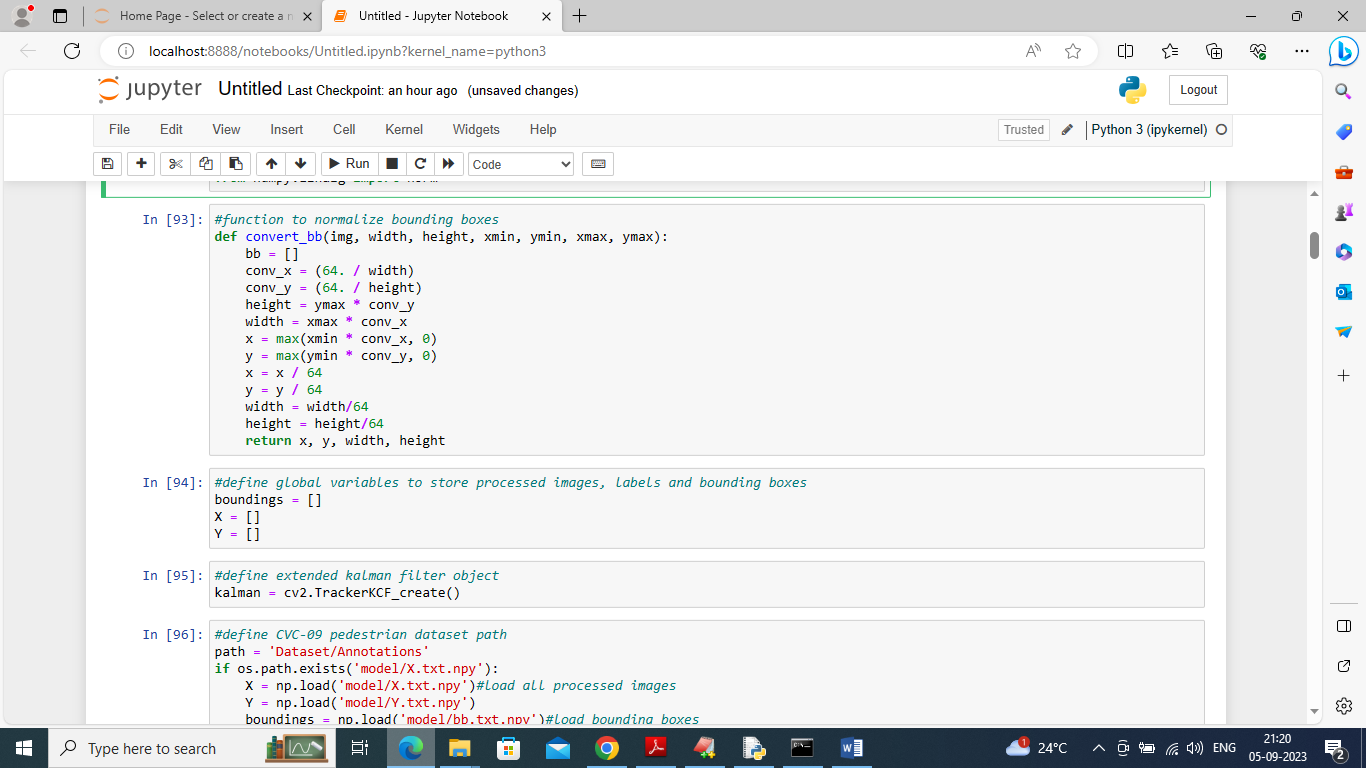
In propose work author has modified YoloV5 but not experimented with advance YOLO family other version as like Yolo6, 7 or 8. So as extension we have trained same dataset with Yolov6 and it’s giving better prediction accuracy compare to Optimized Yolov5.

SCREEN SHOTS

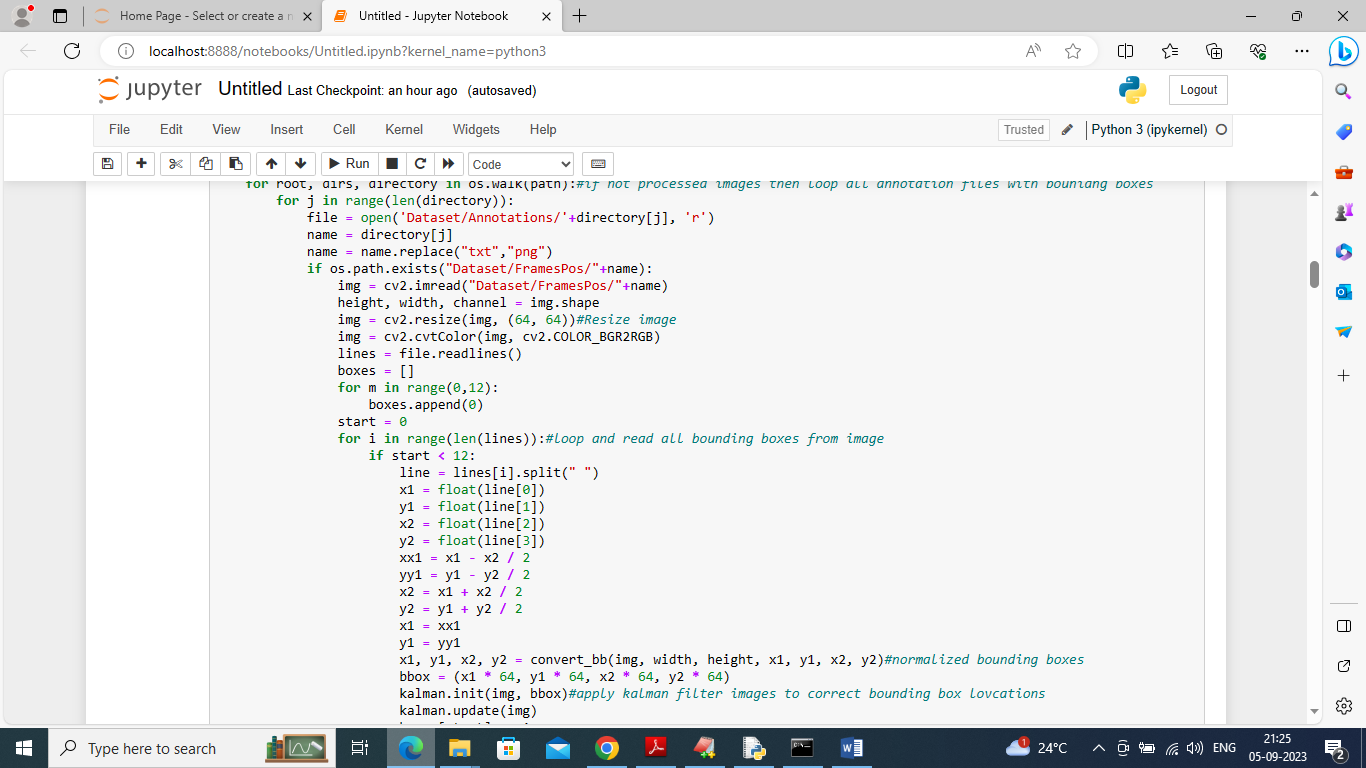
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



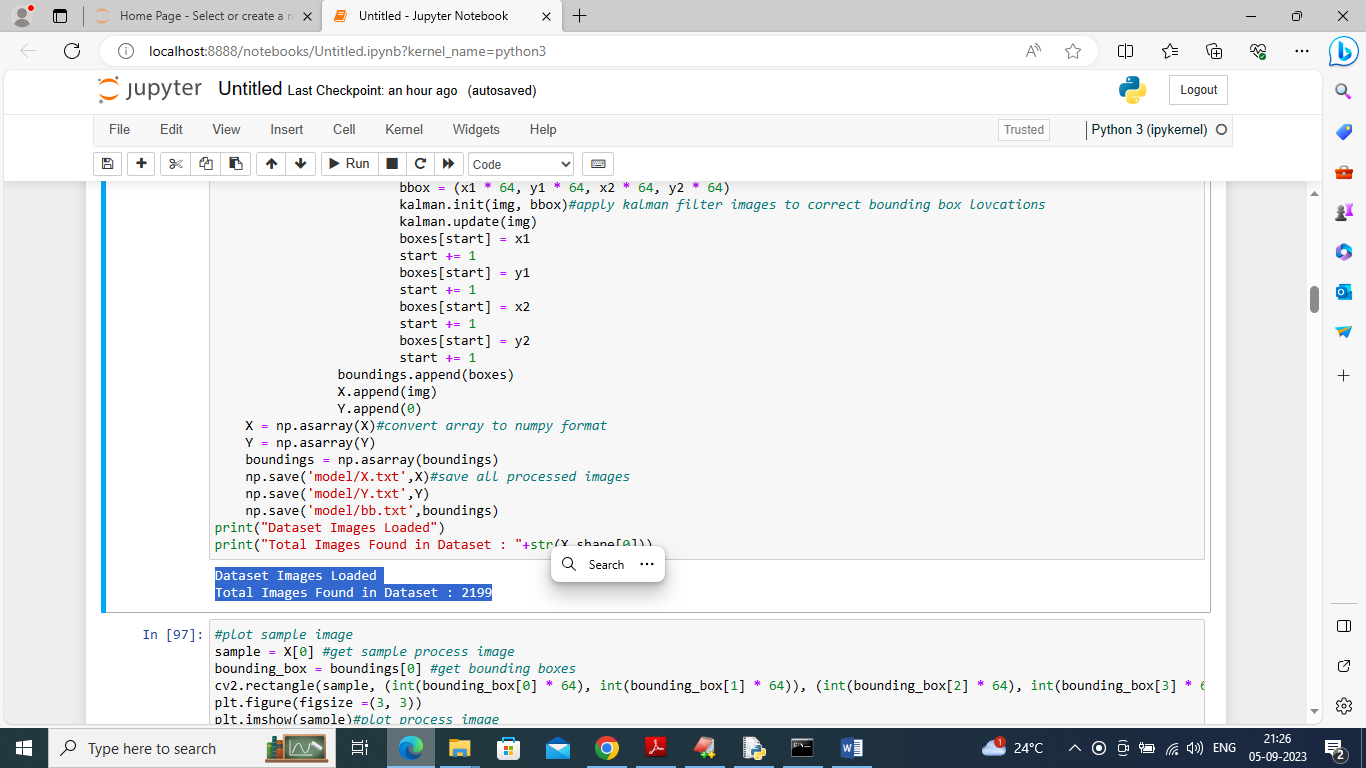
In above screen importing all classes and packages which will help in defining and improving YOLOV5 model and this model can be improved by adding squeeze attention layer



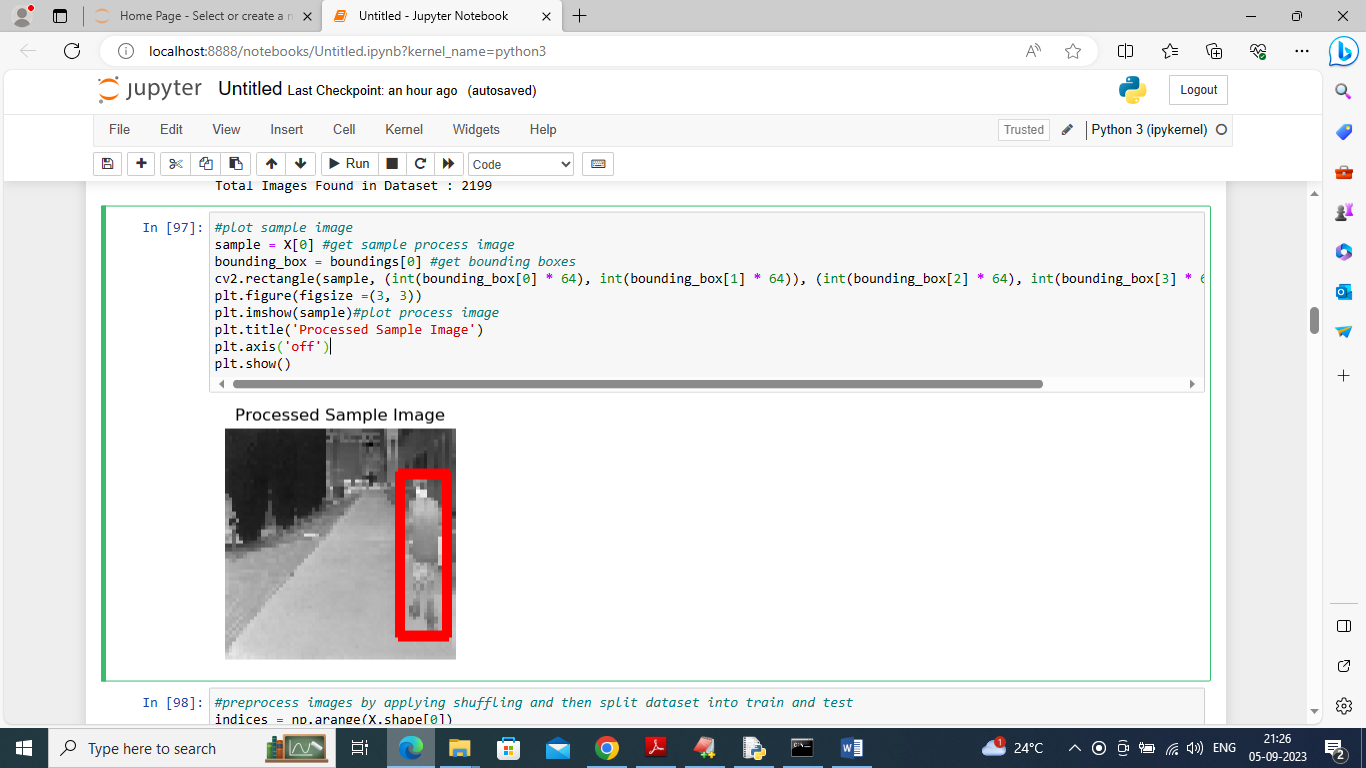
In above screen defining function to normalized bounding boxes which is used to detect and plot box around predicted pedestrians and then defining KALMAN object and global variables



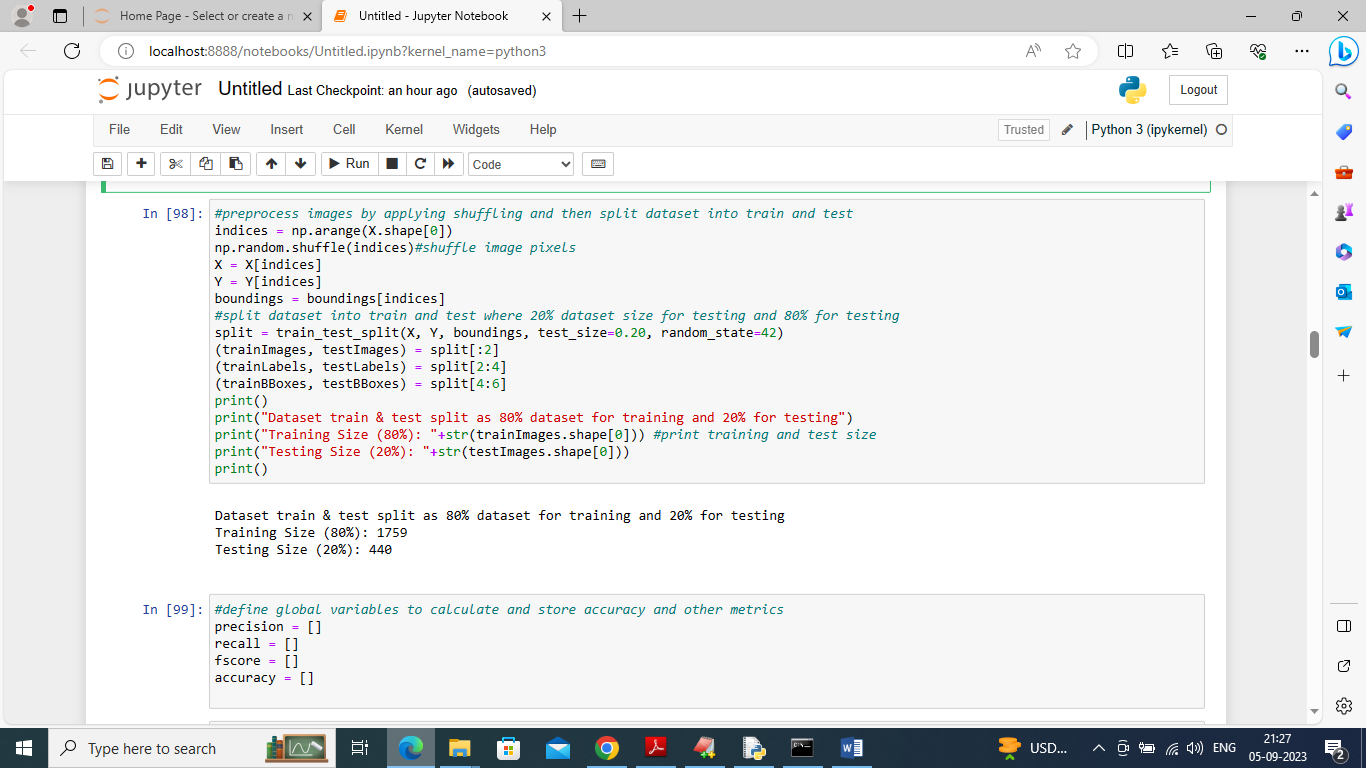
In above screen reading all images and bounding boxes from dataset and then applying KALMAN filter to detect pedestrians and then adding all images and bounding boxes to array variables to generate training data



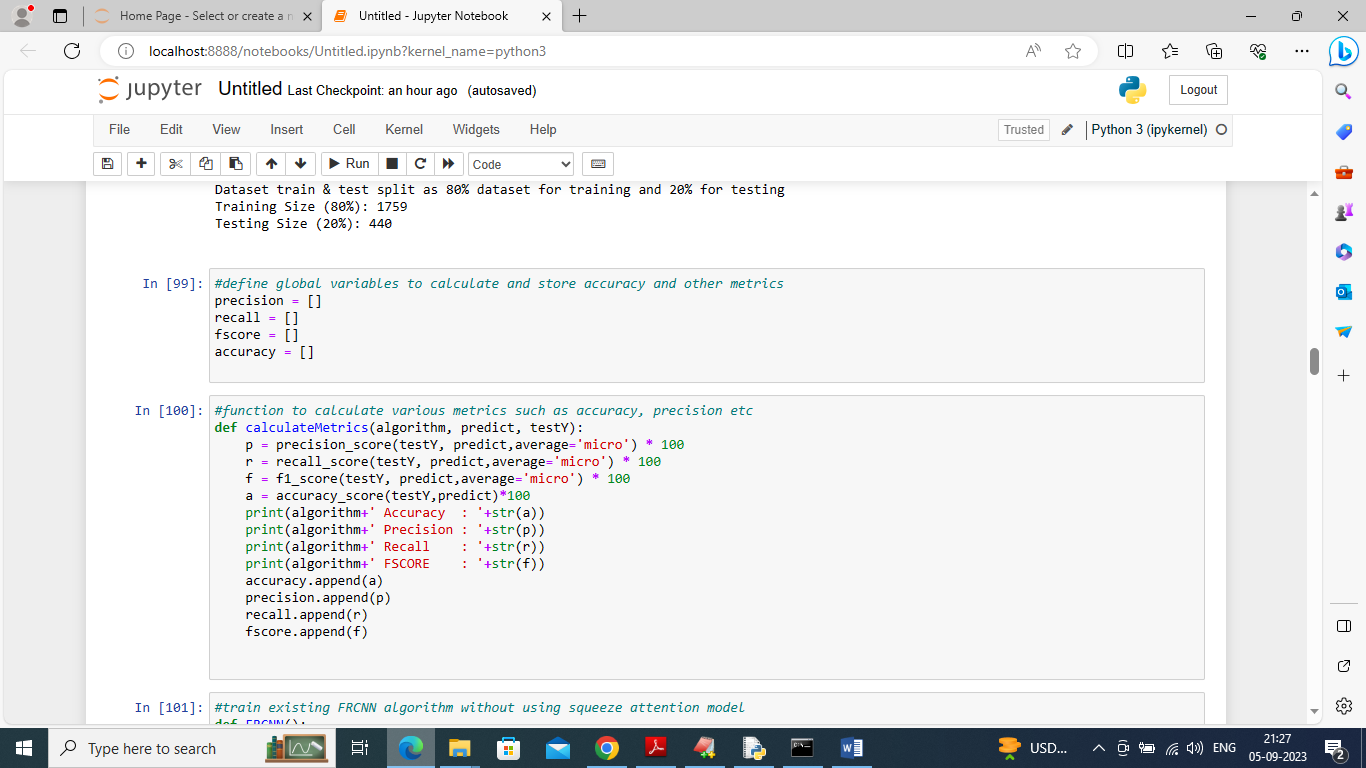
In above screen displaying number of loaded and process images in blue colour text



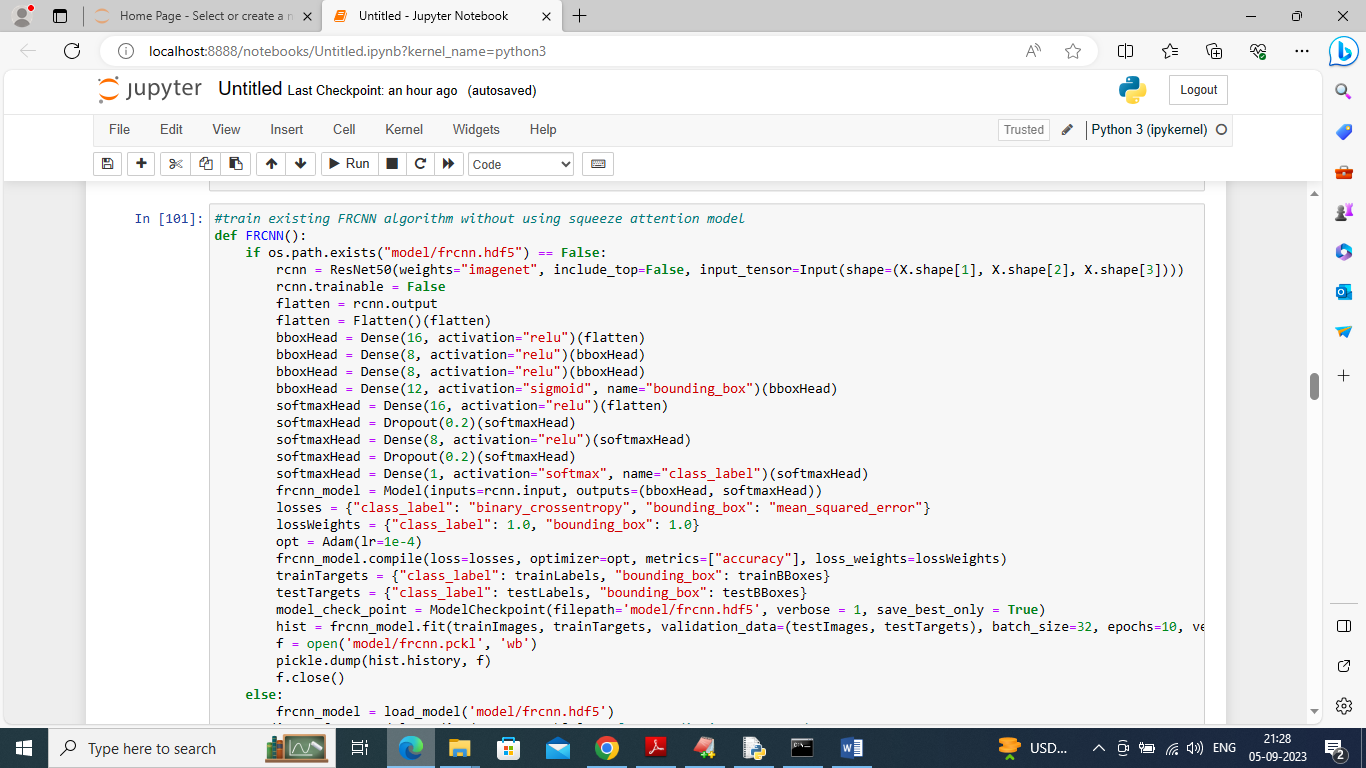
In above screen displaying sample processed image from dataset with bounding box across pedestrians



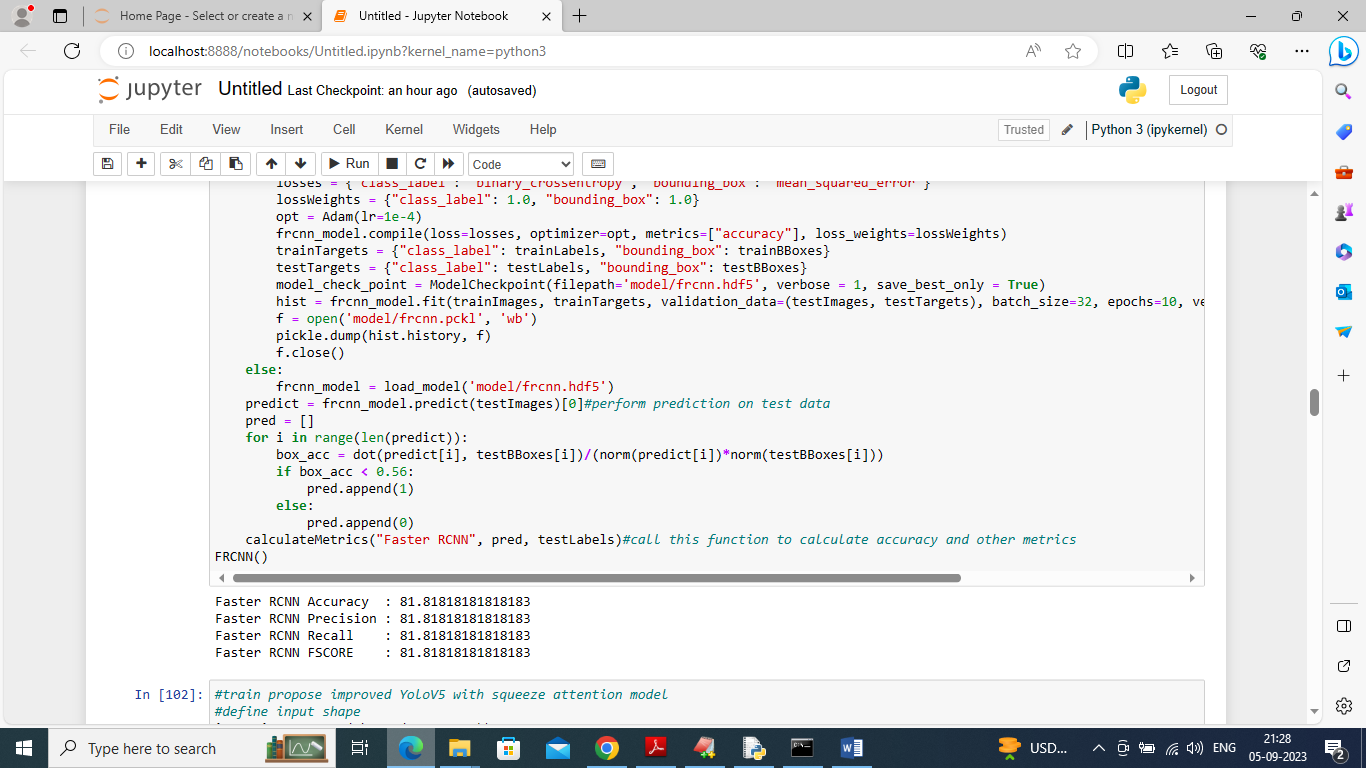
In above screen pre-processing dataset by shuffling all images and then splitting dataset into train and test



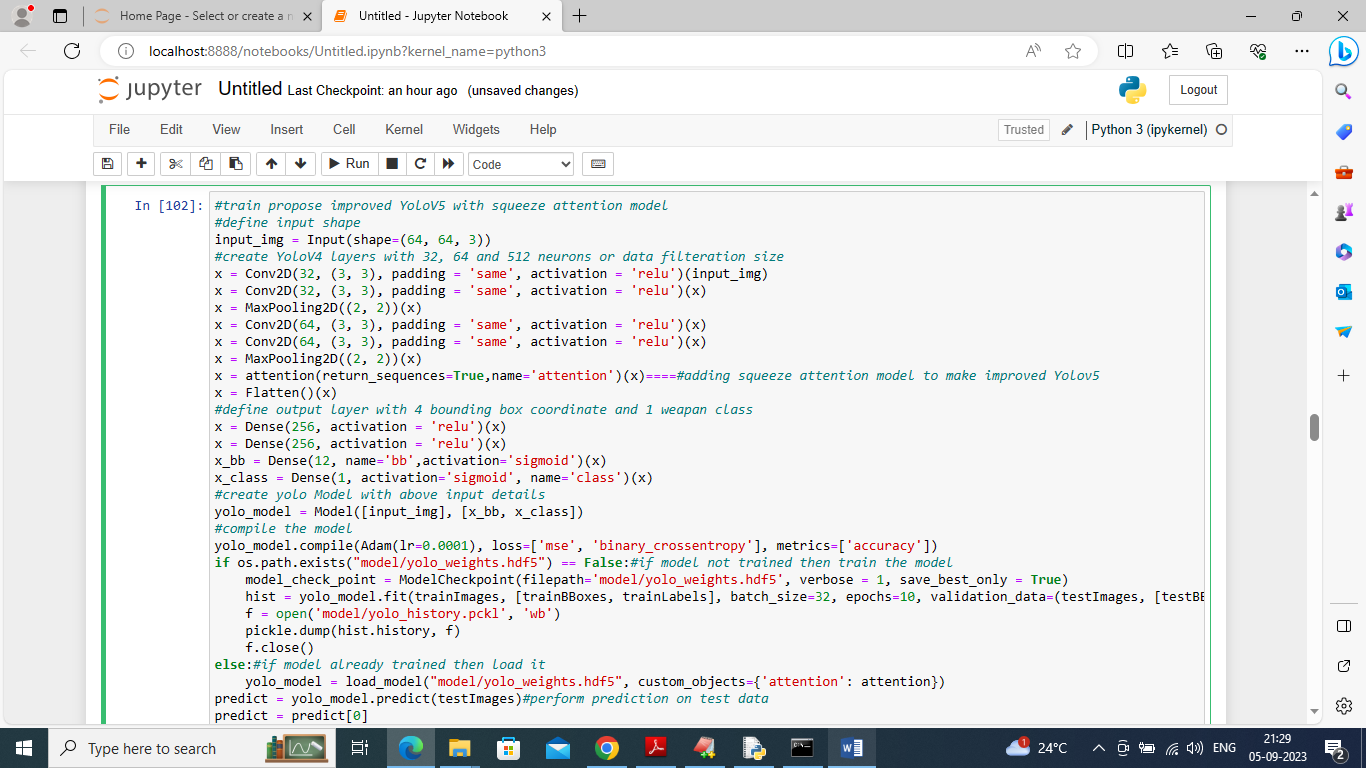
In above screen defining function to calculate accuracy and other metrics



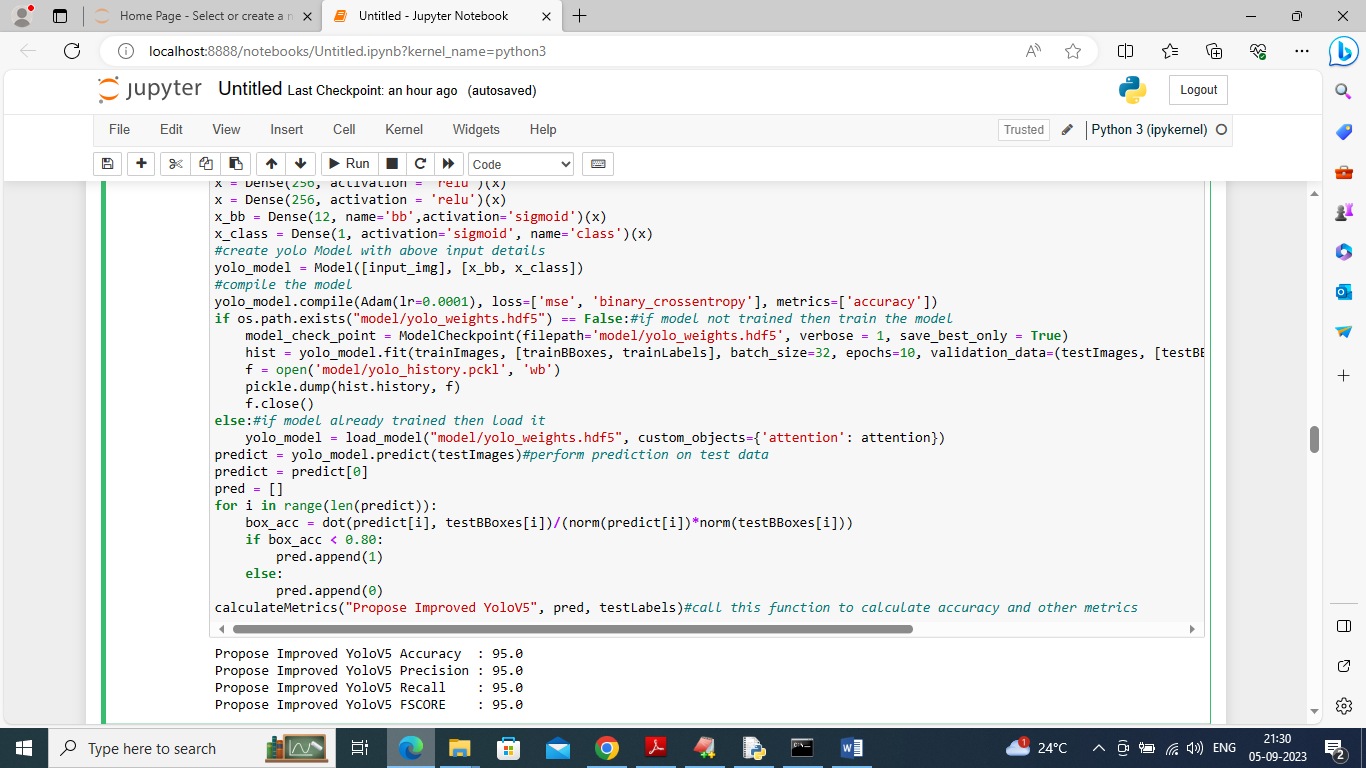
In above screen training existing FRCNN algorithm and after excuting this block will get below output



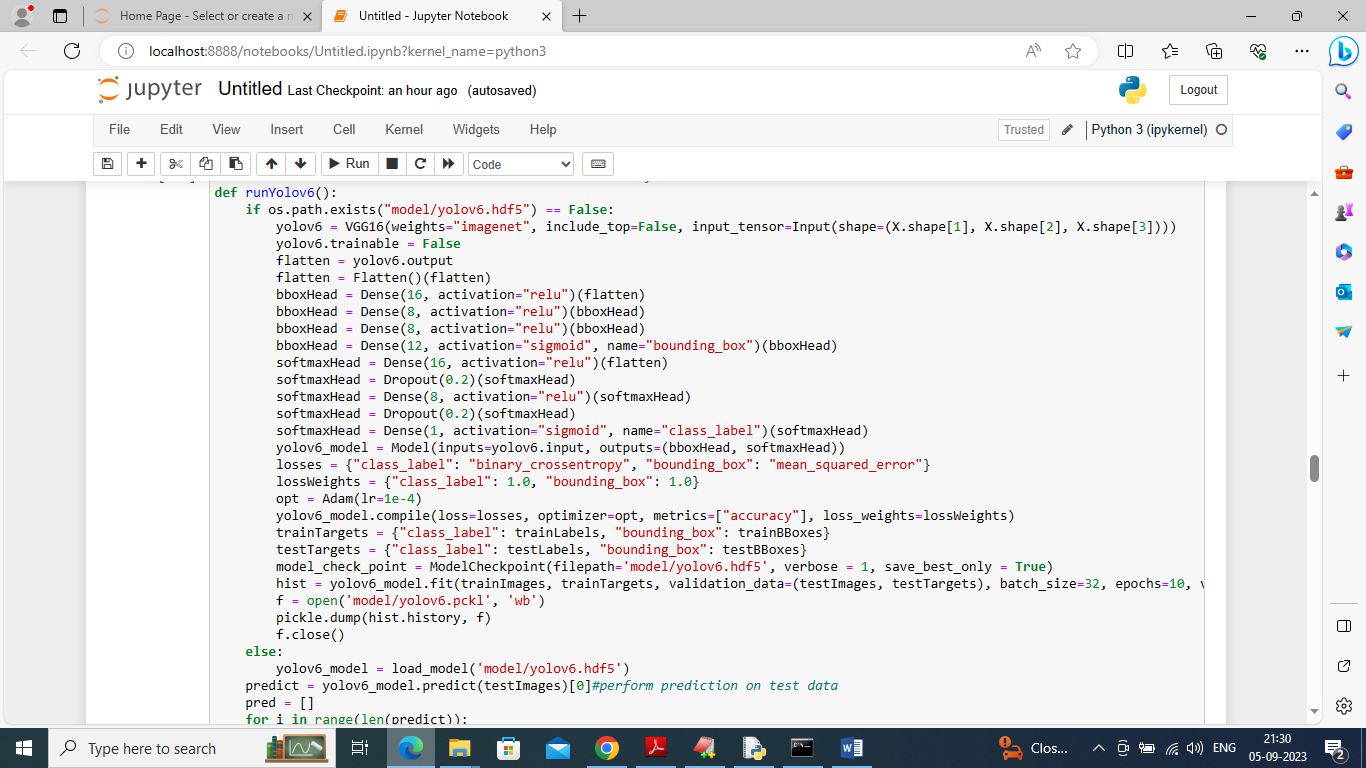
In above screen existing Faster RCNN got 91% accuracy



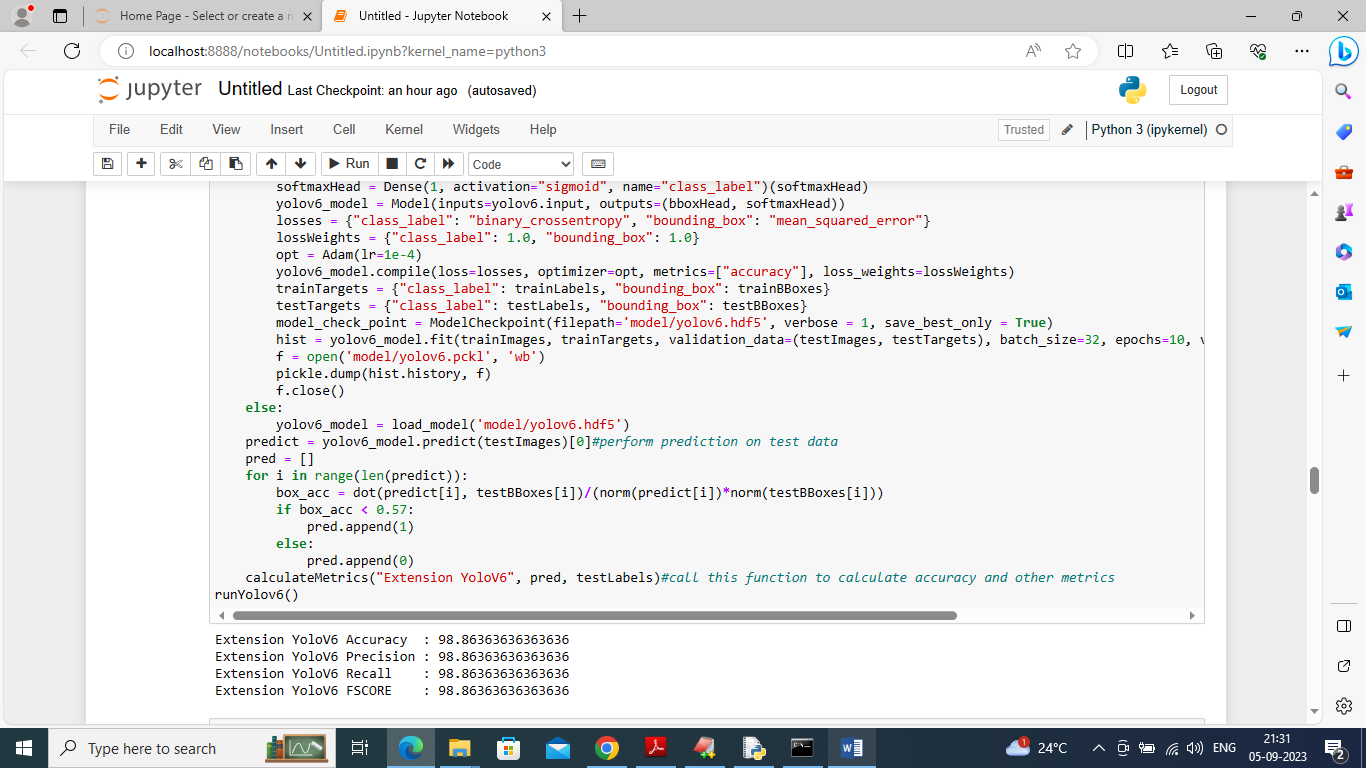
In above screen defining propose YoloV5 model and then improving this model by adding squeezing layer and after executing above block will get below output



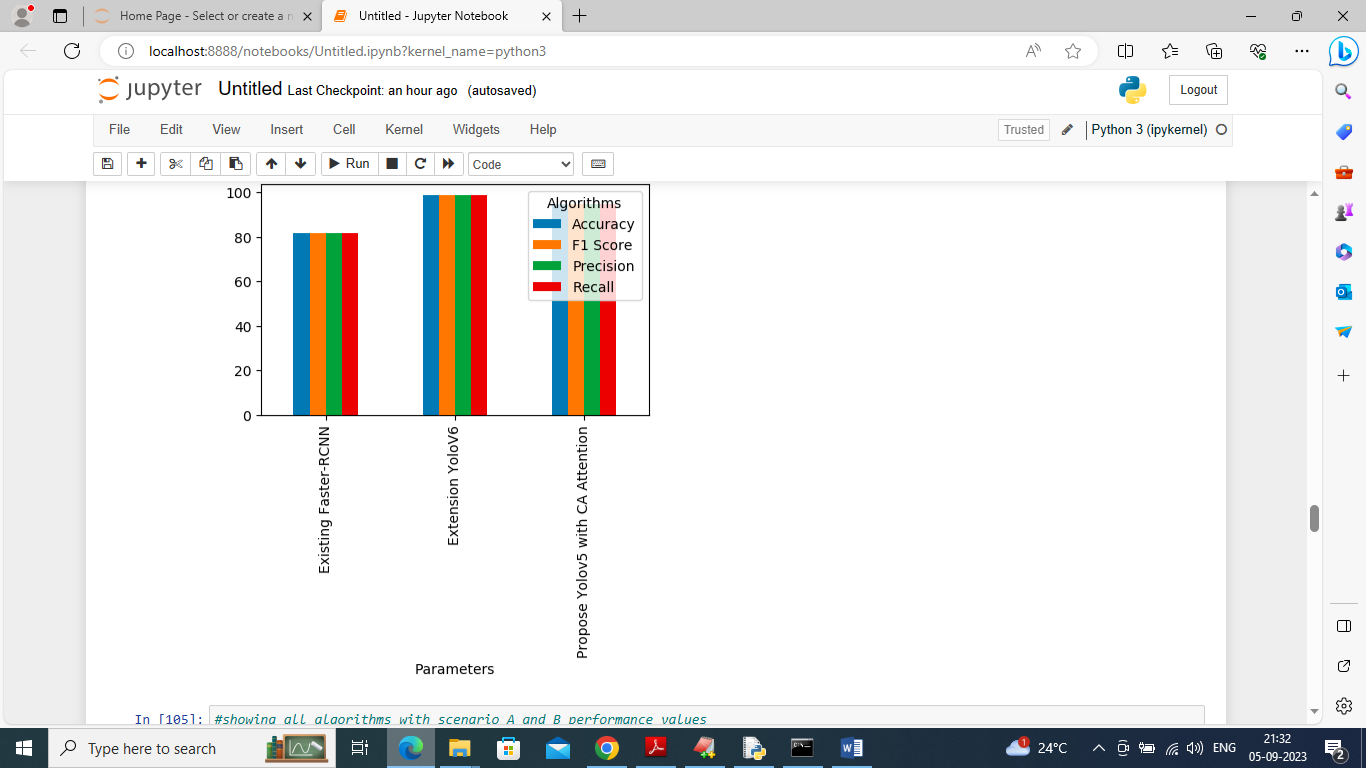
In above screen propose improved Yolov5 got 95% accuracy



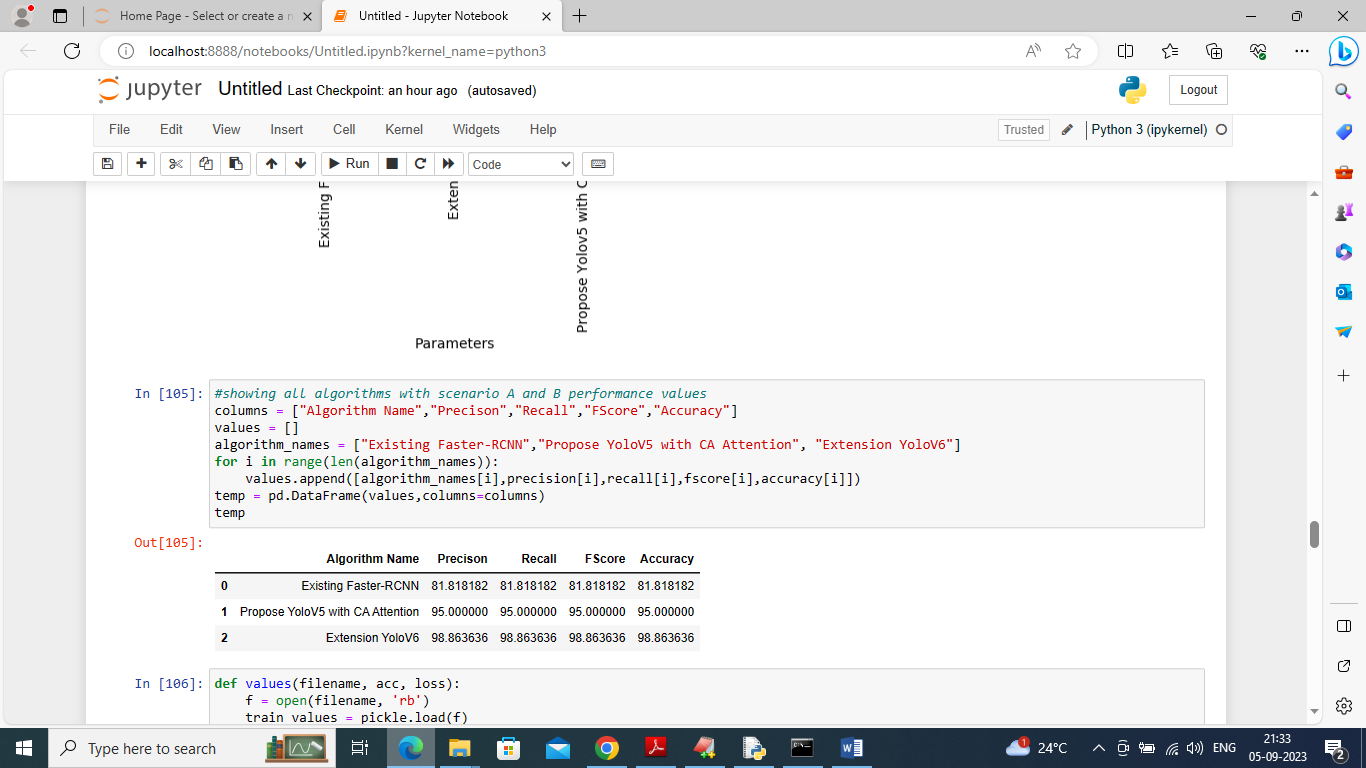
In above screen defining and training extension YoloV6 model and after executing above block will get below output



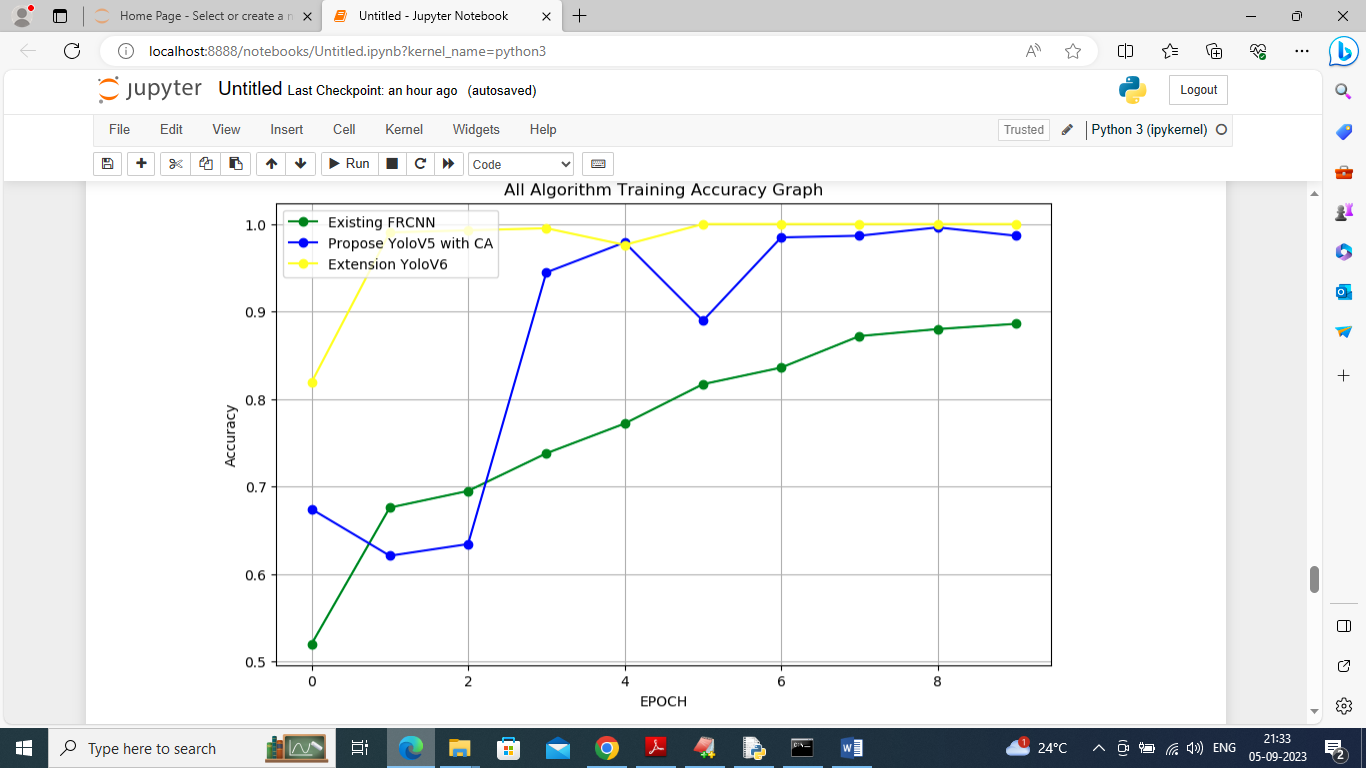
In above screen extension YoloV6 got 98% accuracy



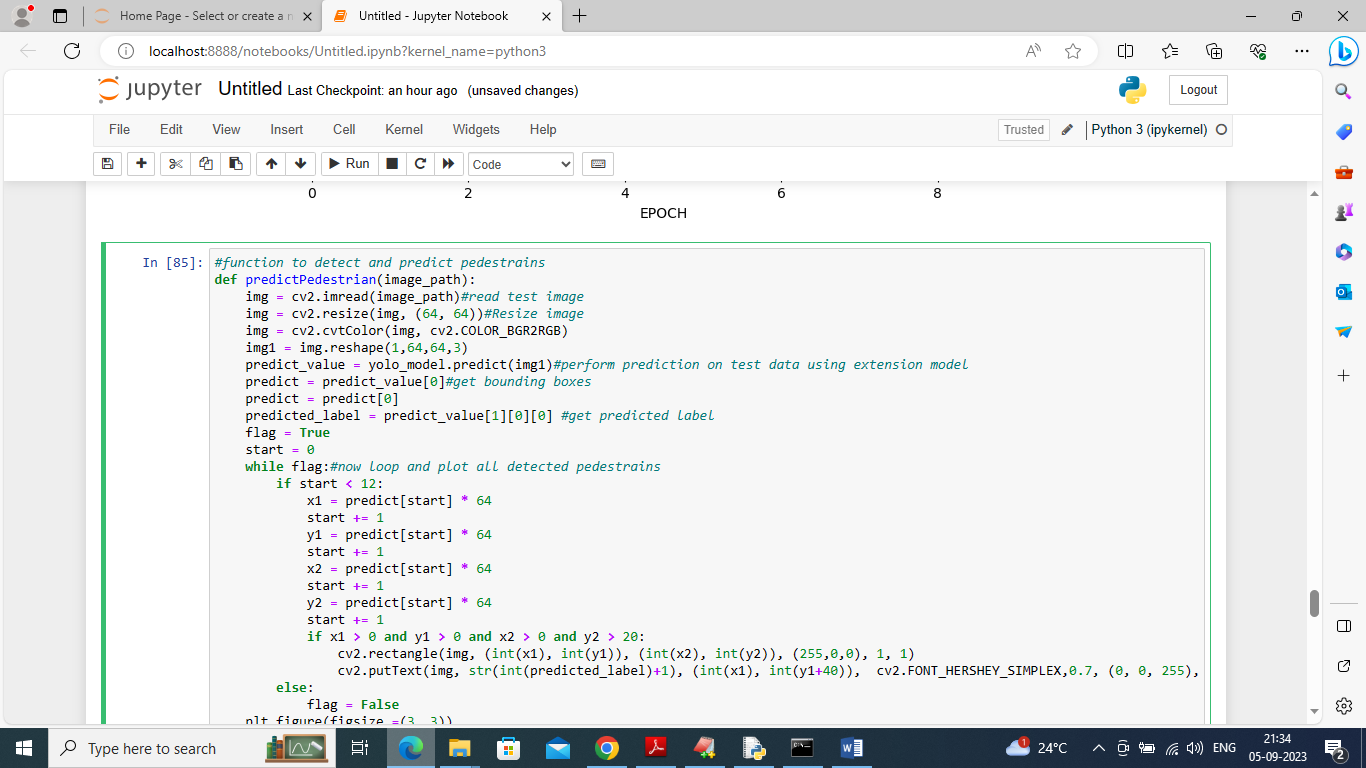
In above screen showing all algorithms performance in graph format where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars



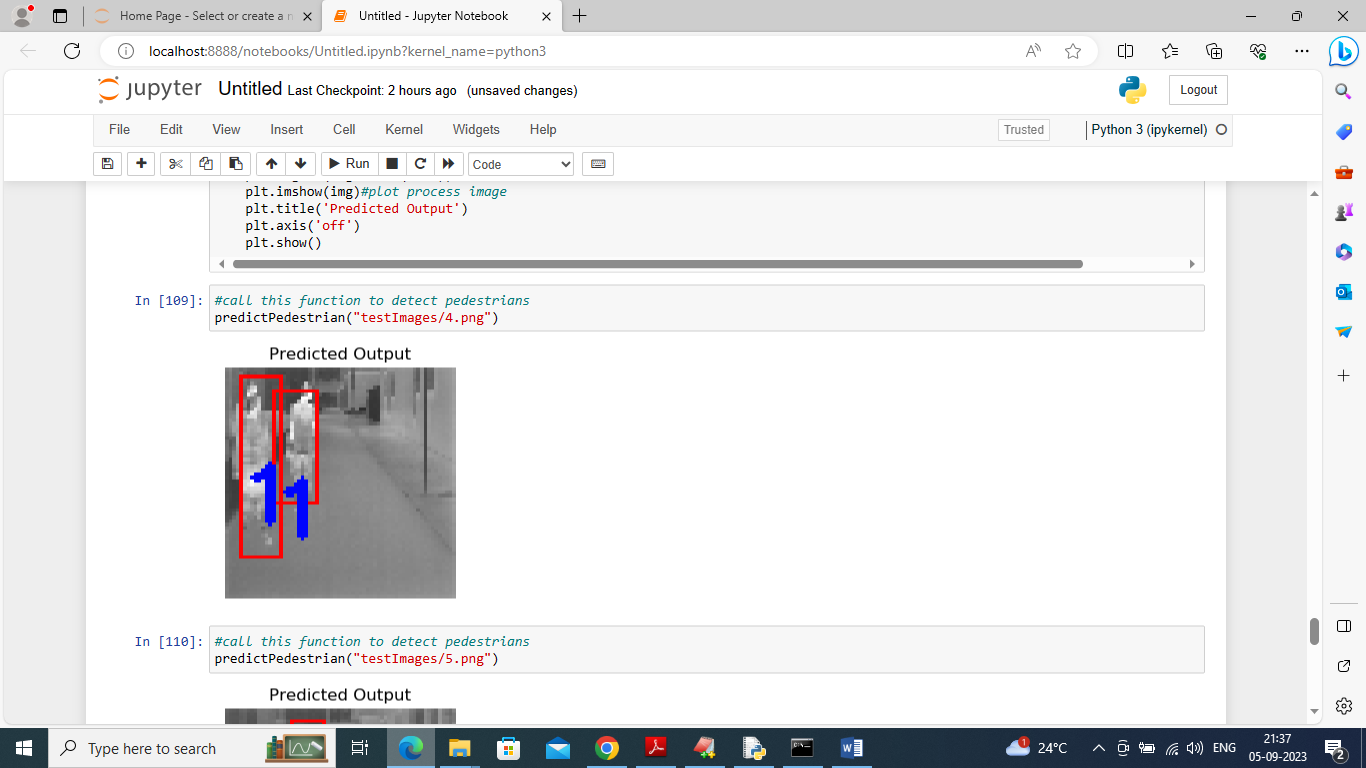
In above screen displaying all algorithm performance in tabular format



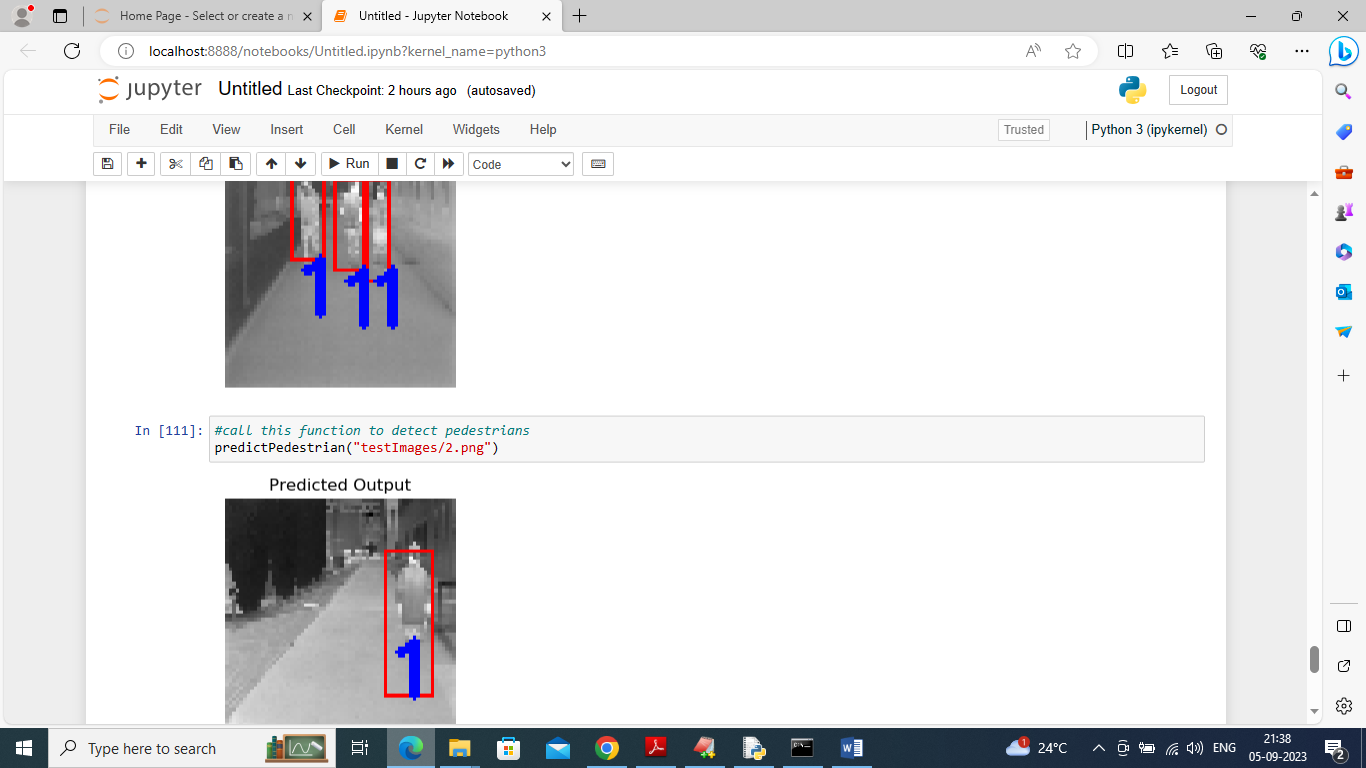
In above graph x-axis represents training epoch and y-axis represents accuracy and each line represents different algorithm. Green line for existing Faster RCNN, blue line for propose YoloV5 and yellow line for extension YoloV6



In above screen defining function to detect and predict pedestrians and this function take test image path as input



In above screen application detected pedestrians and then put red colour bounding box with 1 in blue colour as predicted probability. Above images are of type infrared



In above screen displaying output of other test images

