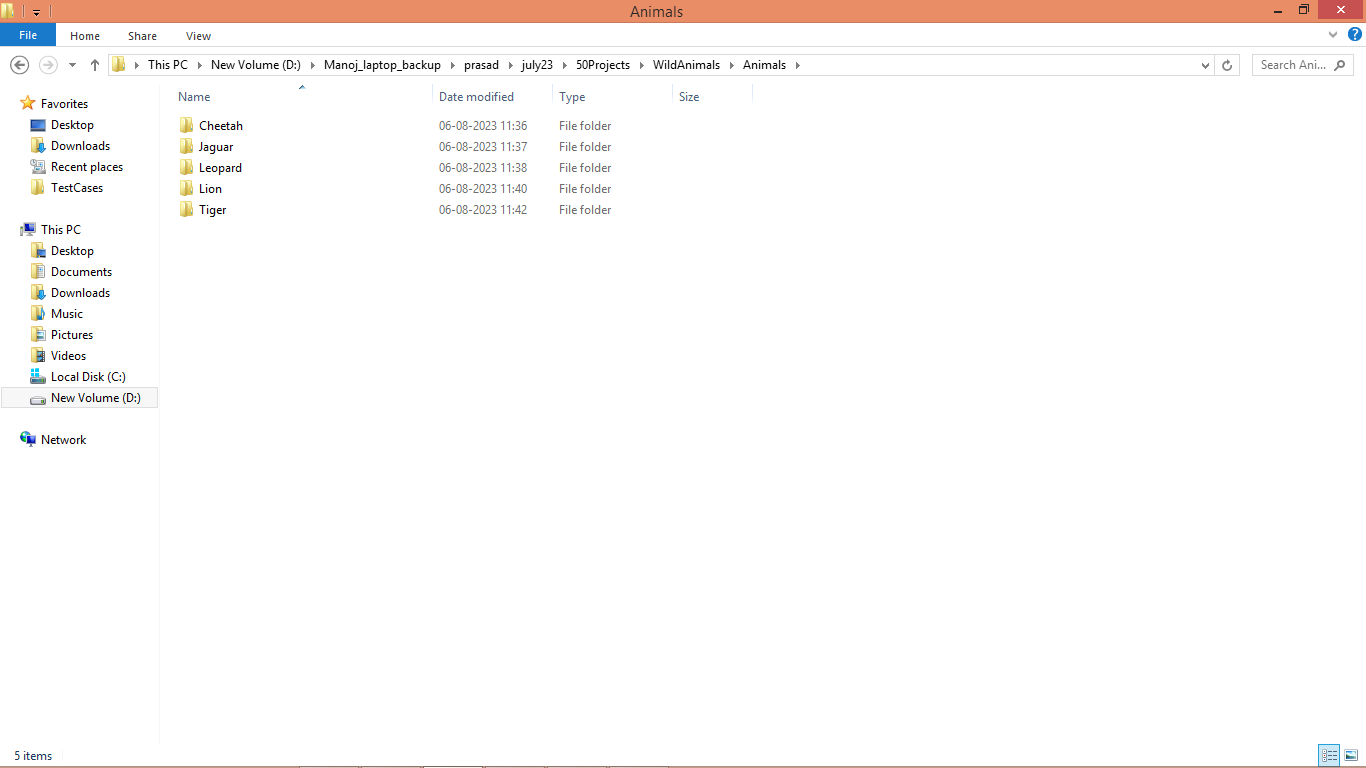
Creating Alert Messages Based on Wild Animal Activity Detection Using Hybrid Deep Neural Networks

Peoples who stayed at forest side will always have fear of animal attacks so to know animal movement always cameras surveillance and deep learning monitoring is applied but this existing technique is very expensive in computing and its detection accuracy is not good enough and to overcome from this problem author of this paper employing combination of VGG19 (Hybrid Visual Geometry Group) + BI-LSTM (Bidirectional Long Short-Term Memory) algorithm to detect animals and its activities.

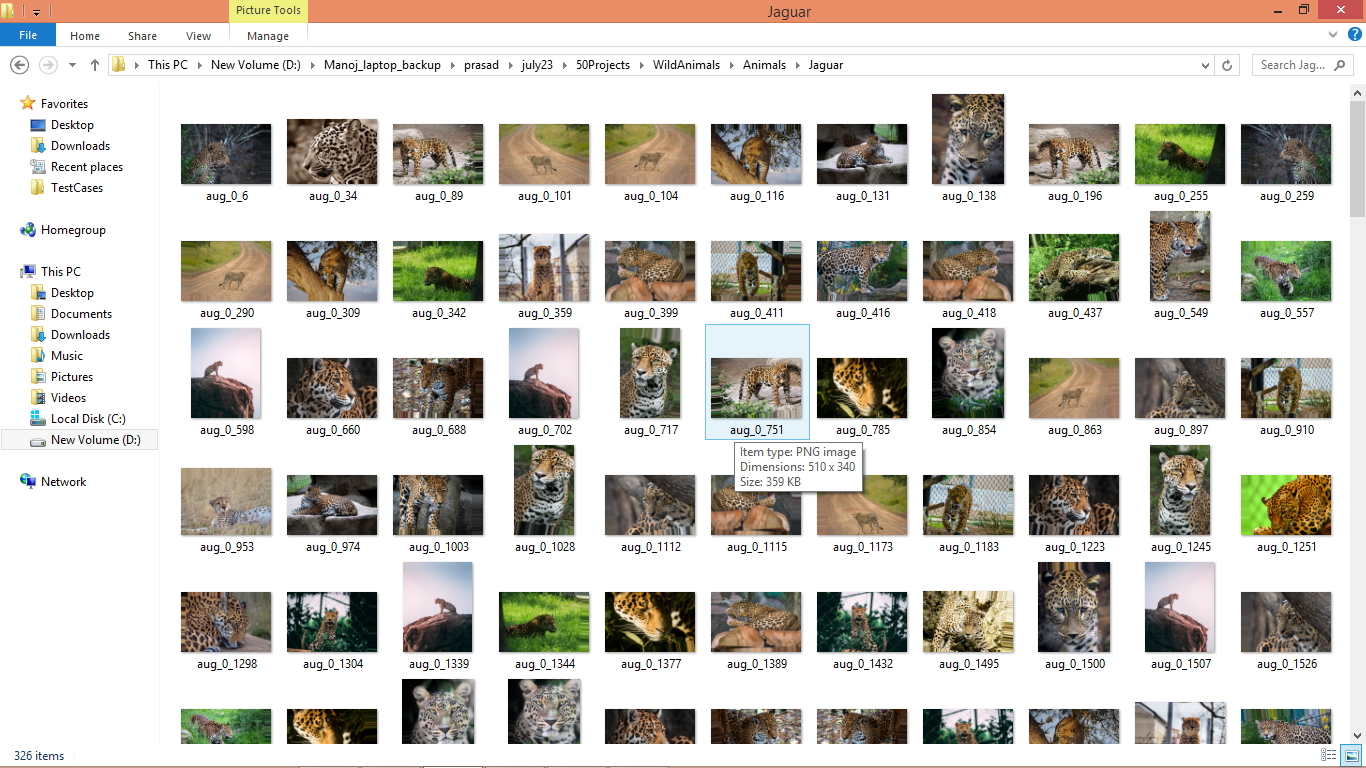
Propose work consists of 5 phases where in first phase dataset images will get process to extract features and then object detection model will be applied in second phase to detect animal and then in 3rd phase VGG19 and BI-LSTM algorithm will be applied to classify animal and in 4th phase SMS alert will be generated with location to forest officer and in 5th phase forest officers will take action to save animal or human lives.

In implementation as academic project 4th phase with SMS cannot be implemented as its not free of cost and 5th phase also not possible so we have implemented first 3 phases by using existing CNN algorithm and with propose VGG19 + BI-LSTM.

To train all algorithms author has used 4 different datasets but all those dataset not available on internet so we have used ‘Wild Animal Dataset’ from KAGGLE website. In below screen we are showing dataset details



In above screen we can see folders with different animal names and just go inside any folder to view images like below screen



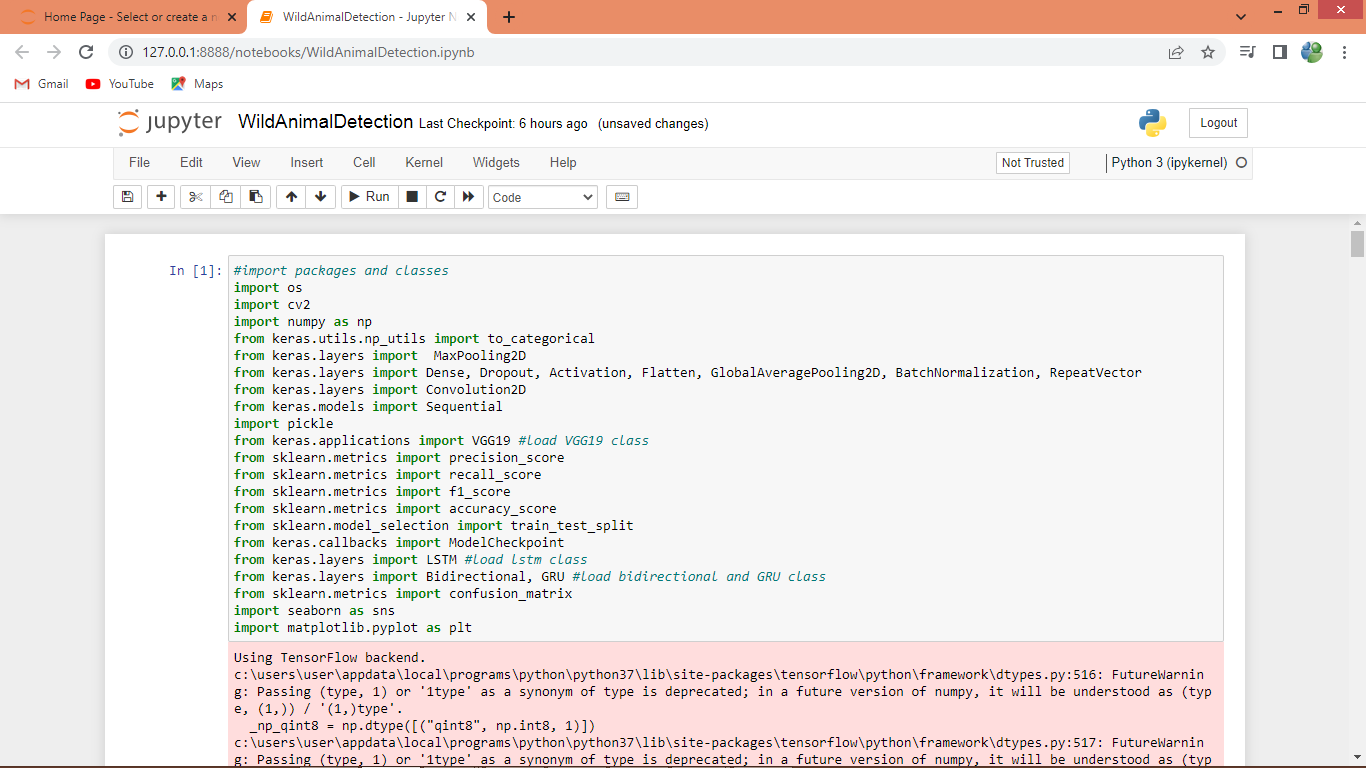
So by using above dataset images we will train and test all algorithms performance and author has tested with many algorithms and it’s not possible to train all algorithms so we are experimenting with existing CNN and propose VGG19 + BI-LSTM.

Extension Concept

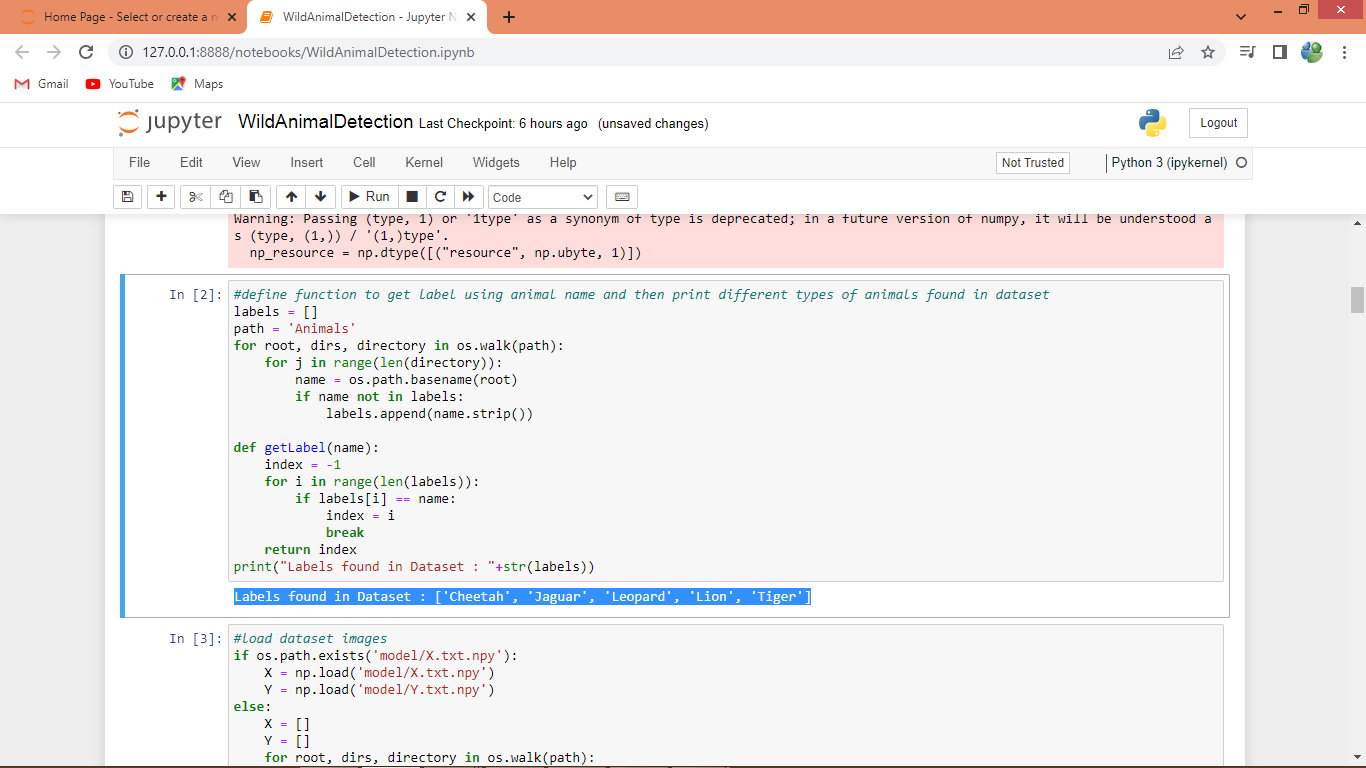
As extension we have combined Bidirectional GRU (Gated Recurrent unit) layer to CNN algorithm and its giving better accuracy compare to propose algorithm. The main reason of using GRU in place of LSTM in extension work as its provide better result in image features optimization compare to LSTM as LSTM good in temporal data processing but not good in image features optimization so we just experiment with GRU and got better result.

SCREEN SHOTS

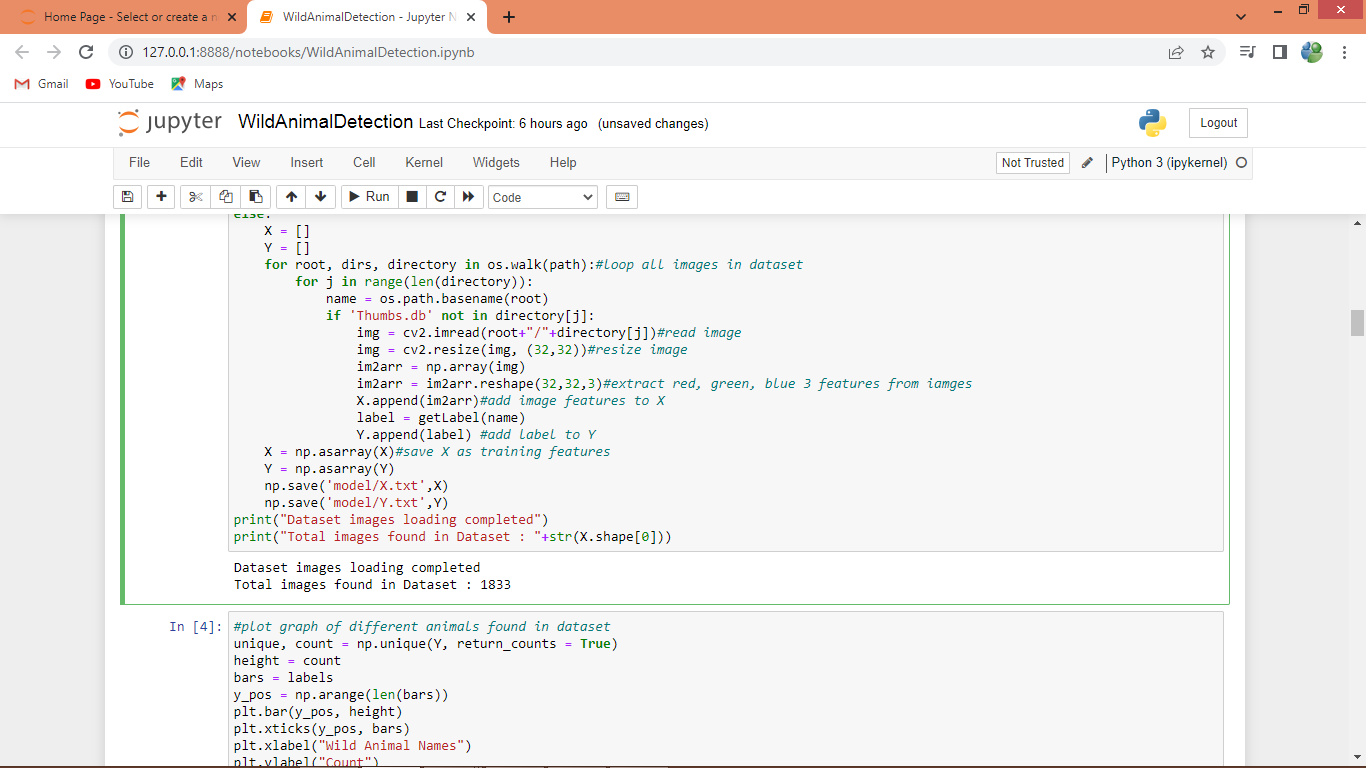
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments.



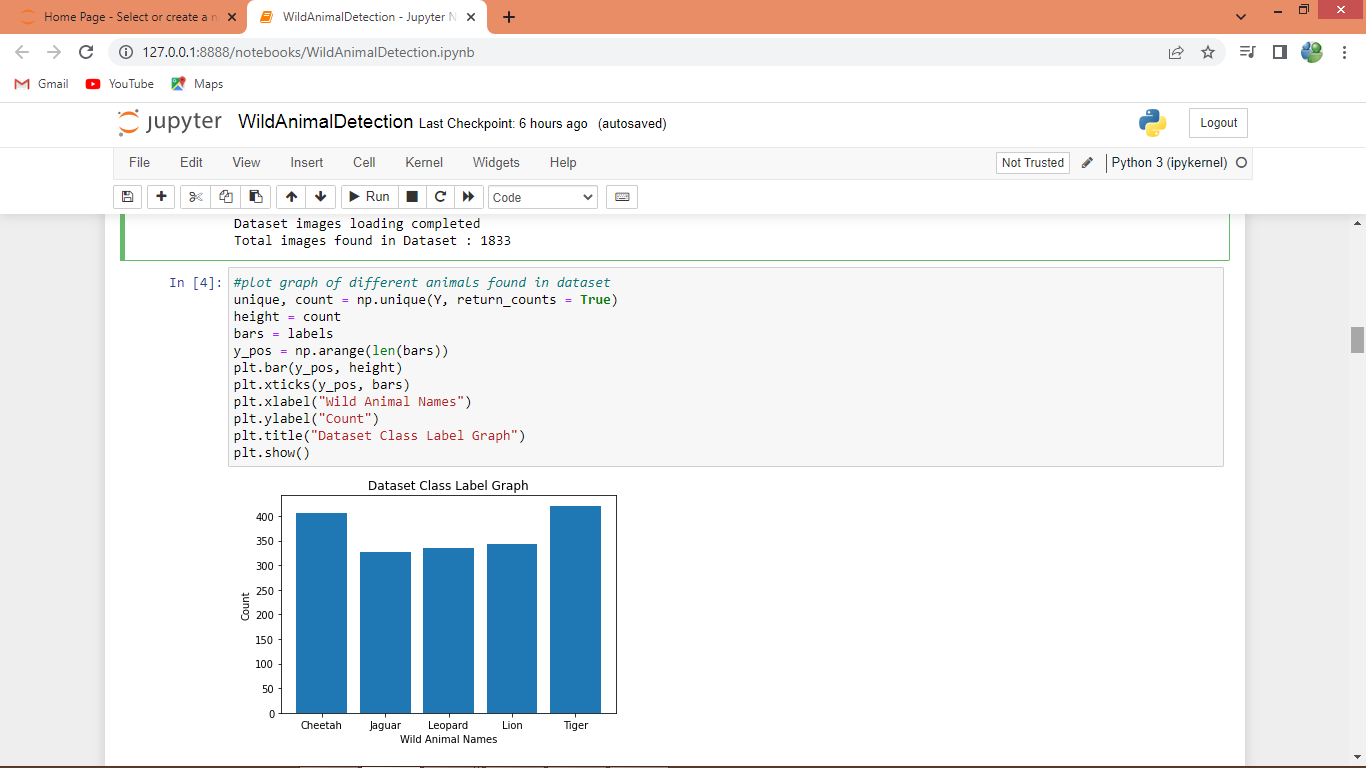
In above screen importing all packages and classes



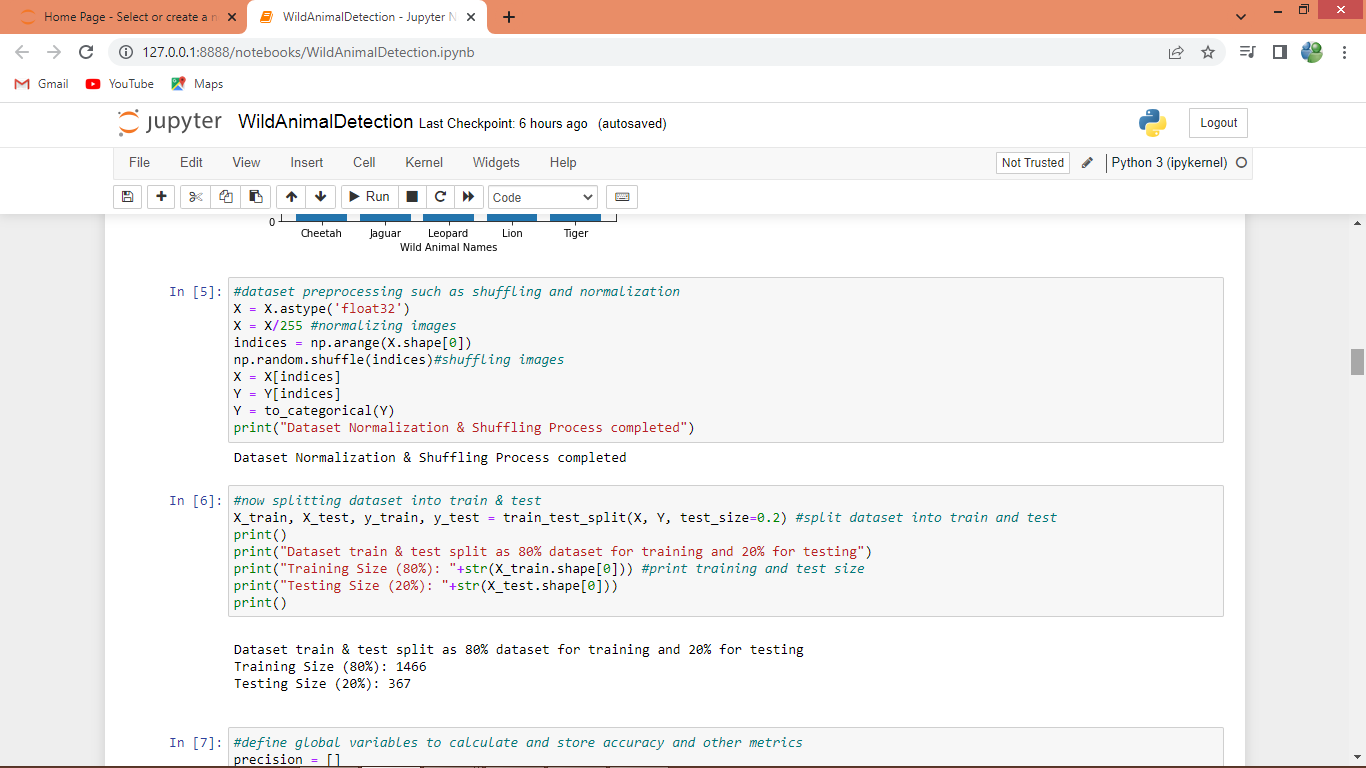
In above screen finding and displaying all wild animals names found in dataset



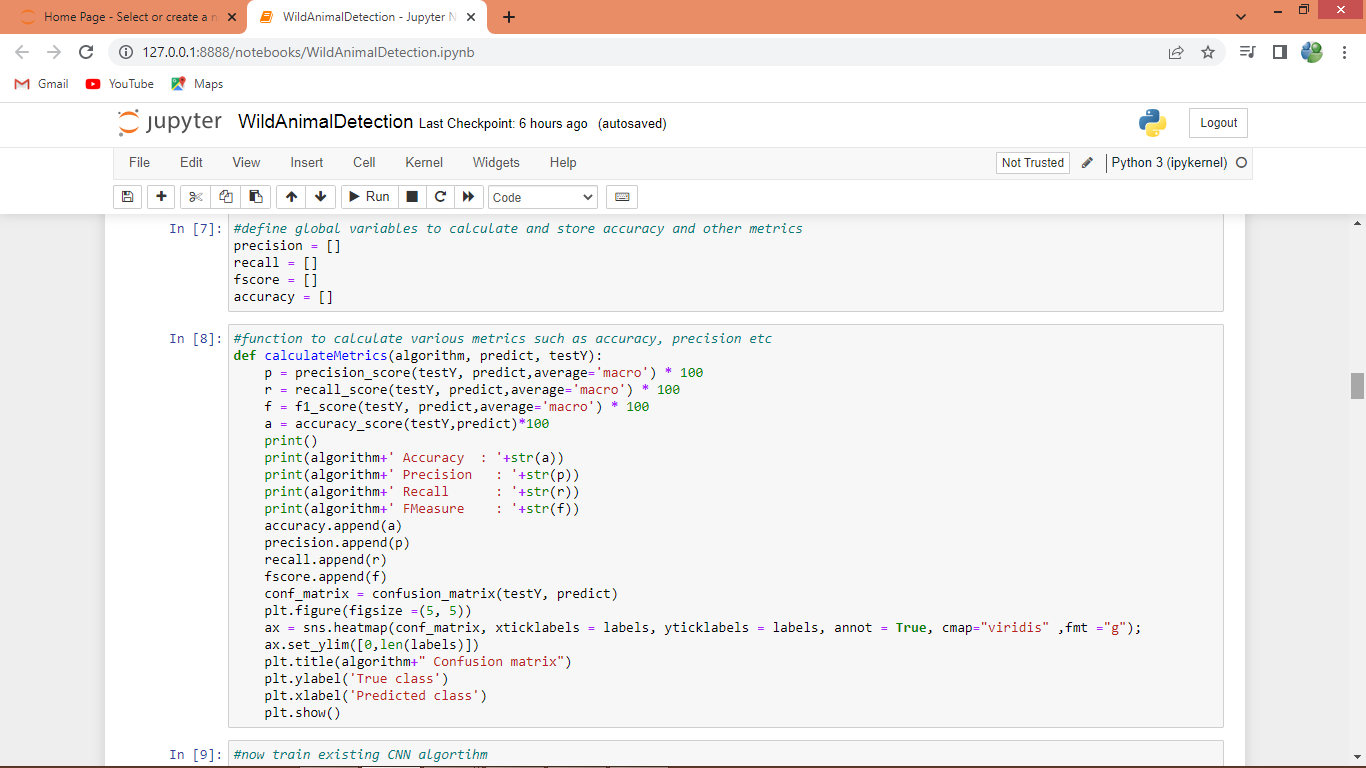
In above screen reading all images from dataset and then extracting features



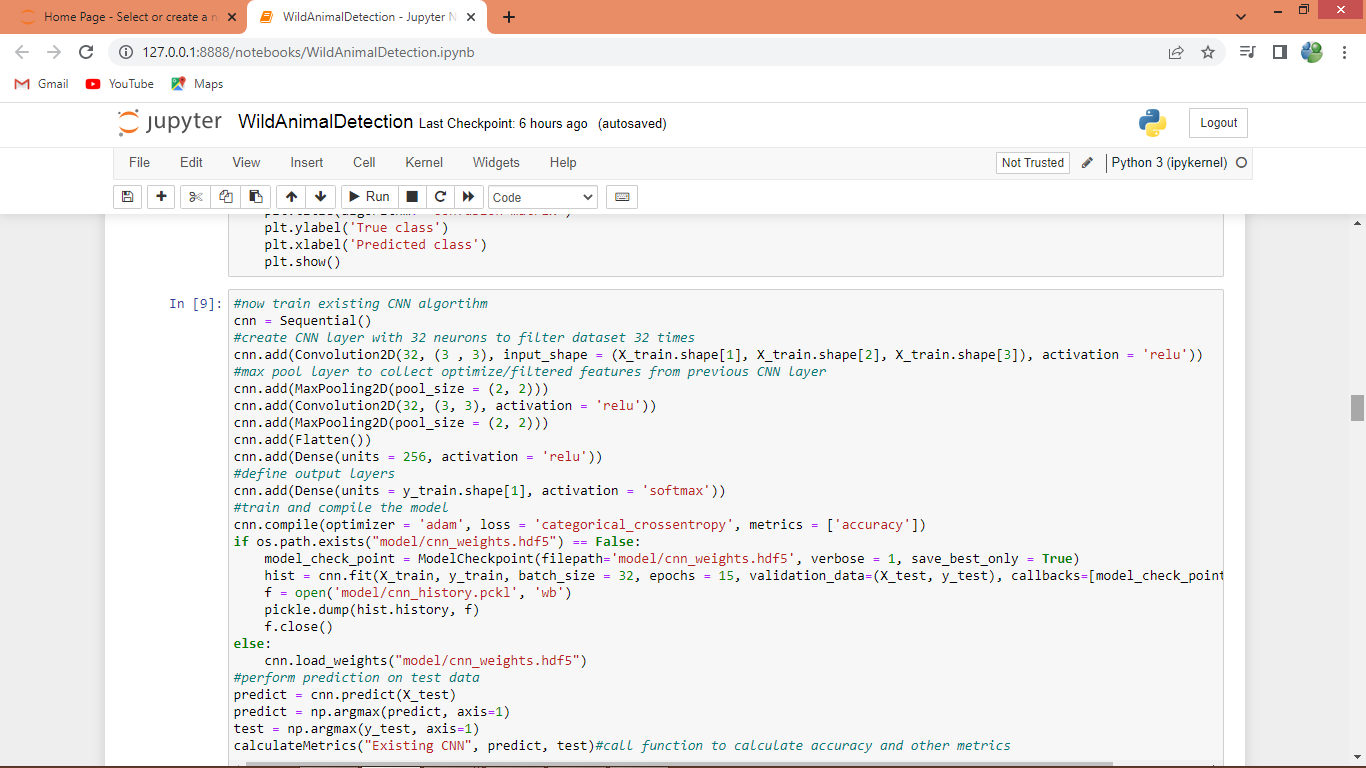
In above screen plotting graph of each animal count in dataset where x-axis represents animal names and y-axis represents counts



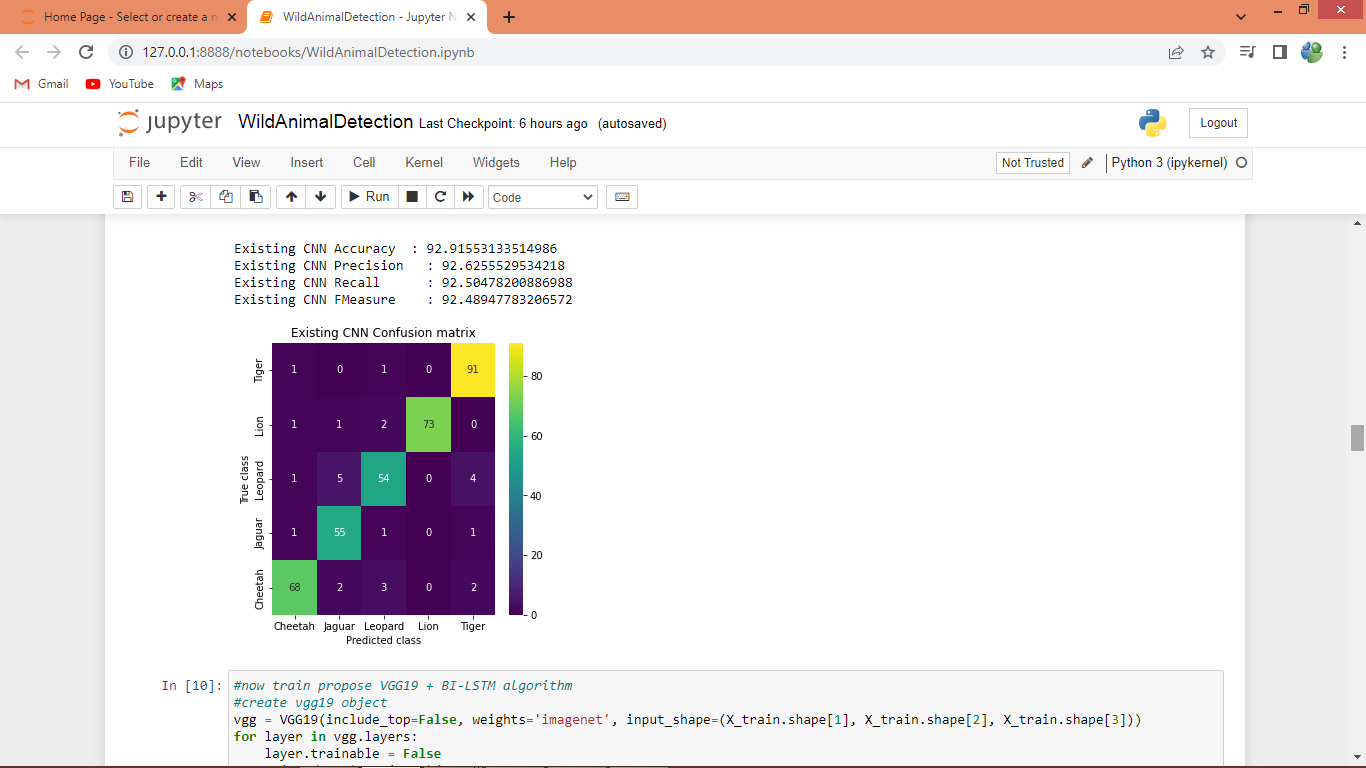
In above screen dataset processing such as shuffling, normalization and then splitting into train and test



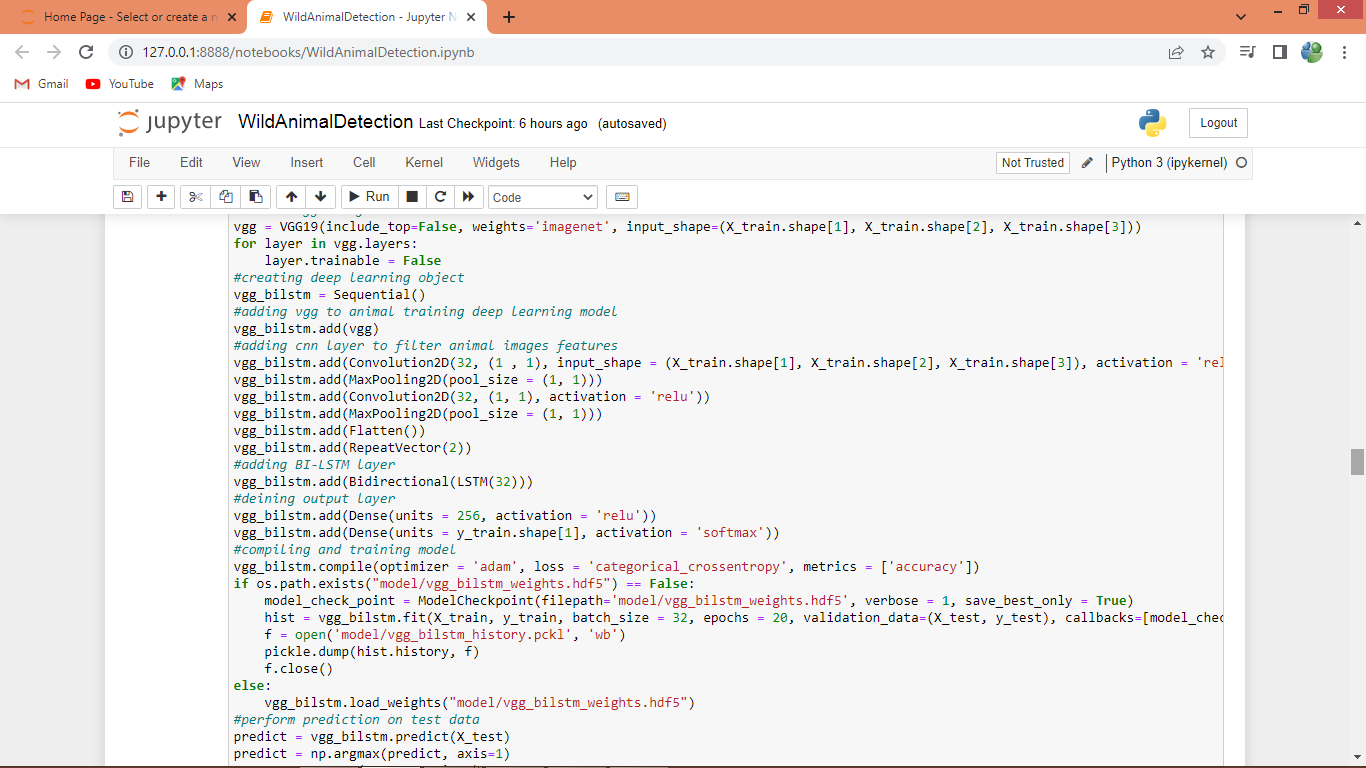
In above screen defining function to calculate accuracy and other metrics



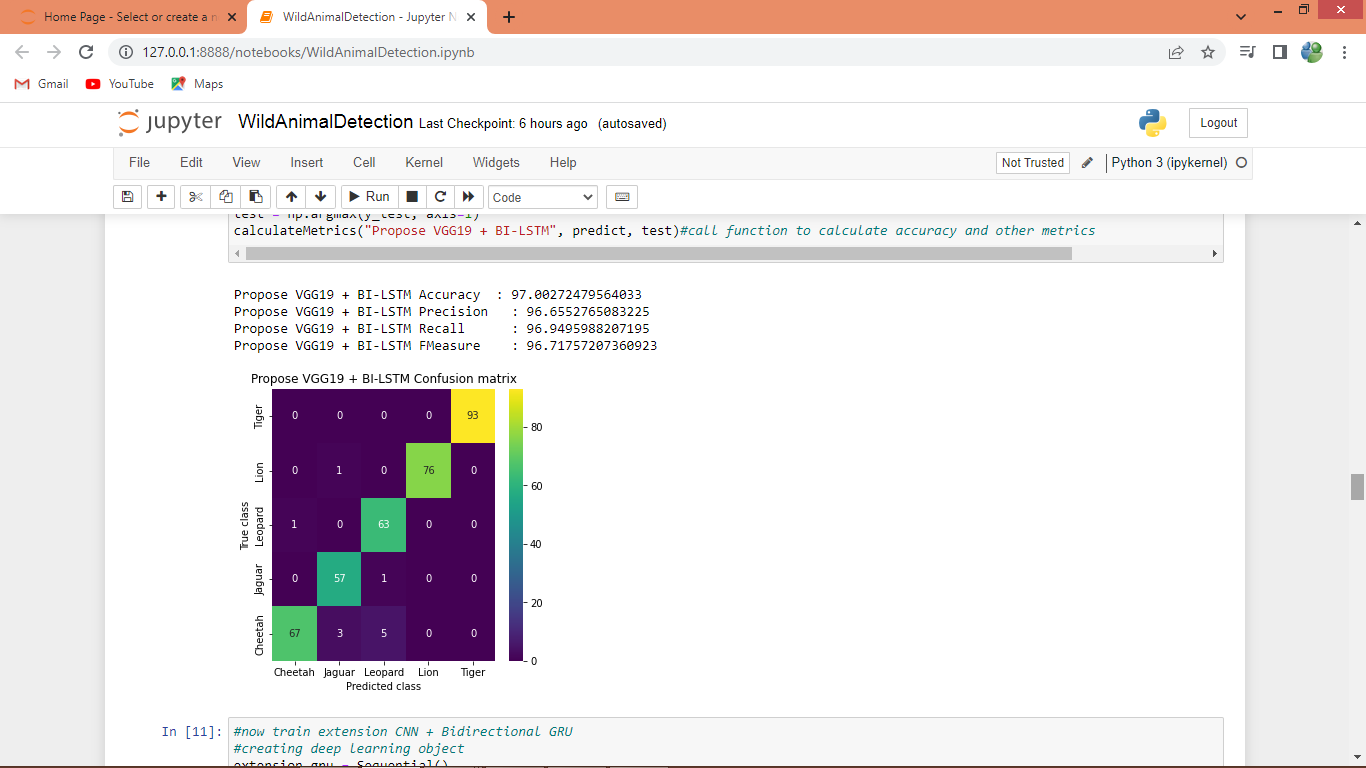
In above screen training existing CNN algorithm and after executing above block will get below output



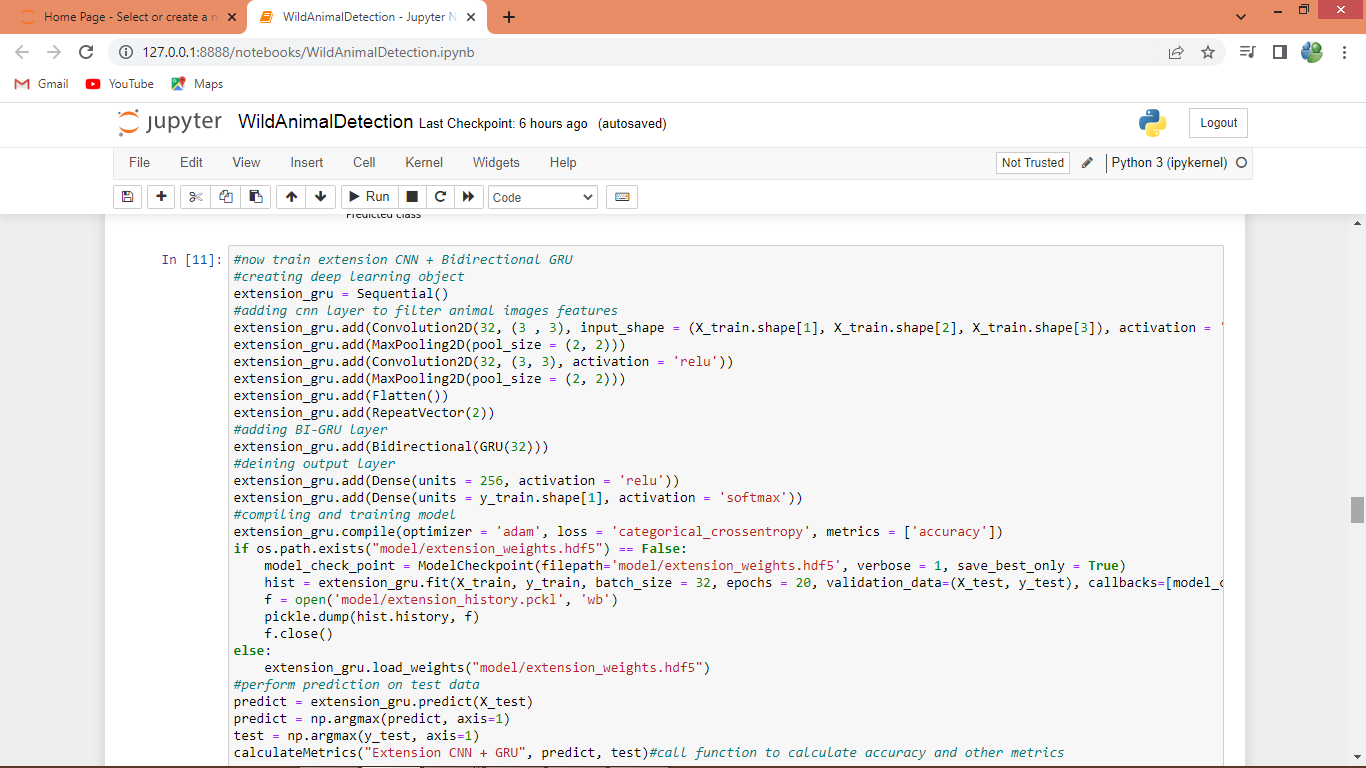
In above screen with existing CNN we got 92% accuracy and can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True labels and all different colour boxes in diagnol represents correct prediction count and blue boxes represents incorrect prediction count which are very few



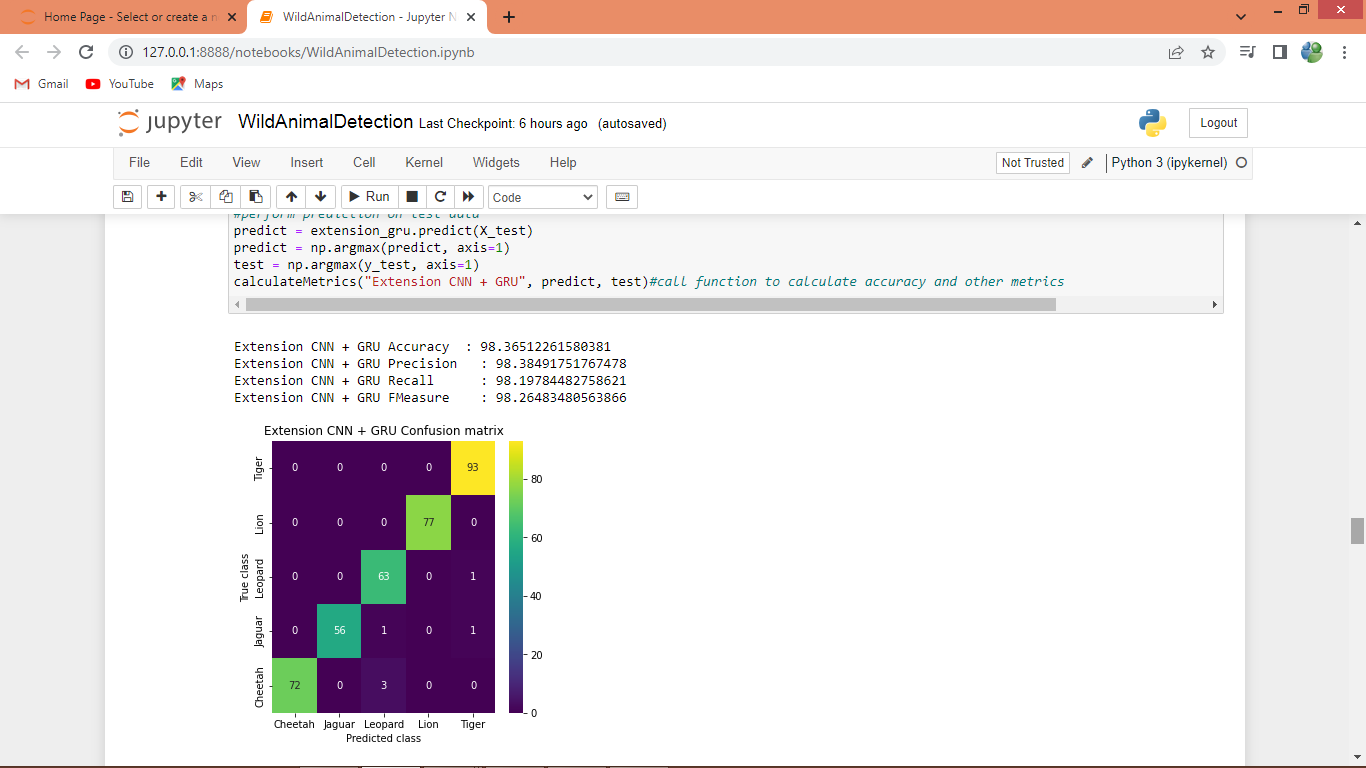
In above screen training propose VGG19 + BI-LSTM and after executing this block will get below output



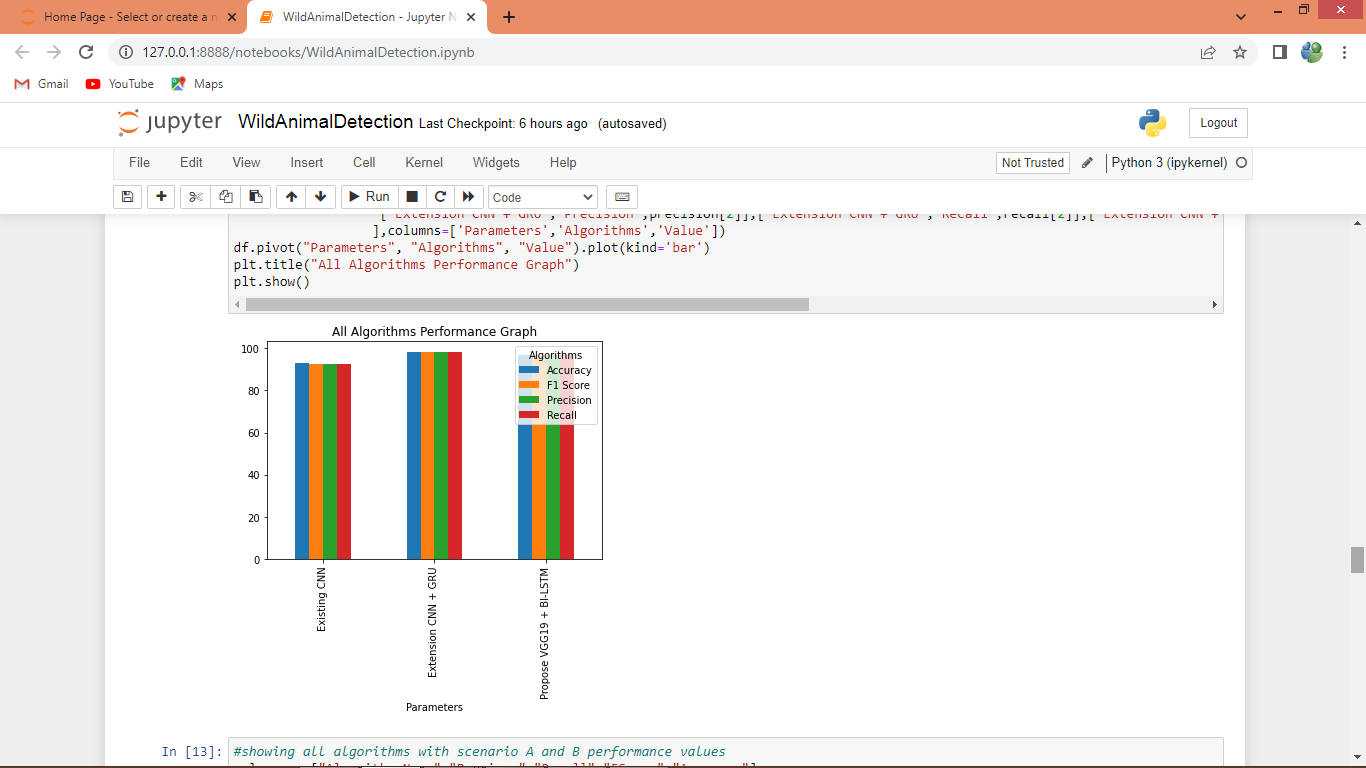
In above screen with propose VGG19 + BI-LSTM got 97% accuracy and can see other metrics also



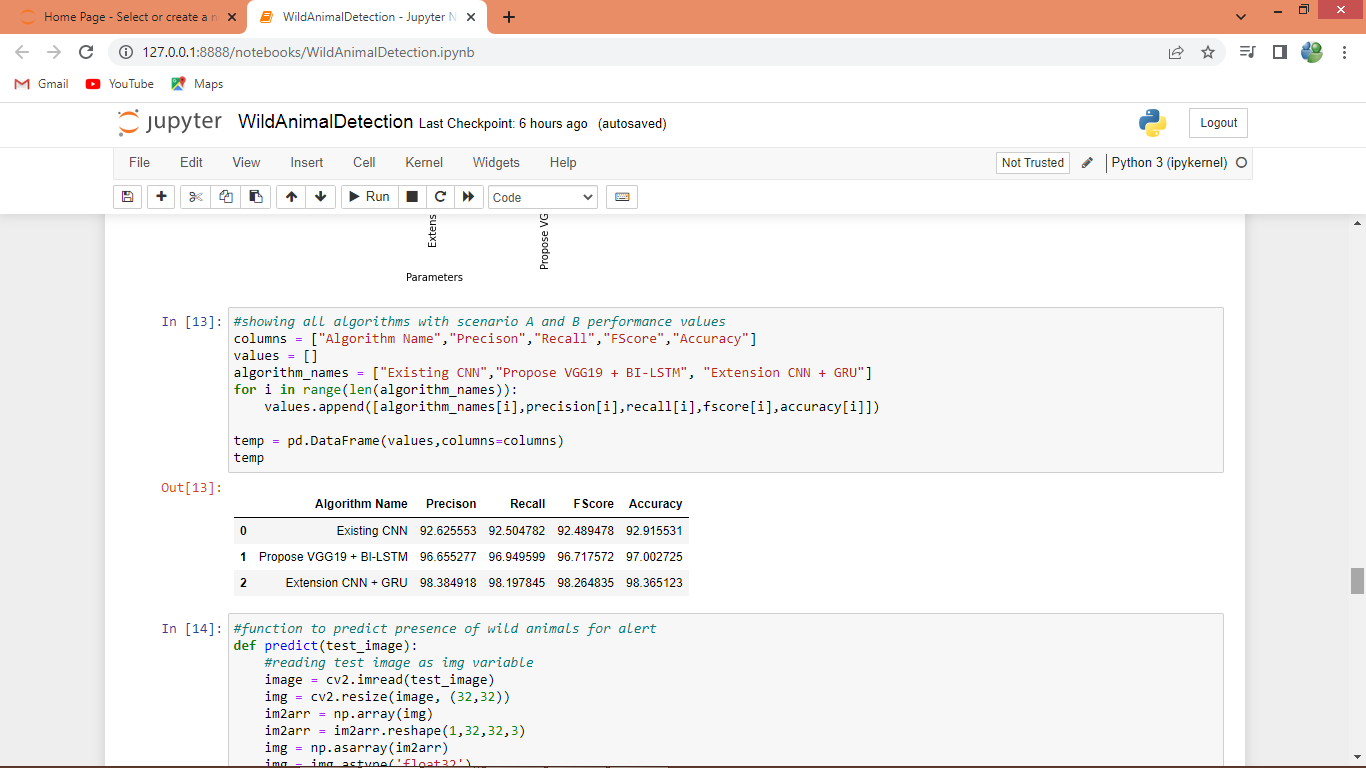
In above screen training CNN with GRU as extension and after executing above block will get below output



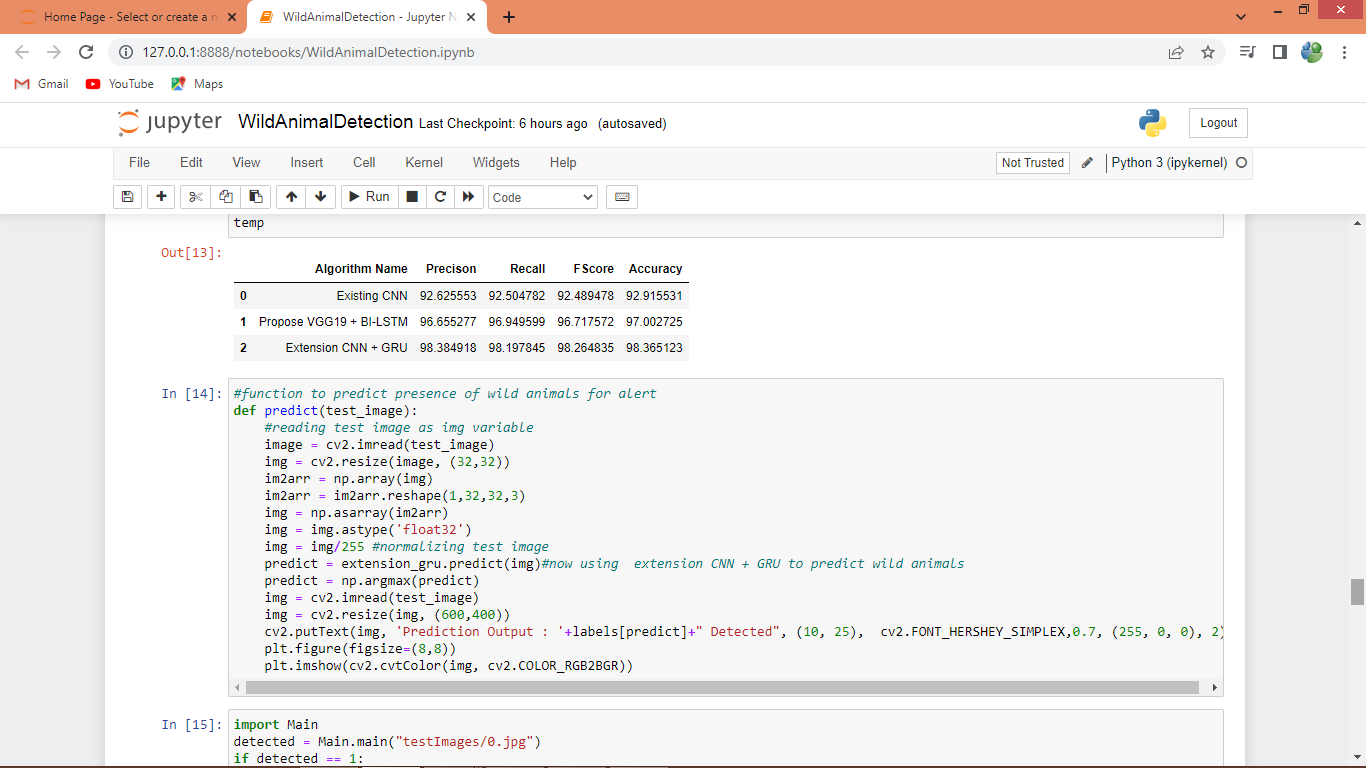
In above screen extension got 98% accuracy and can go above 99% also and in below screen we can see comparison graph



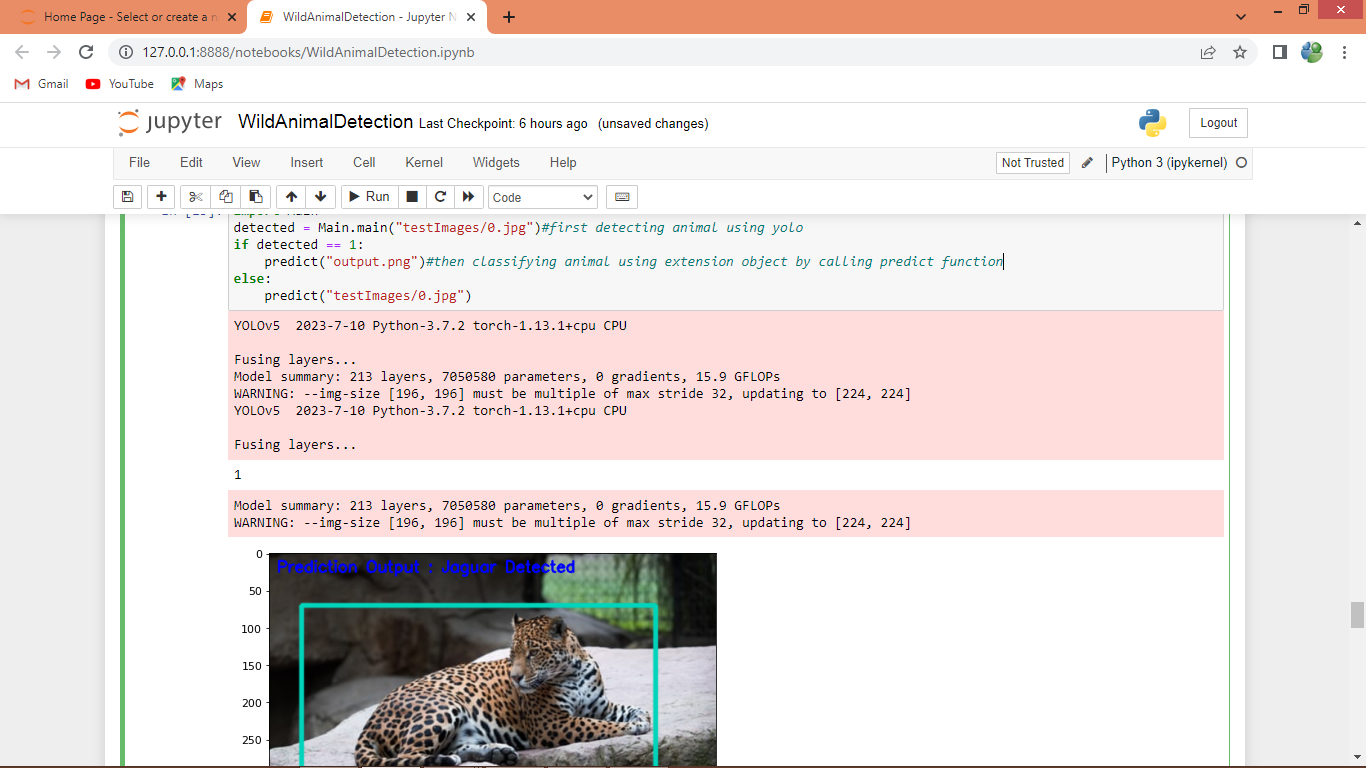
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms extension got high performance



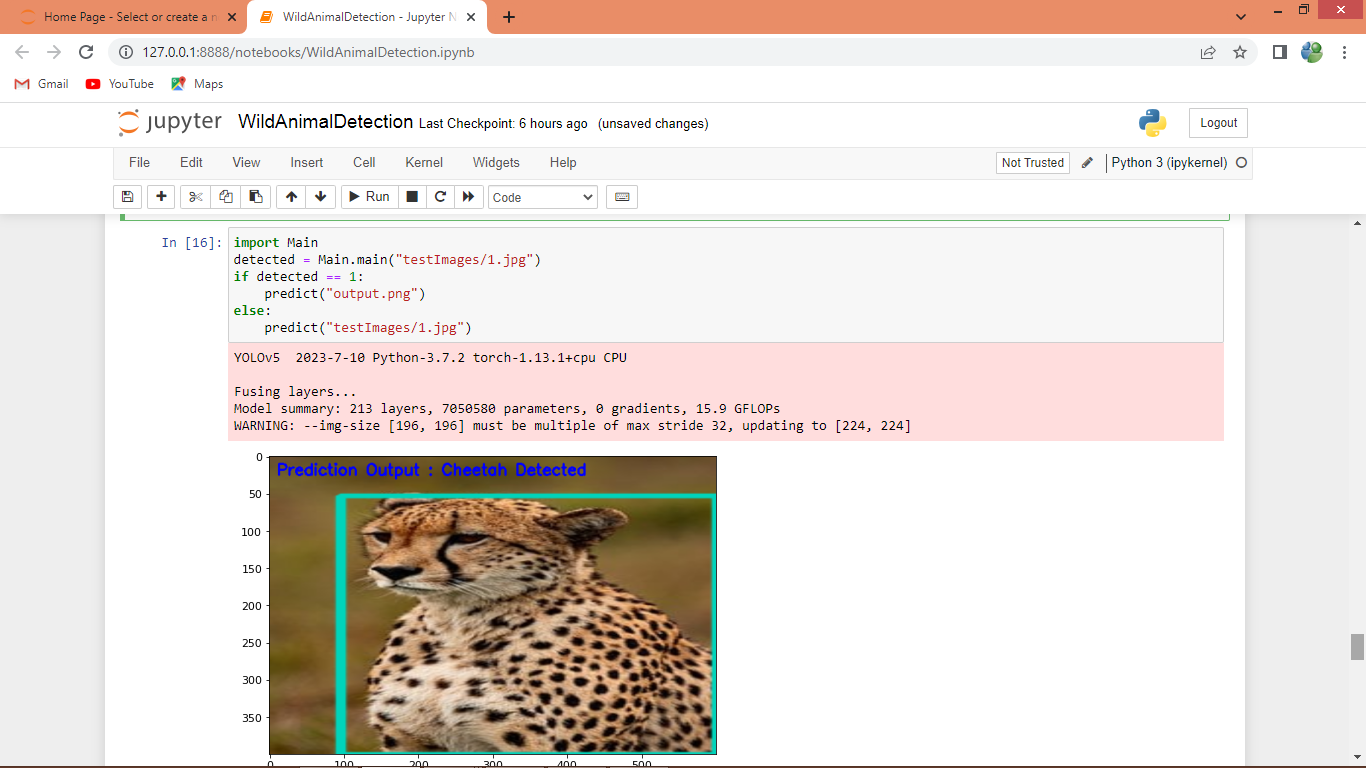
In above screen displaying all algorithm performance in tabular format



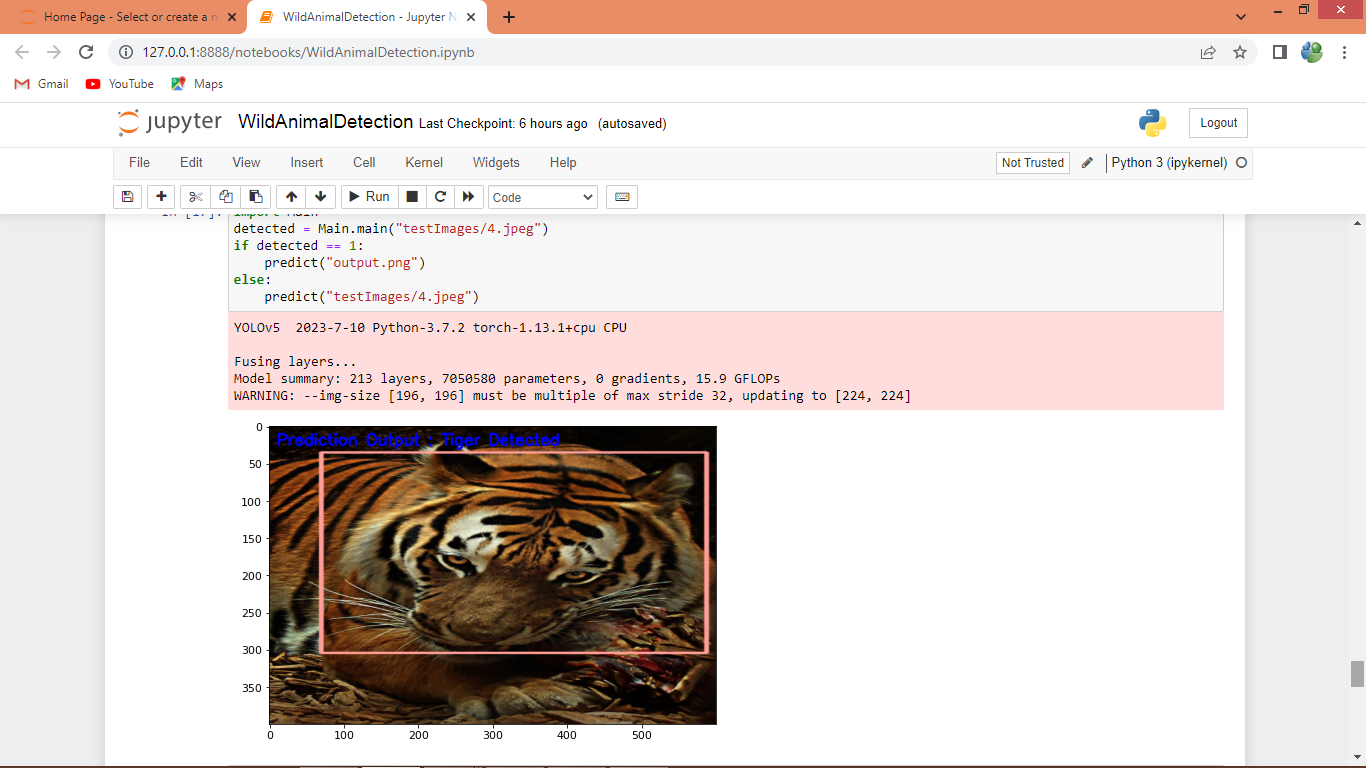
In above screen defining prediction function with extension object to predict and detect animals from test images



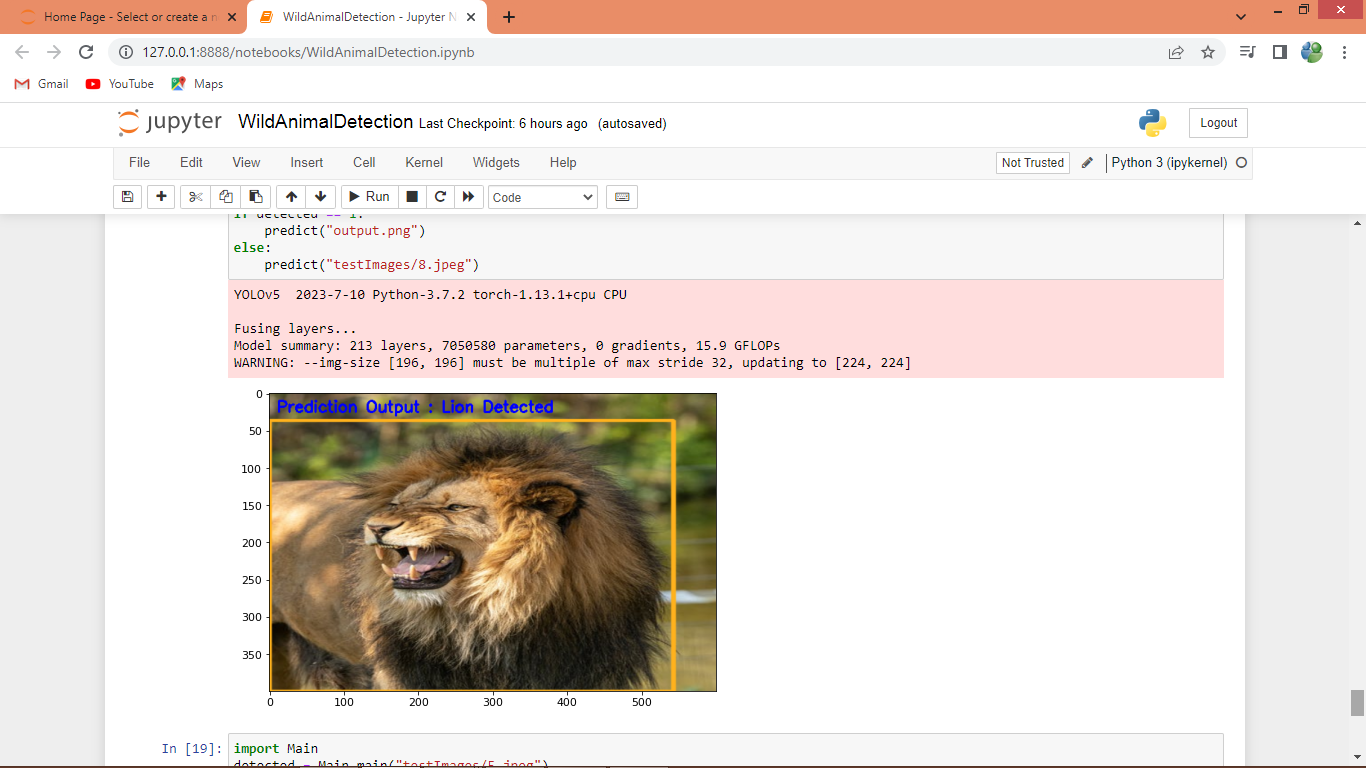
In above screen showing predict function output where bounding box predicted using YOLO and then in blue colour text showing animal classification with CNN + GRU extension model



In above screen cheetah is predicted



In above screen tiger is detected



In above screen lion is detected