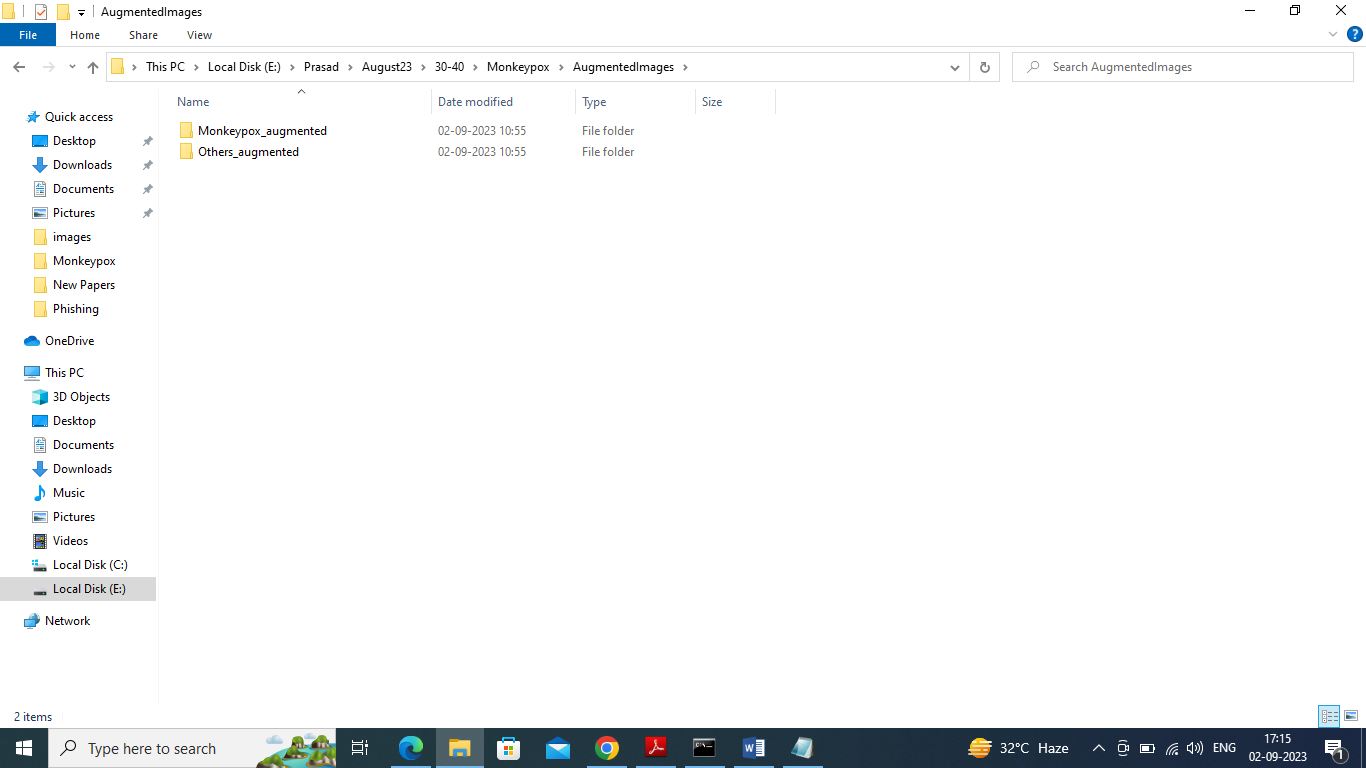
Monkeypox Diagnosis With Interpretable Deep Learning

After Covid19 world has seen rise in the Monkeypox disease also known as Monkeypox Virus (MPXV) disease, is caused by infection with the virus of the same name and is usually found in monkeys. In propose paper author employing deep learning algorithms to predict Monkeypox from skin images as deep learning prove its success in almost all fields for accurate prediction.

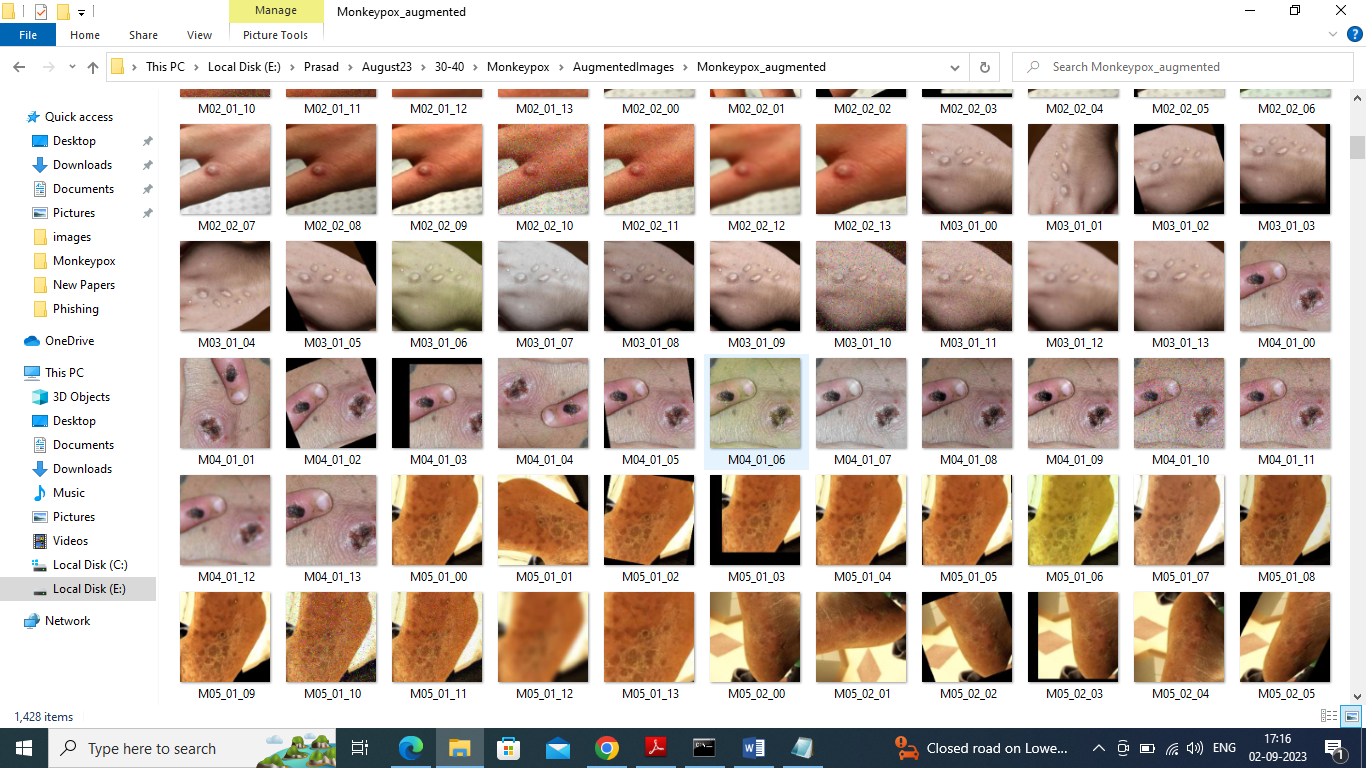
In propose author utilized various pre-trained deep learning algorithms such as Resnet50, InceptionV3, VGG16, VGG19, MobileNetv2 and many more. All algorithms are modified with new layers such as Dense, Dropout and Global Average Pool layer. Among all algorithms Modified MobileNetV2 and VGG19 are the top performing models.

Each algorithm performance is evaluated in terms of accuracy, precision, recall and FSCORE. After training models were interpreted using Local Interpretable Model Agnostic Explanations (LIME) algorithm. LIME will analyse trained model and training features to extract and plot all those features which played or helped algorithm in predicting particular label. This interpretation will help us in knowing which features contribute more in prediction.

To train all algorithms author has used own collected Monkeypox dataset but not publish on internet so we have used available dataset from KAGGLE which contains two different classes such as Monkeypox and Non-Monkeypox and below is the dataset details



In above screen we have two folders with augmented images (normally dataset has only 200 images so by using augmented technique we have increased to 1000). Just go inside any folder to view images like below screen



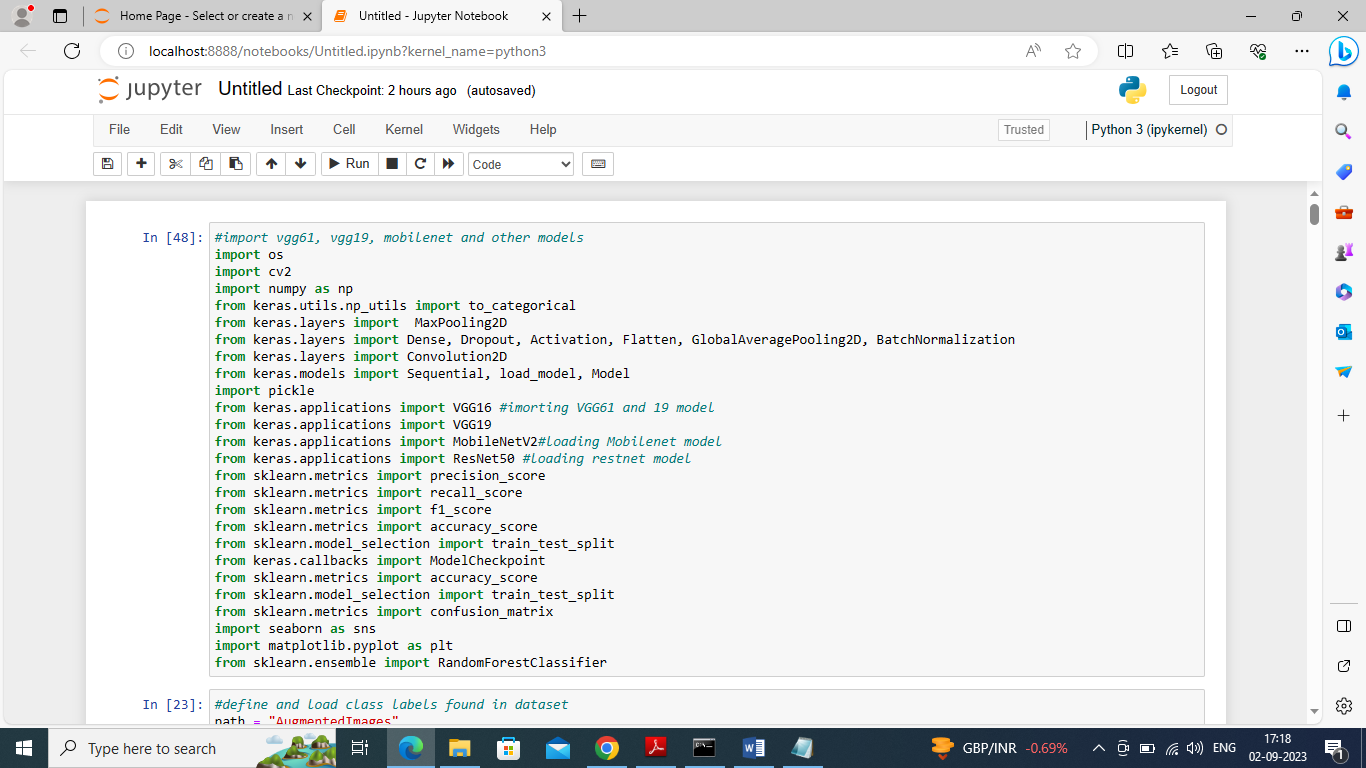
So by using above images we will train and test all algorithm performances.

Extension Concept

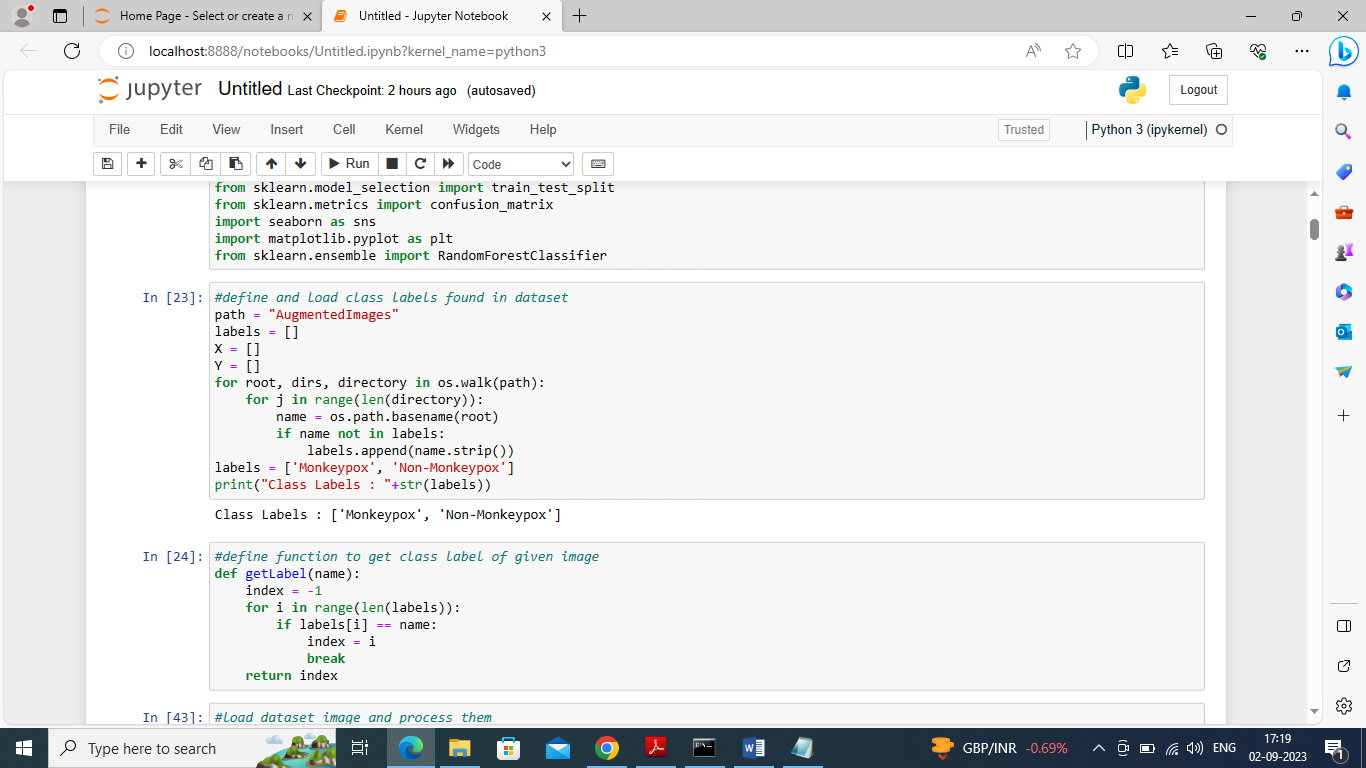
As extension we have extracted optimized features from best propose model called MobilenetV2 and then retrained those optimized features with Random Forest to further enhance accuracy and this model obtaining 100% accuracy on optimized features and this model is called as Hybrid Modified Extension MobileNetV2.

SCREEN SHOTS

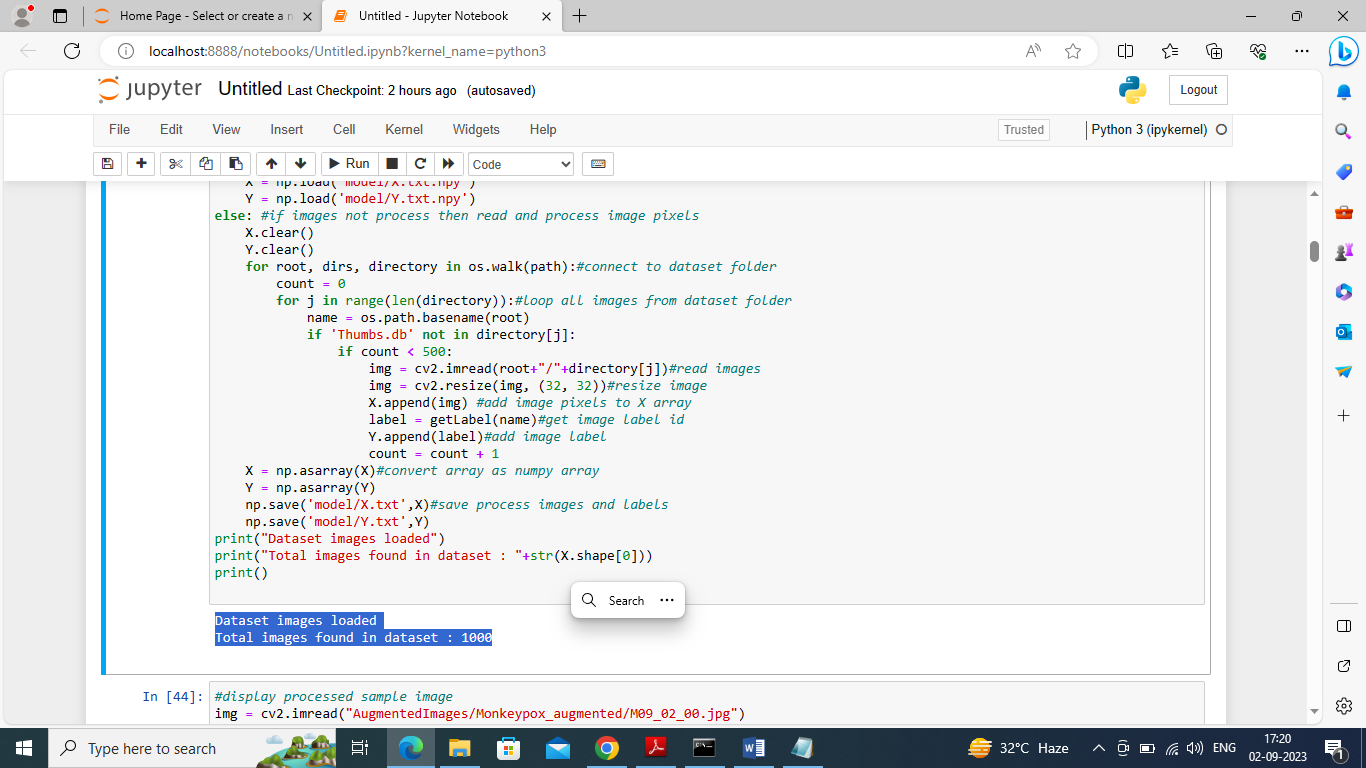
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



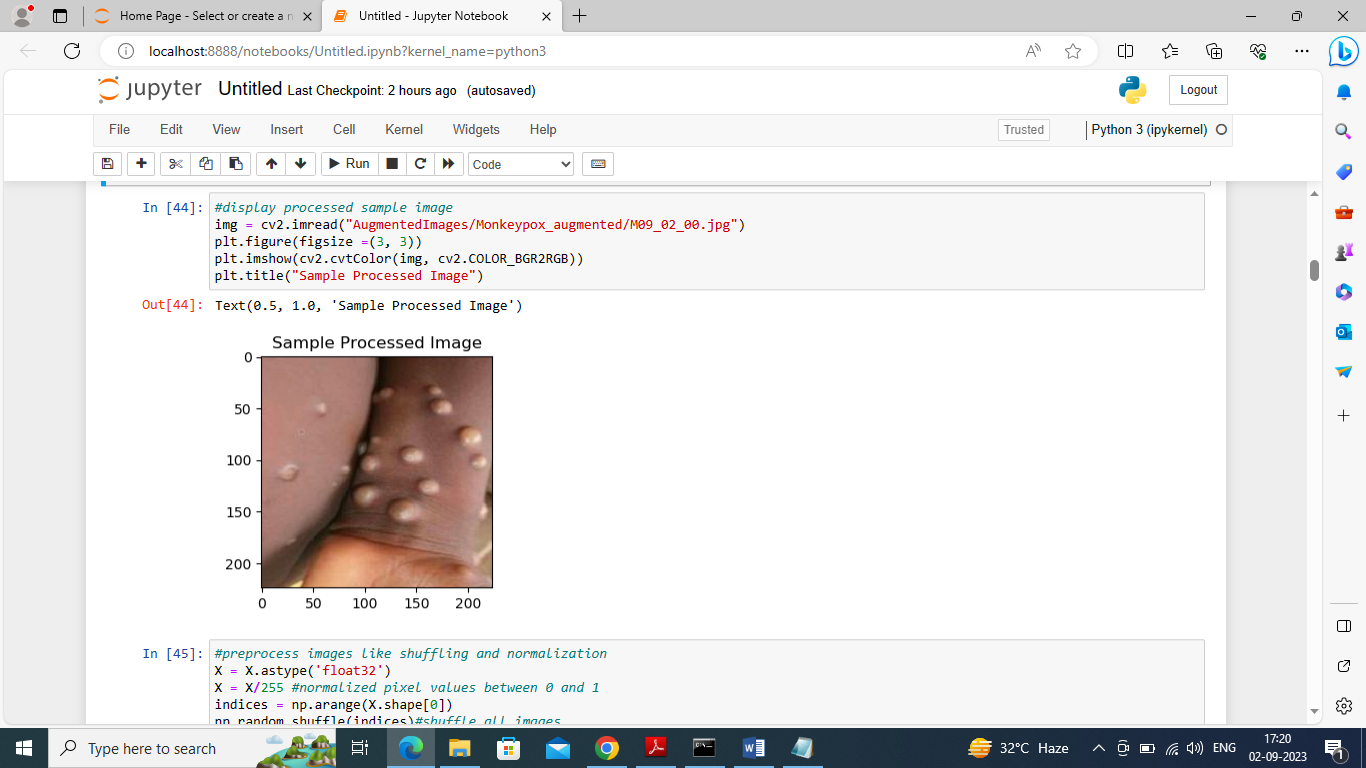
In above screen importing required python classes and packages



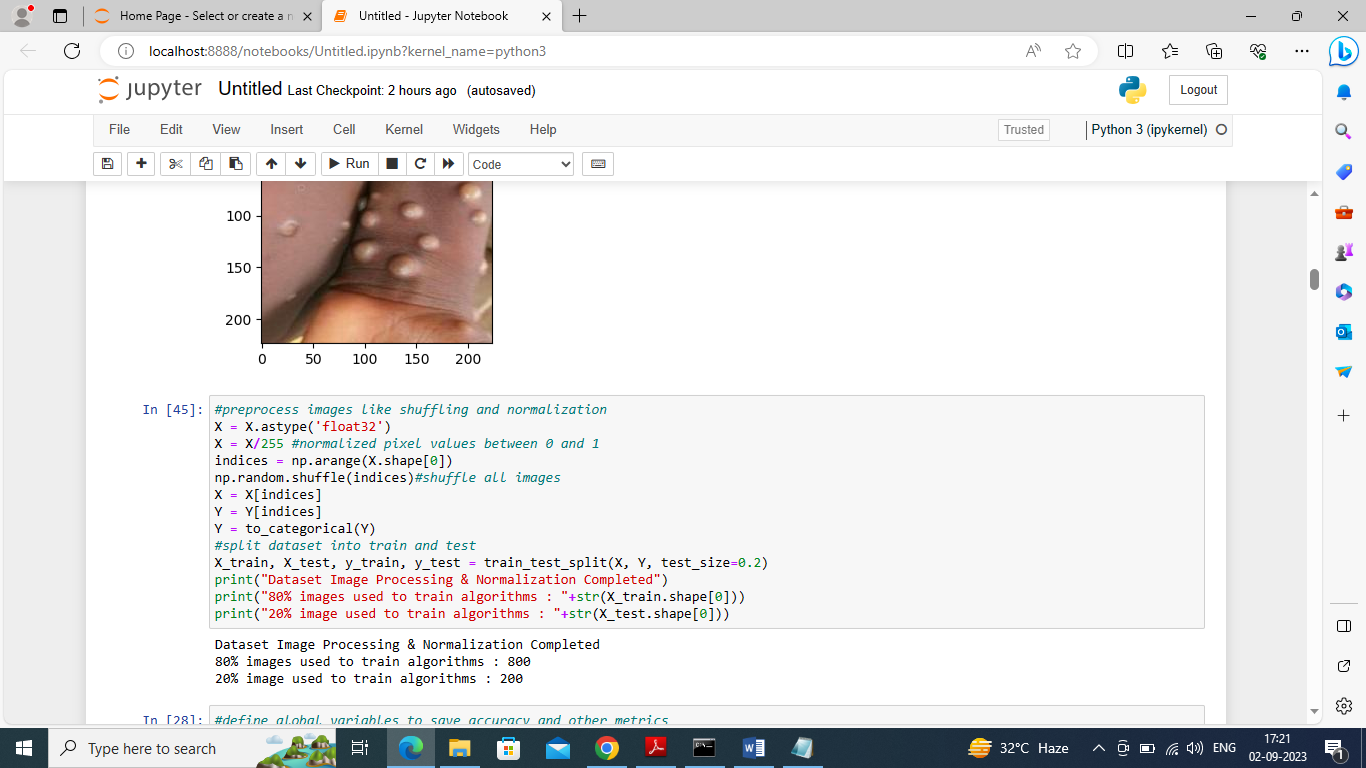
In above screen finding and displaying available classes in dataset and then defining function to get Integer class label ID of given image



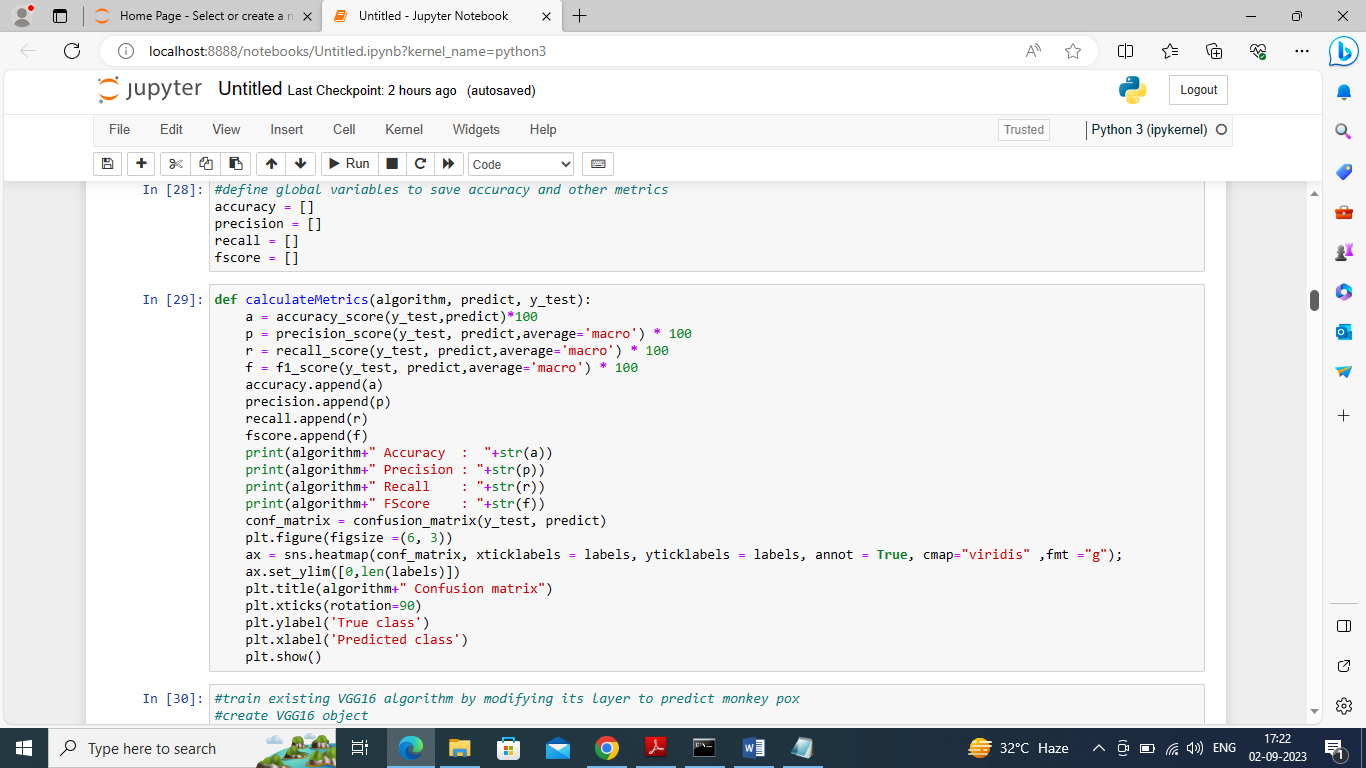
In above screen connecting to dataset folder and then looping and loading all images and then displaying total 1000 images loaded from the dataset folder



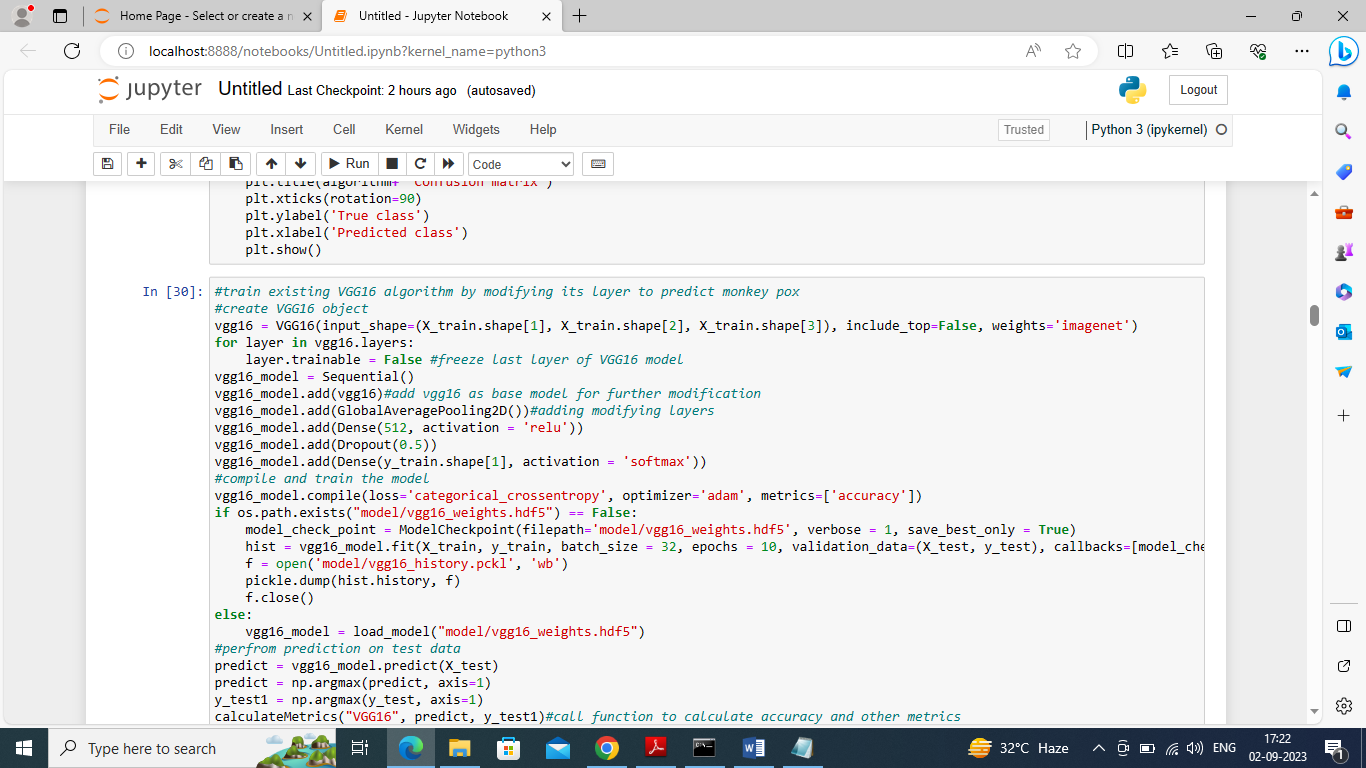
In above screen displaying sample loaded image



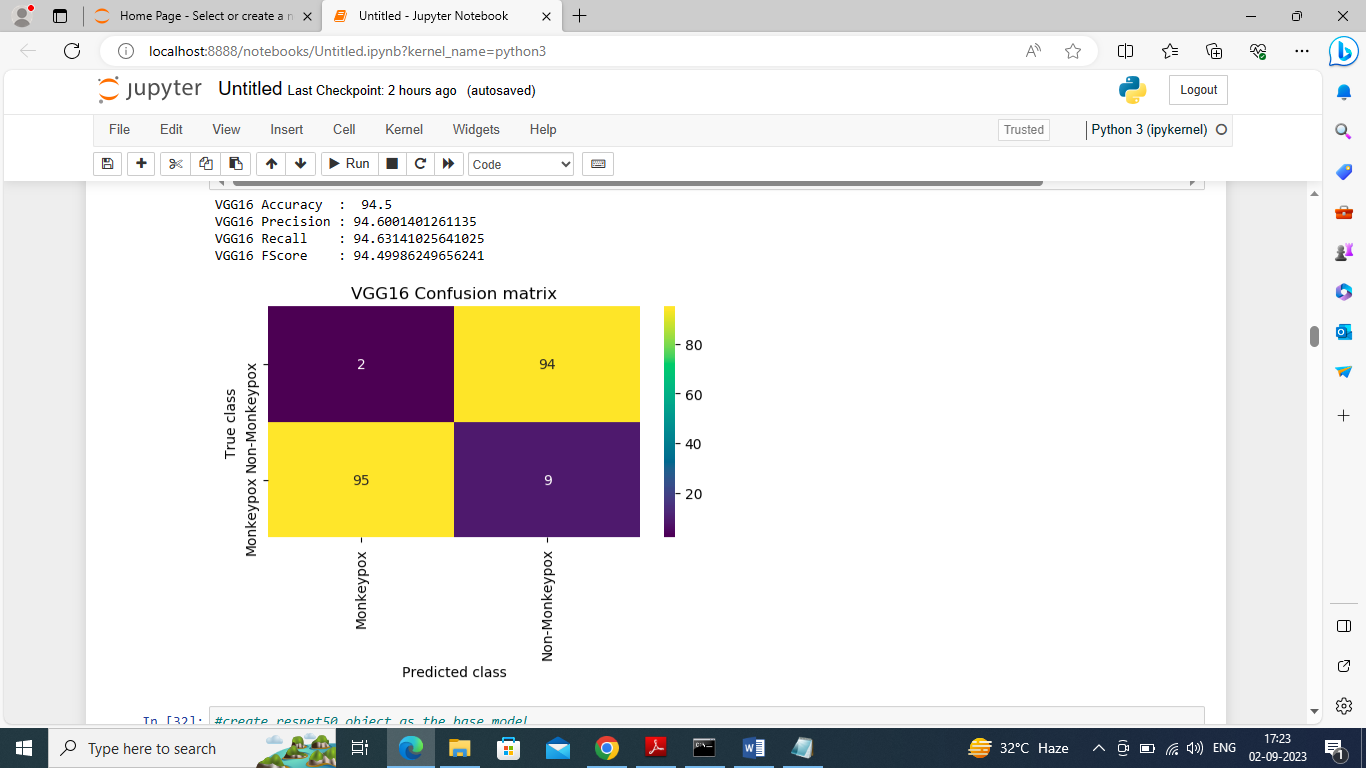
In above screen applying pre-processing techniques such as normalizing and shuffling images pixel values and then splitting dataset into train and test where application using 80% dataset for training and 20% for testing



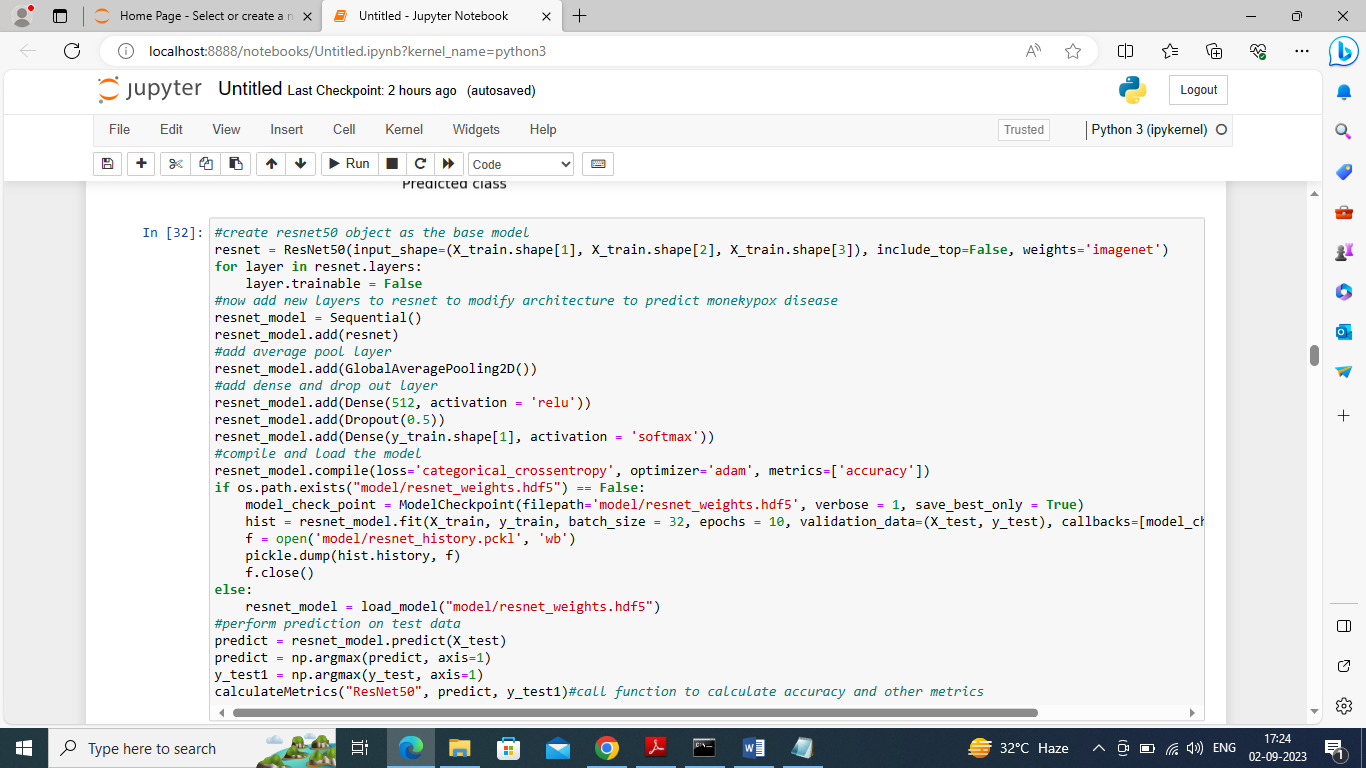
In above screen defining function to calculate accuracy and other metrics



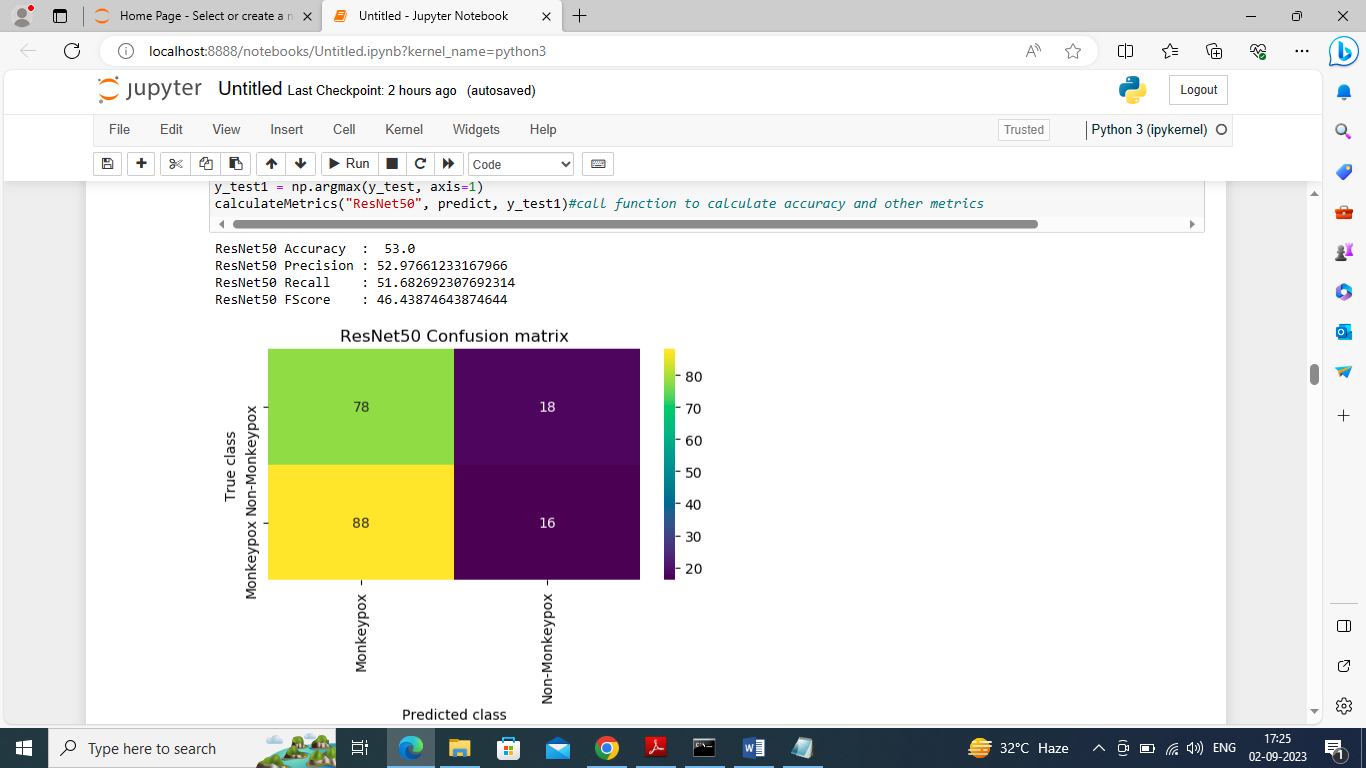
In above screen modifying VGG16 algorithm and then after executing above block will get below output



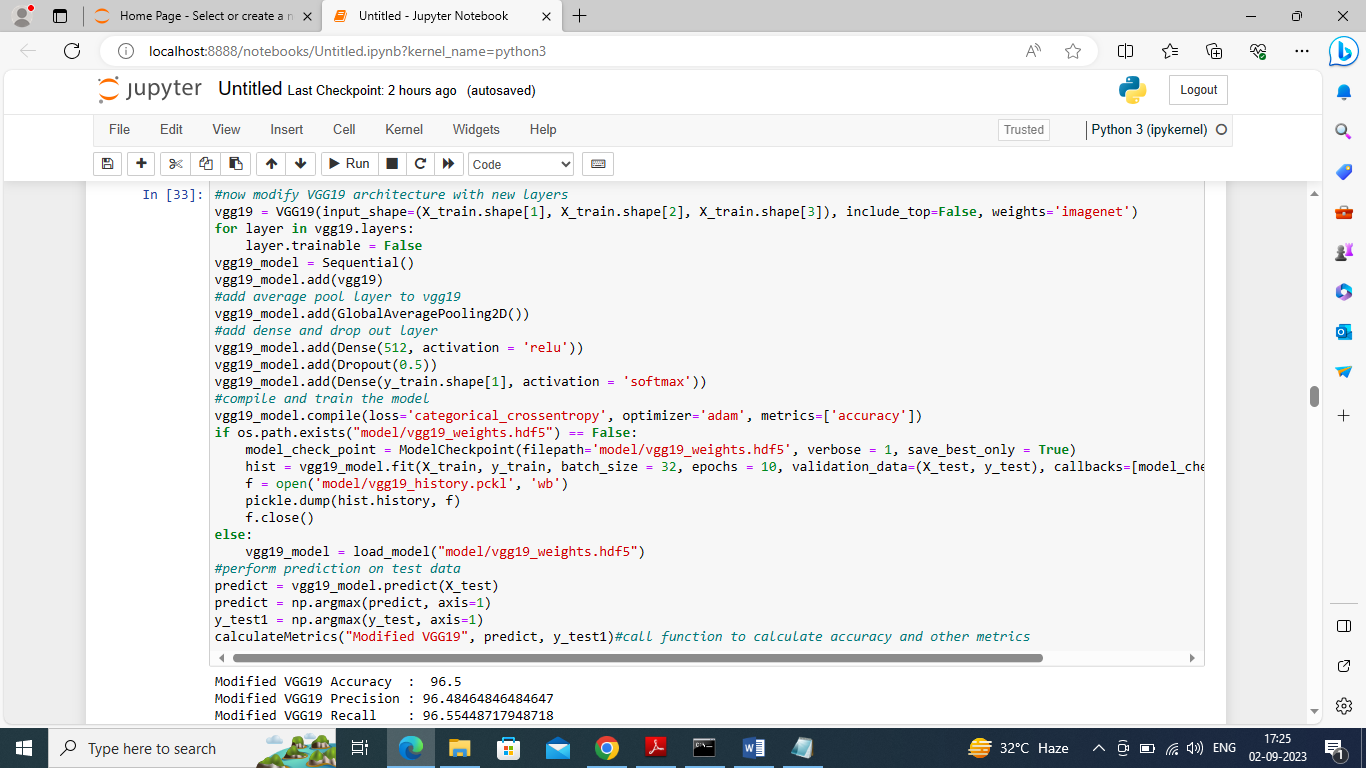
In above screen VGG16 got 94% accuracy and can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count



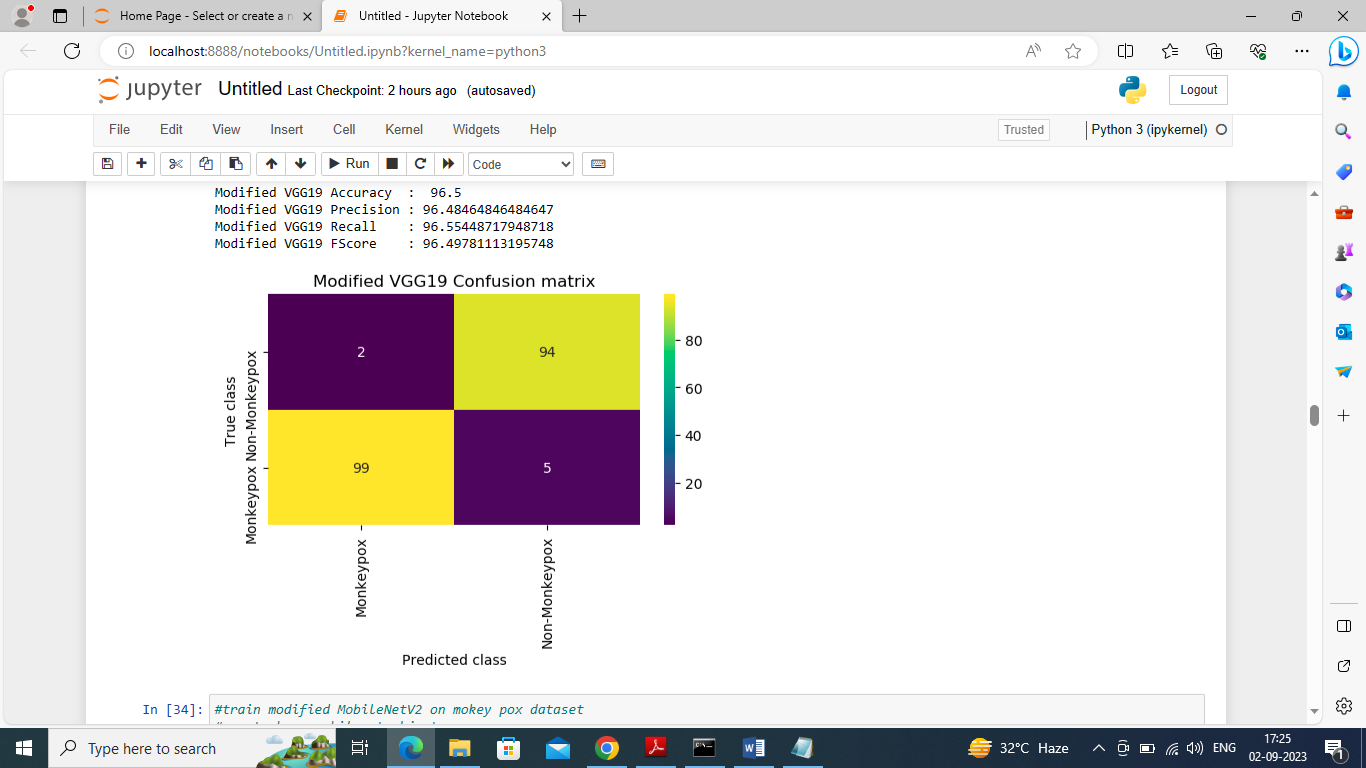
In above screen training Modified Resnet50 algorithm and below is the output



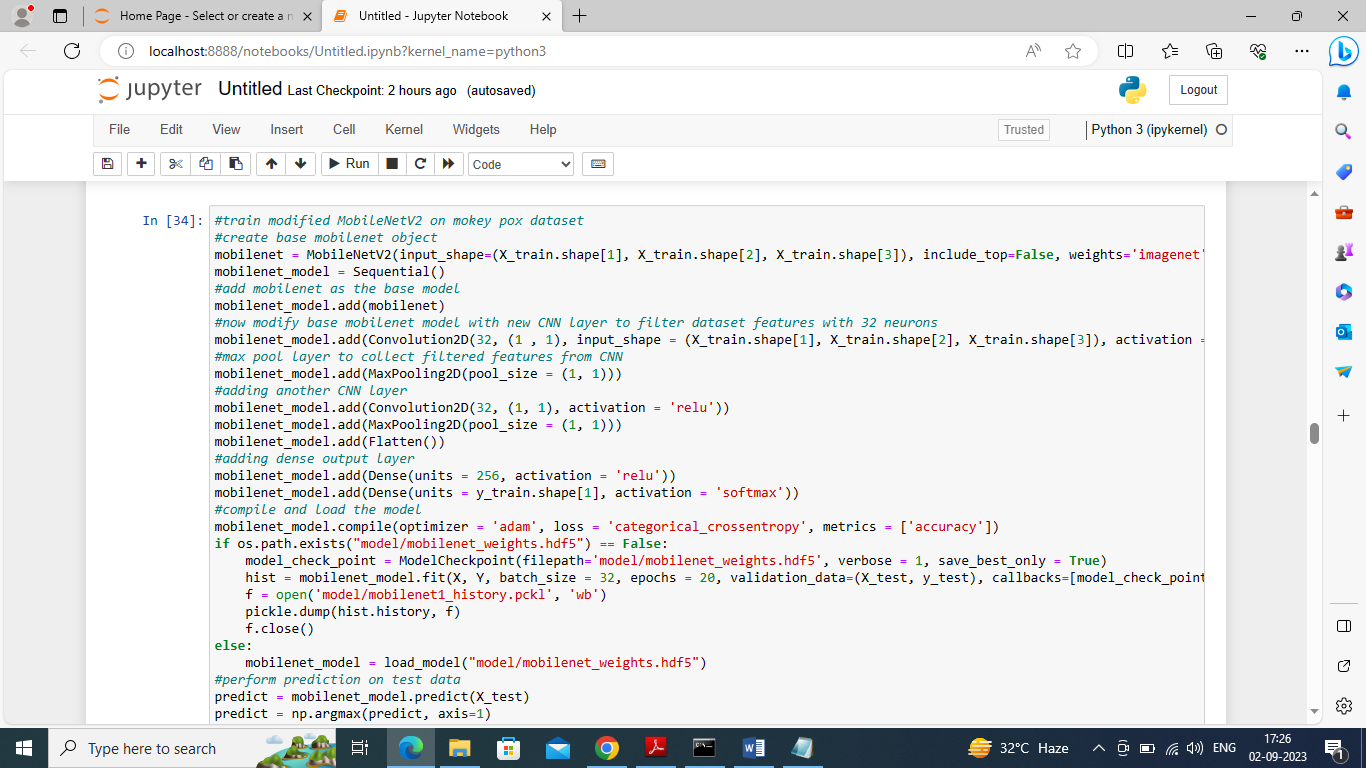
In above screen modified Resnet50 got 52% accuracy and performing poorly



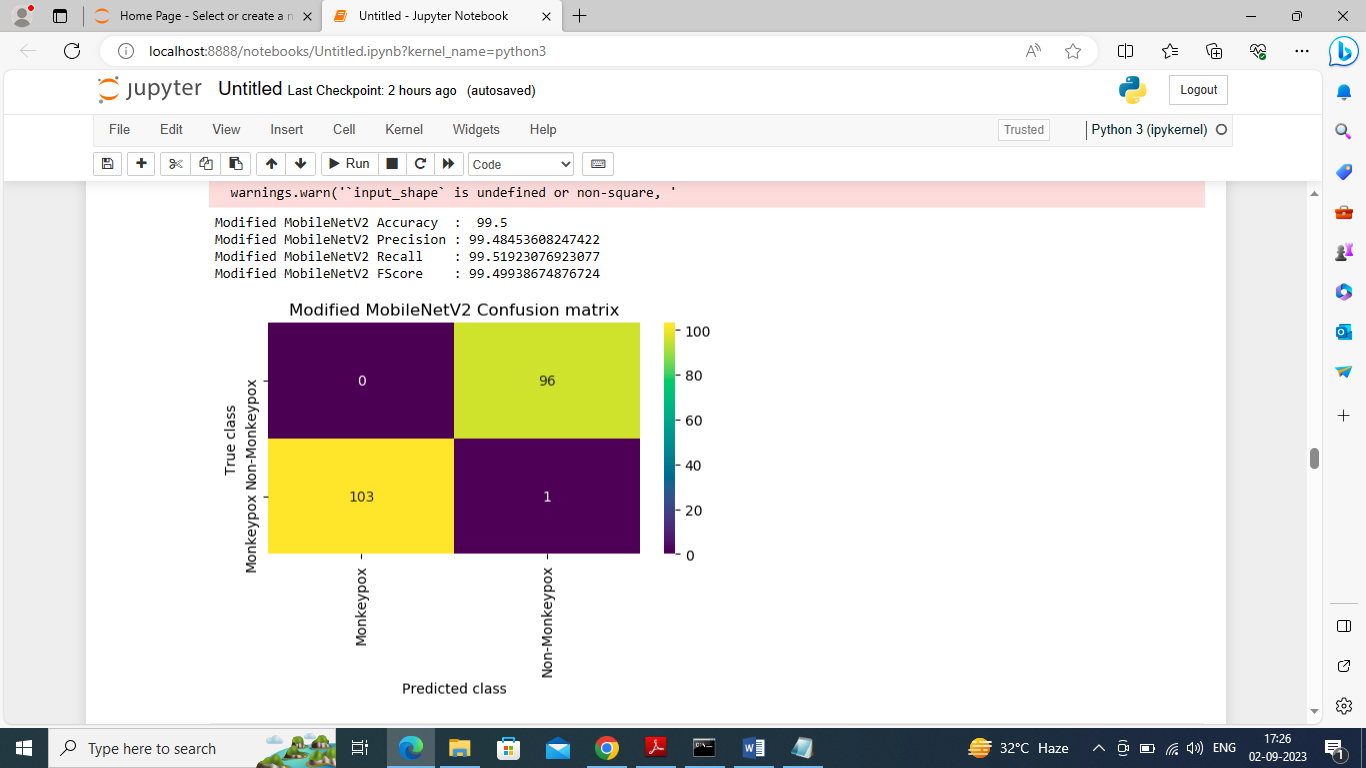
In above screen training modified VGG19 algorithm and below is the output



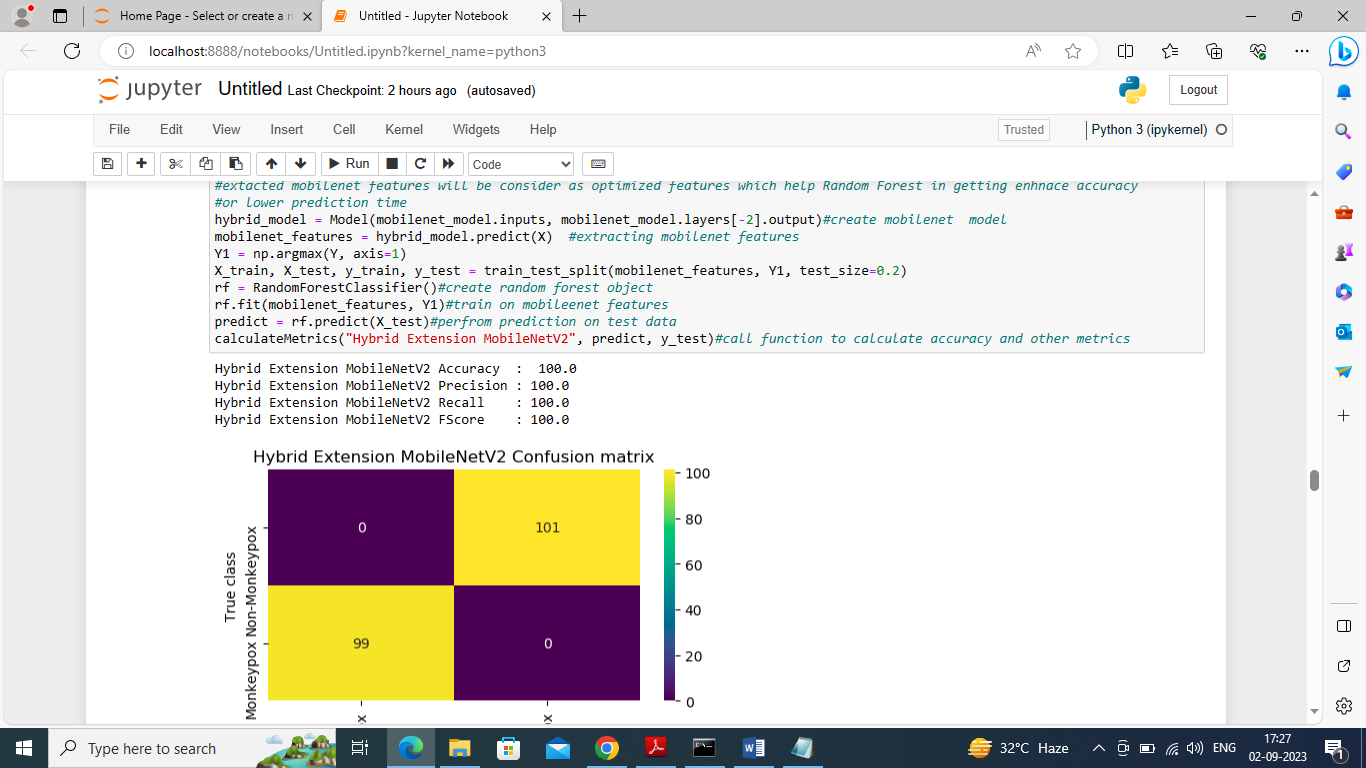
In above screen modified VGG19 got 96% accuracy



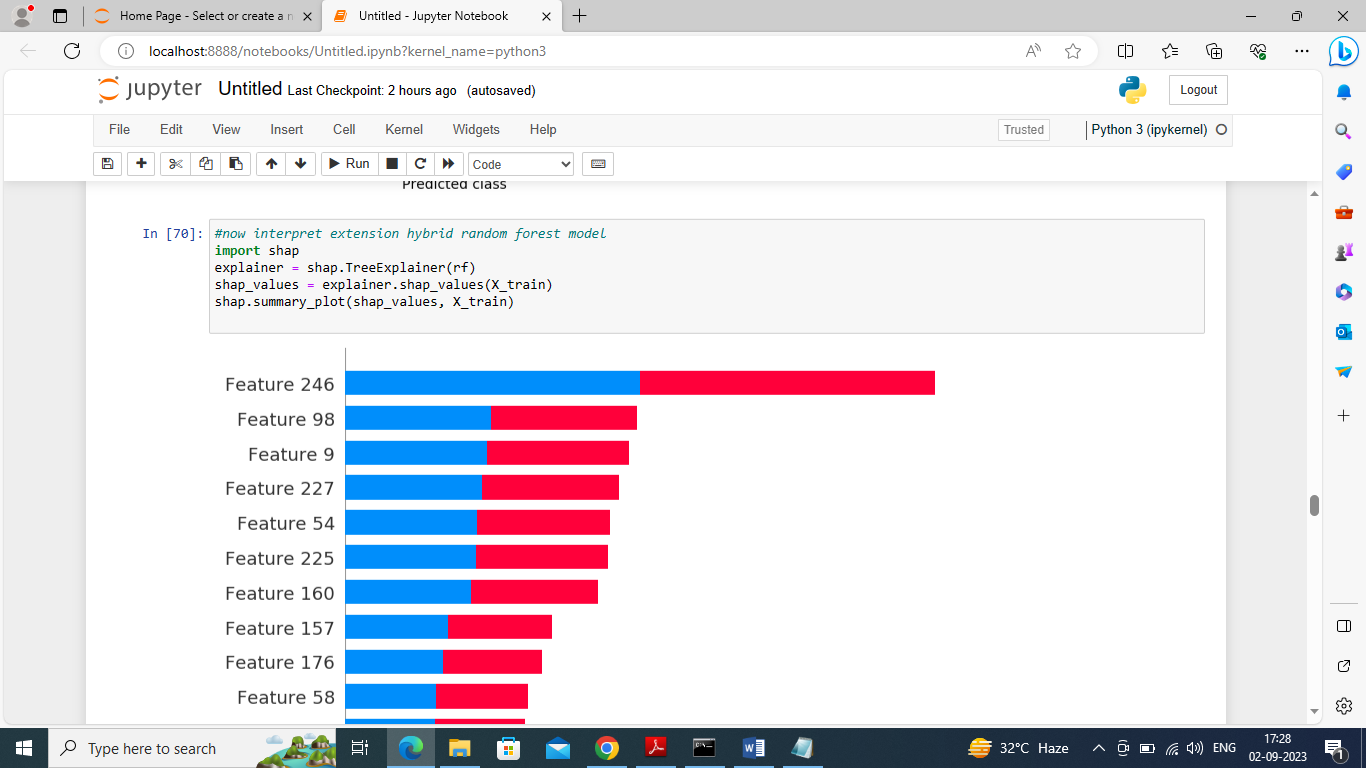
In above screen training modified MobileNetV2 algorithm and below is the output

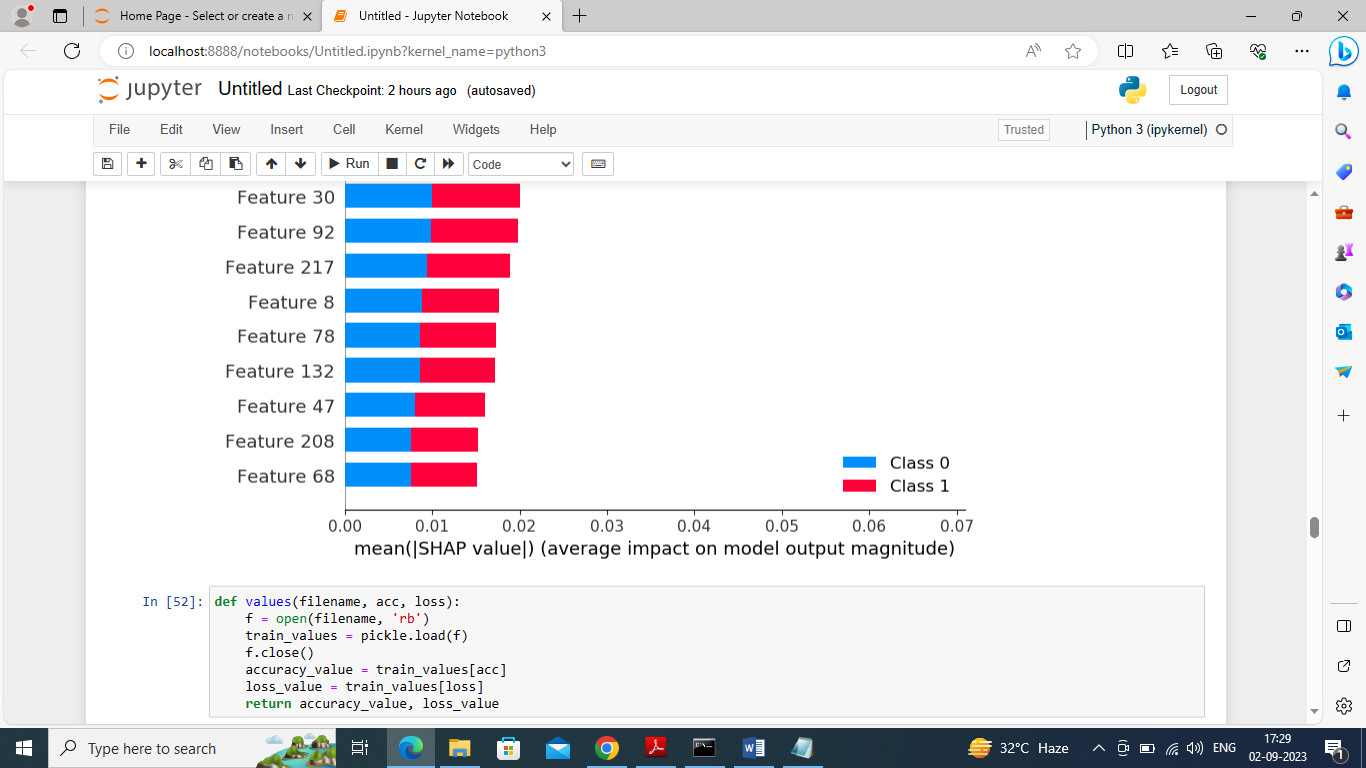


In above screen Modified MobileNetV2 got 99.5% accuracy

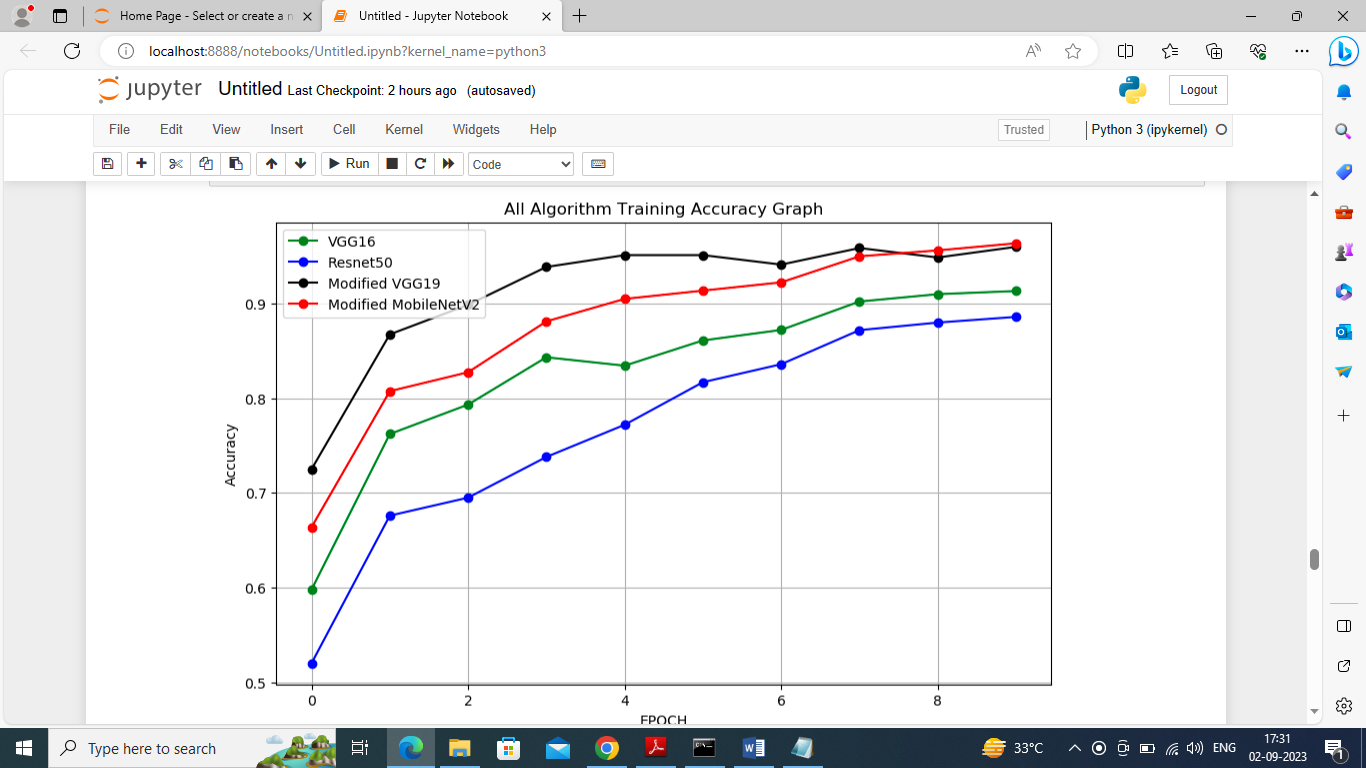


In above screen we are extracting features from MobileNetV2 and then retraining with Random Forest to build extension hybrid model and this model got 100% accuracy

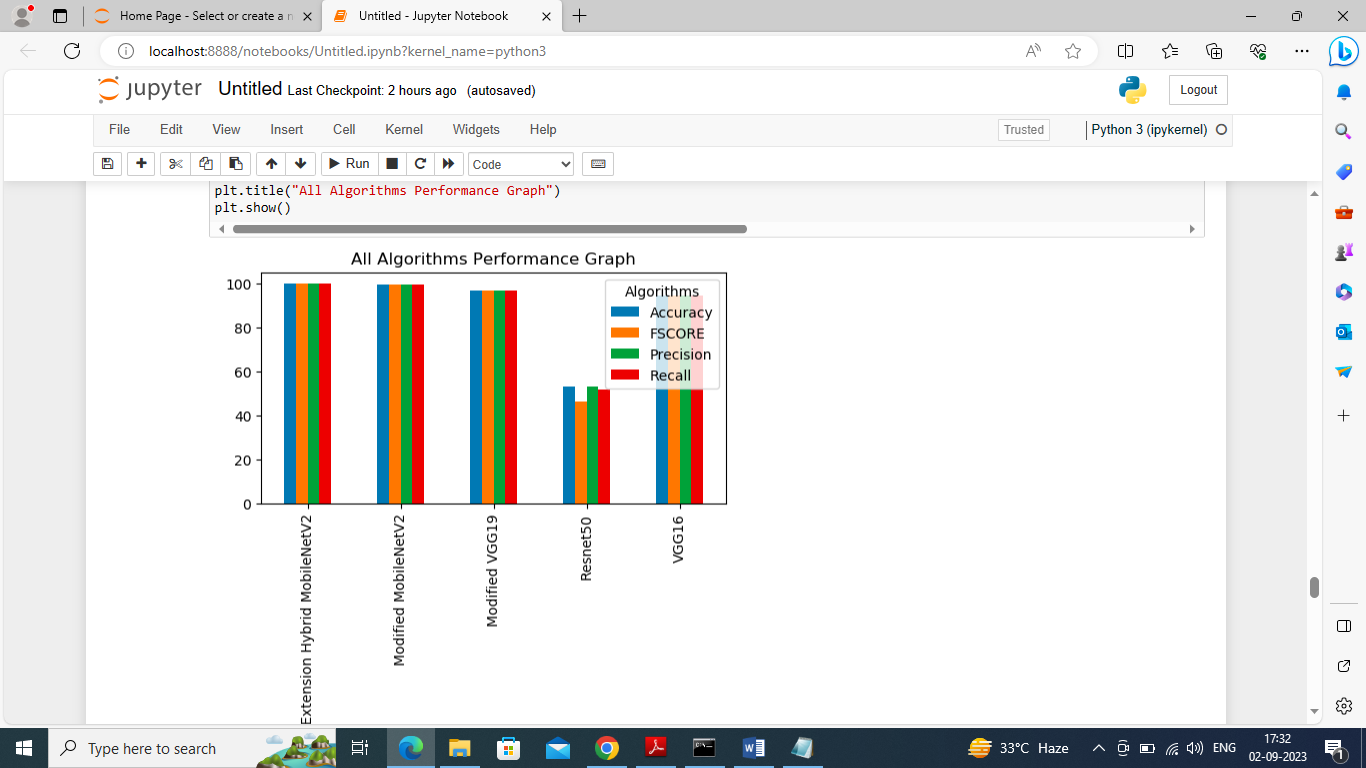




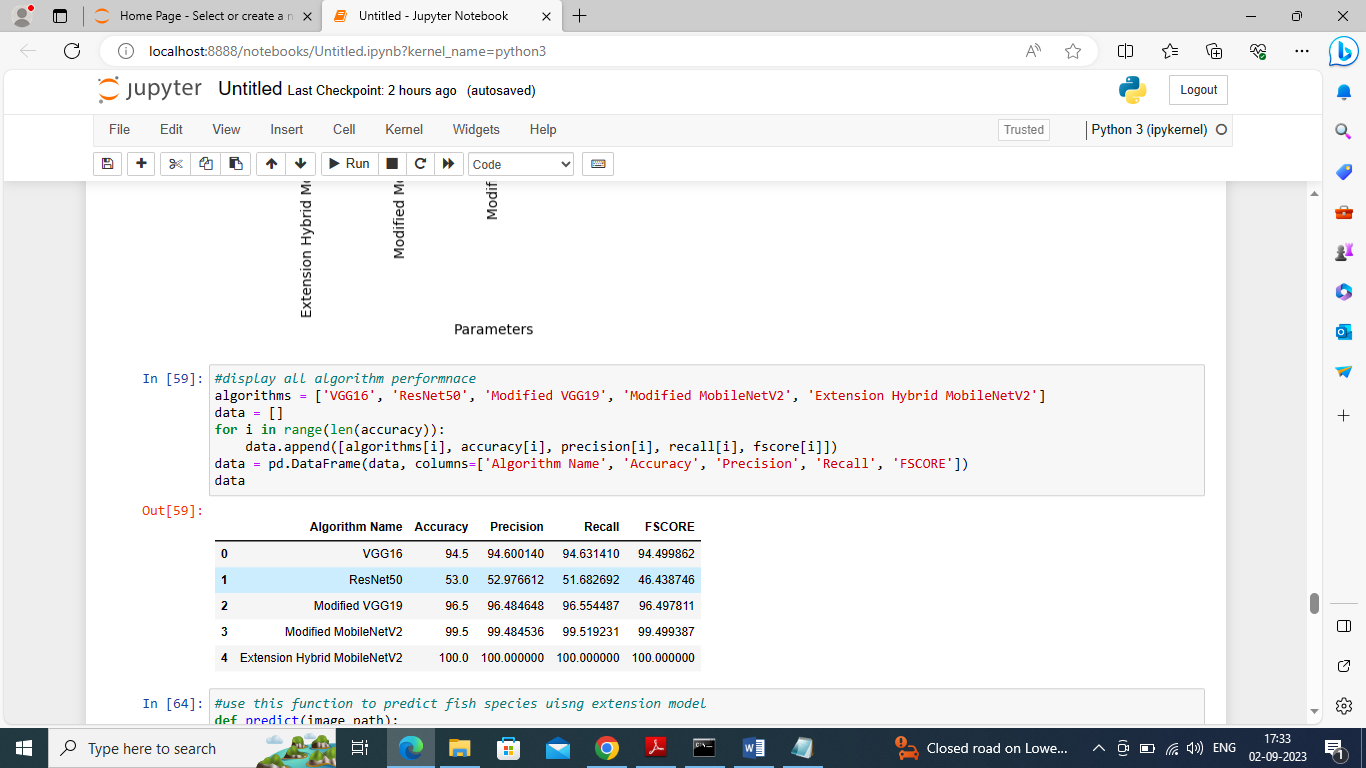
In above two screen using SHAP we are interpreting extension hybrid model and in graph left side we can see image features name and blue bar represents class 0 prediction (monkey pox) and class 1 represents Non-monkey-pox and blue portion represents class 0 and red portion represents class 1. Then with bars we can see how much percent that feature is contributing for that class prediction. The more the bar size increase the highest is the features contribution. So by using this SHAPELY we have interpreted model performance



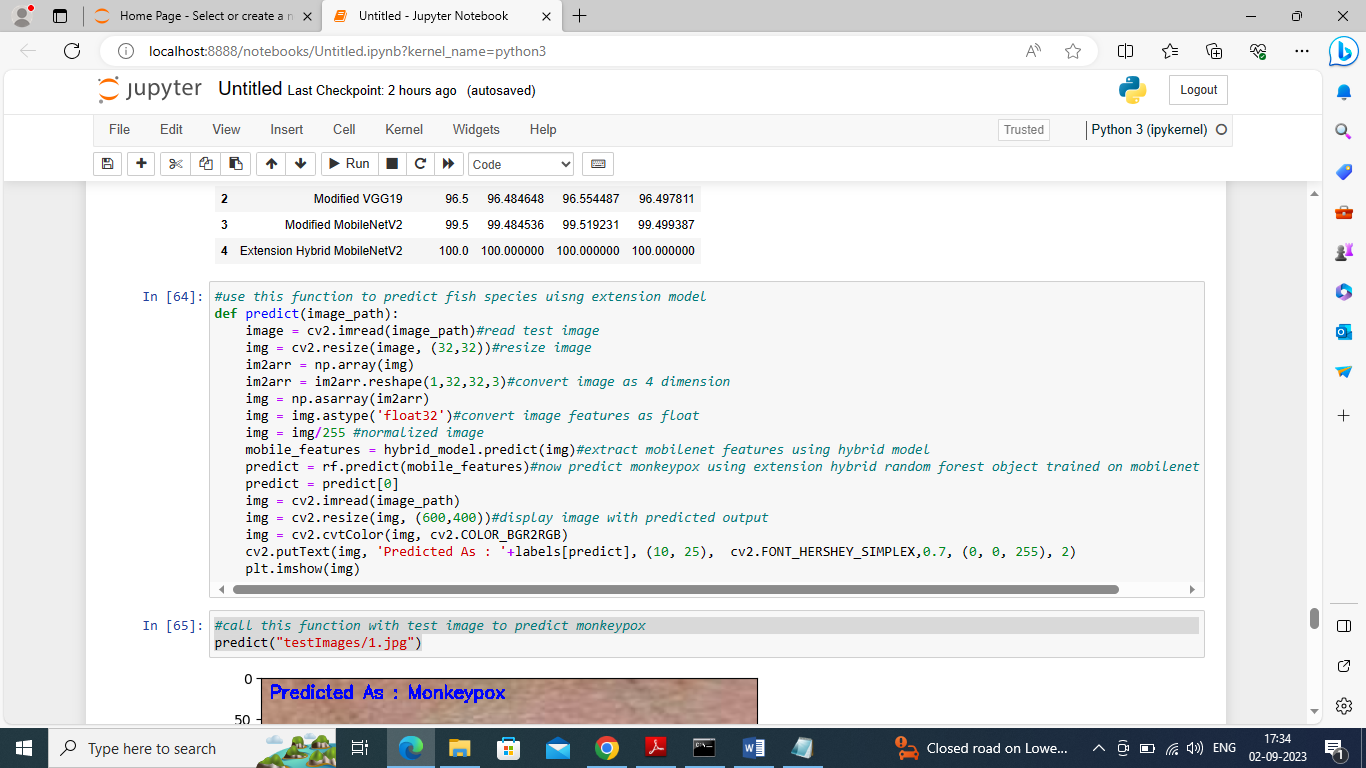
In above graph x-axis represents training epoch and y-axis represents training accuracy and different lines represents different algorithms and in all algorithms Modified VGG19 and Mobilenetv2 perform high.



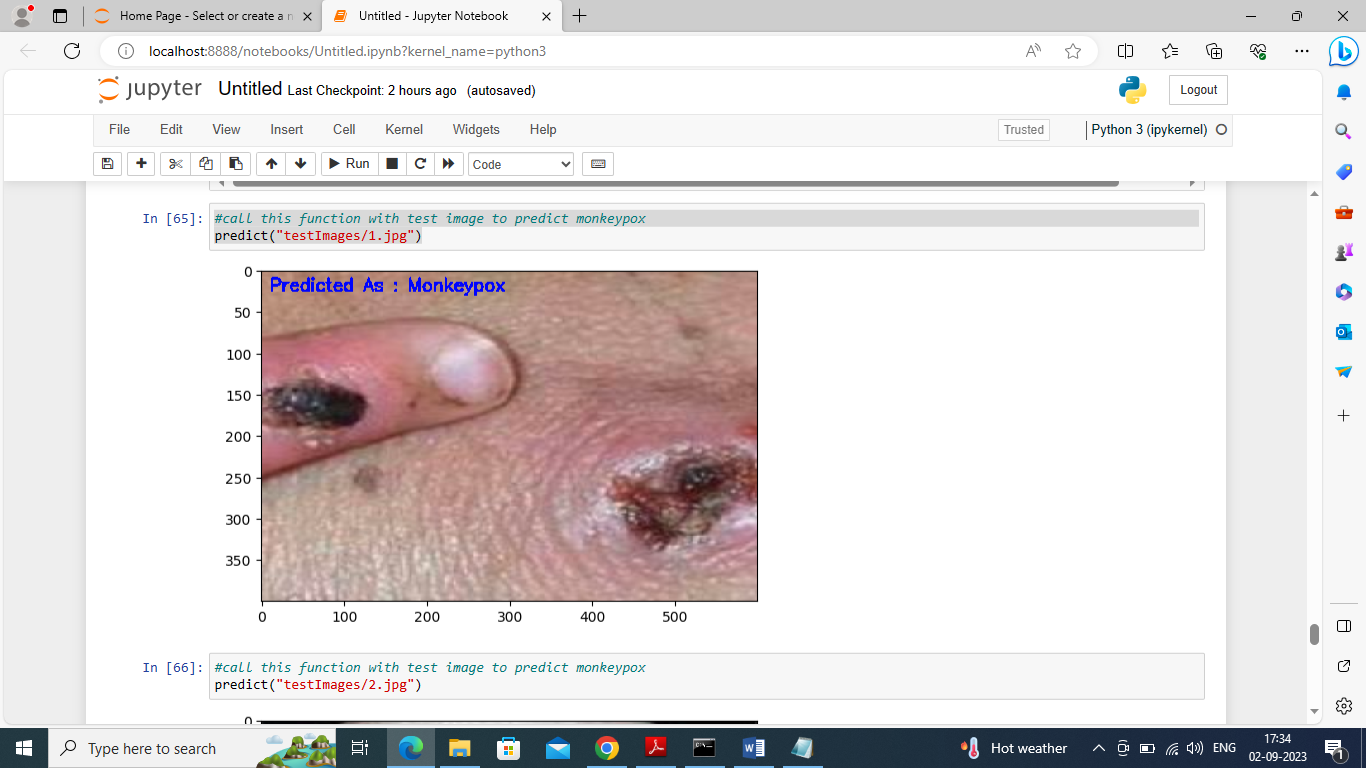
In above graph displaying each algorithm performance where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars



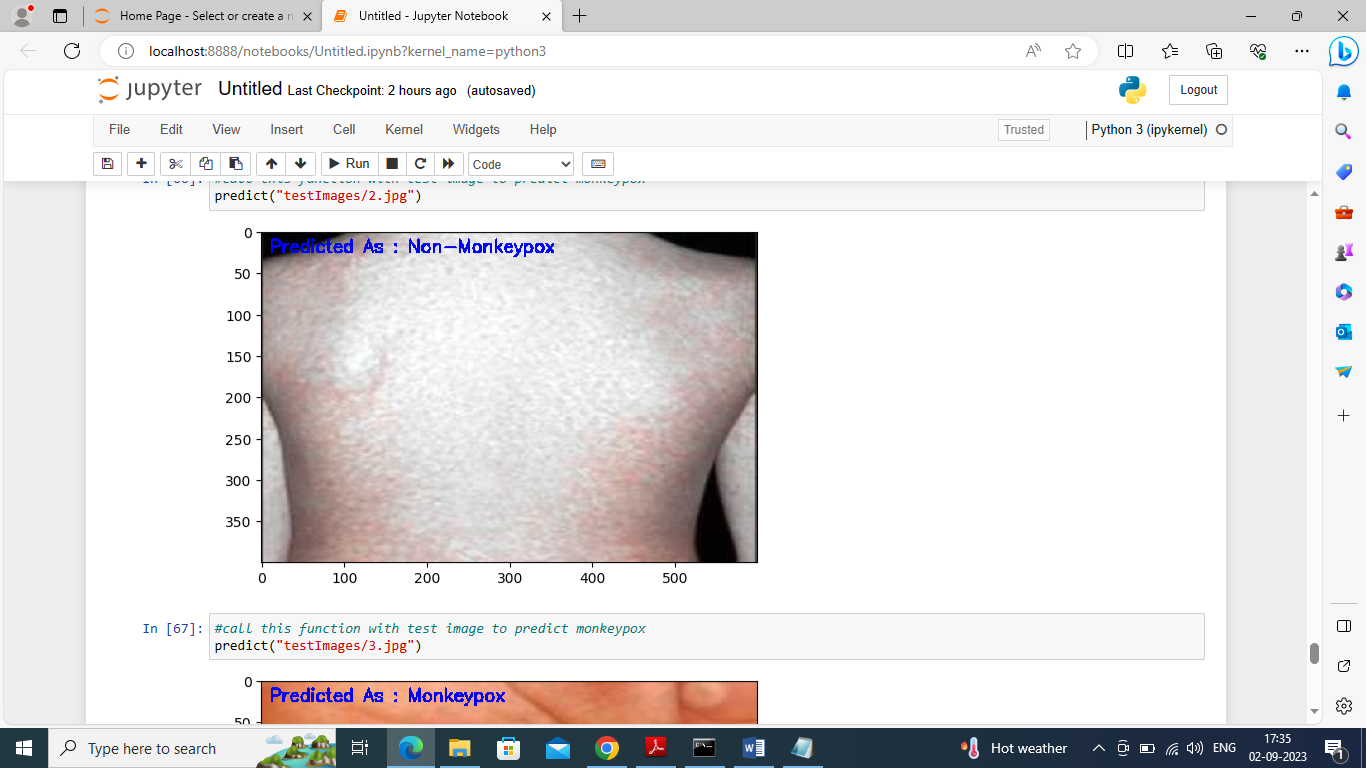
In above screen displaying each algorithm performance in tabular format

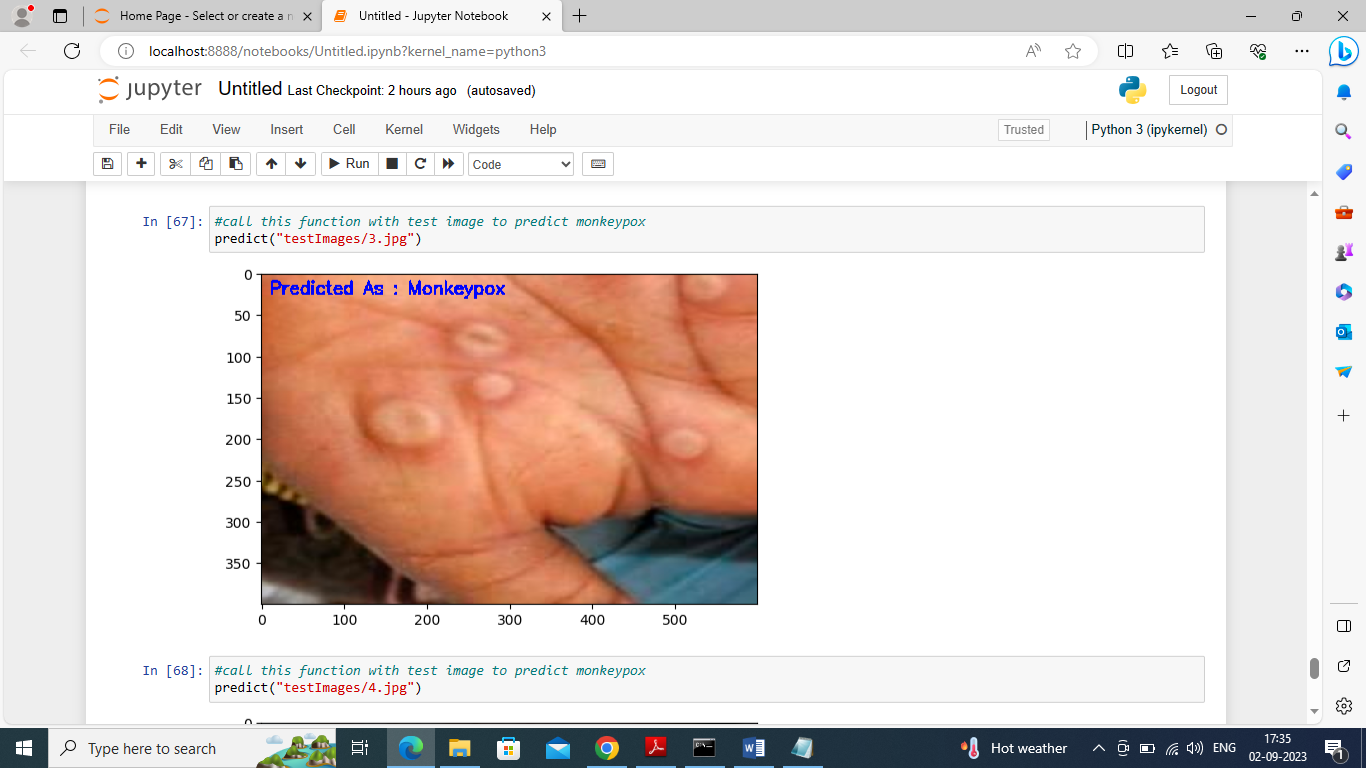


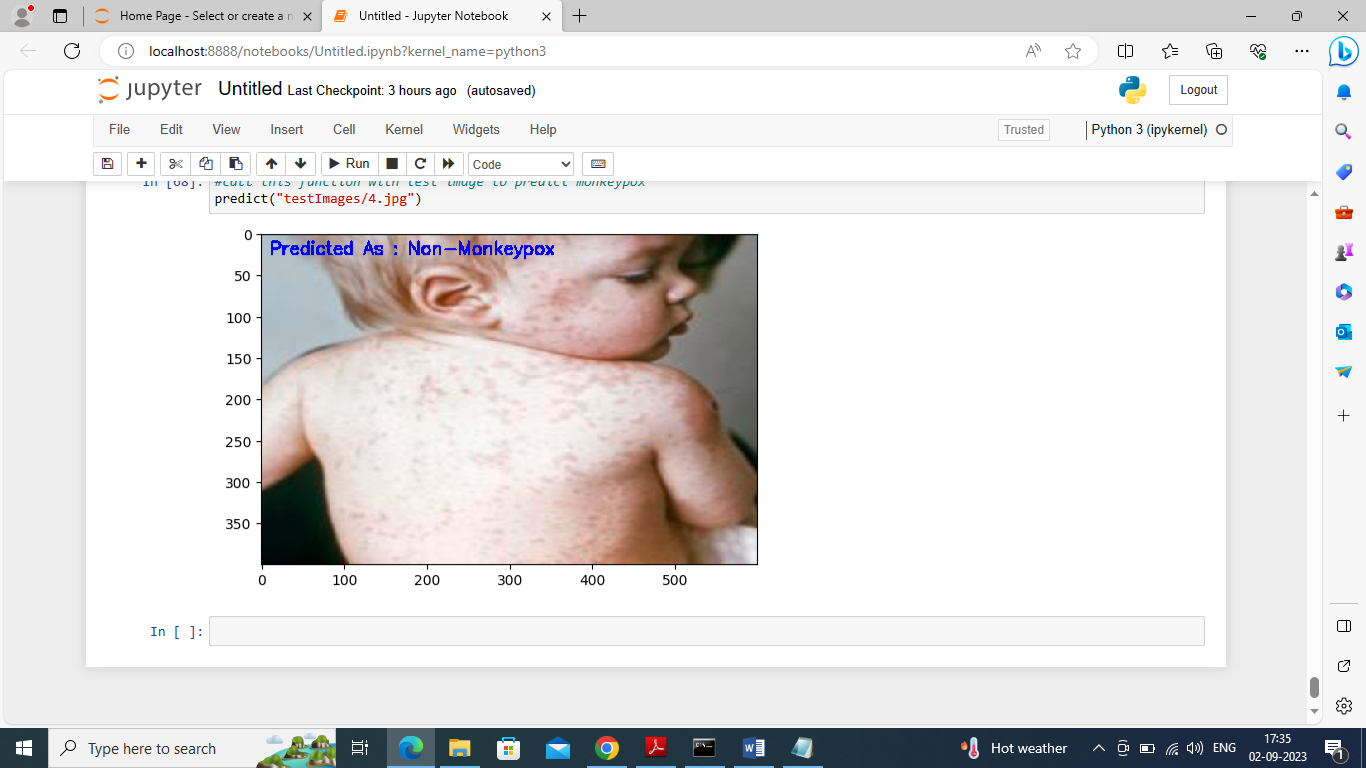
In above screen defining predict function to read input image and then classify image as output



In above image predicted as Money pox







In above screen we can see prediction from different images