CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images

AI algorithms are working everywhere from human medical disease prediction to road side traffic monitoring. AI also utilizing in generating high quality fake synthetic images which is impossible for human eyes to differentiate as Real or Fake images.

In propose paper author employing CNN2D algorithm to distinguish between Fake and Real images. To train CNN2D author has generated fake synthetic images from CIFAR10 dataset and then utilize CIFAR Real and fake Generated images.

Author has used explainable GRAD-CAM (Gradient Class Activation Mapping) algorithm to identify features which helps CNN in identifying or classifying fake or real images. A Grad-CAM image is a heat map that highlights the regions of an image that a convolutional neural network (CNN) focuses on when making a prediction. Grad-CAM stands for gradient-weighted class activation mapping, and it's a visualization technique that helps explain how a CNN makes decisions.

In propose paper author has used 2 layers or CNN2D, 2 layers of Maxpooling2D and 2 layers of dense and then experiment with different number of neurons such as 32, 64, 128, 4096 and among all neurons 32 is giving best accuracy so we have used same number of 32 neurons for implementation.

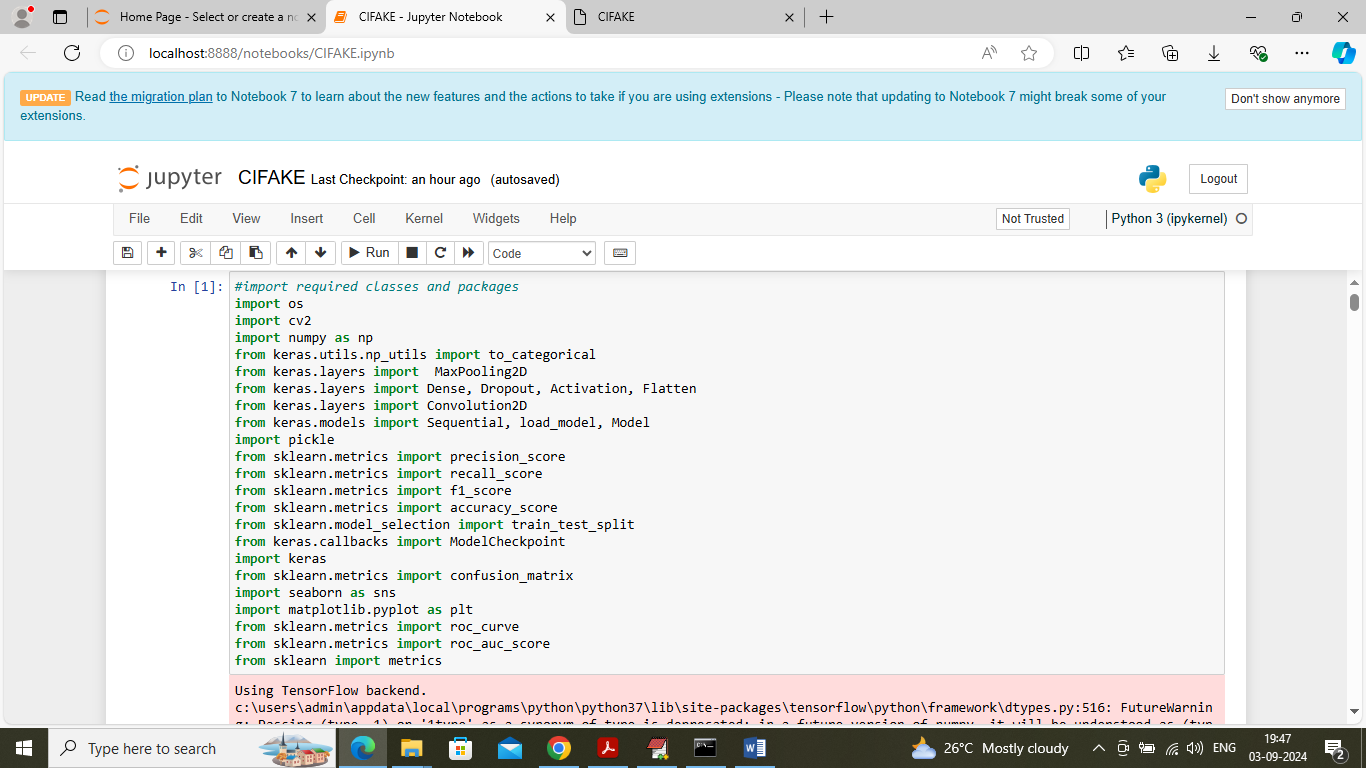
Algorithm performance is evaluated in terms of accuracy, precision, recall, Confusion Matrix and FSCORE. Propose algorithm giving more than 93% accuracy.

Extension Concept:

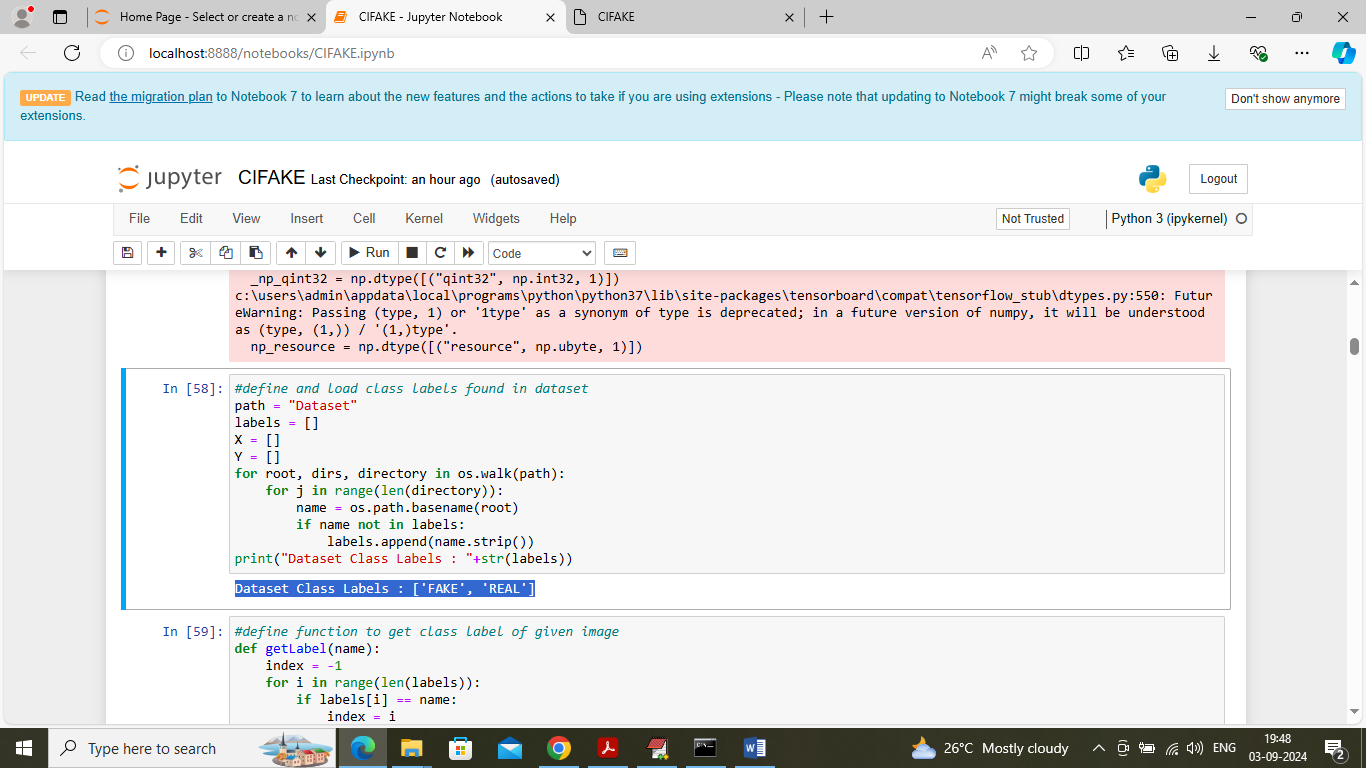
In propose CNNN2D author has not used any additional layers such as Global Average Pooling or Dropout layer which will help CNN in gathering more optimizing features and then drop all those features which has least importance features. Dropping least importance features help CNN2D to concentrate on more optimize features which help in enhancing accuracy.

SCREEN SHOTS

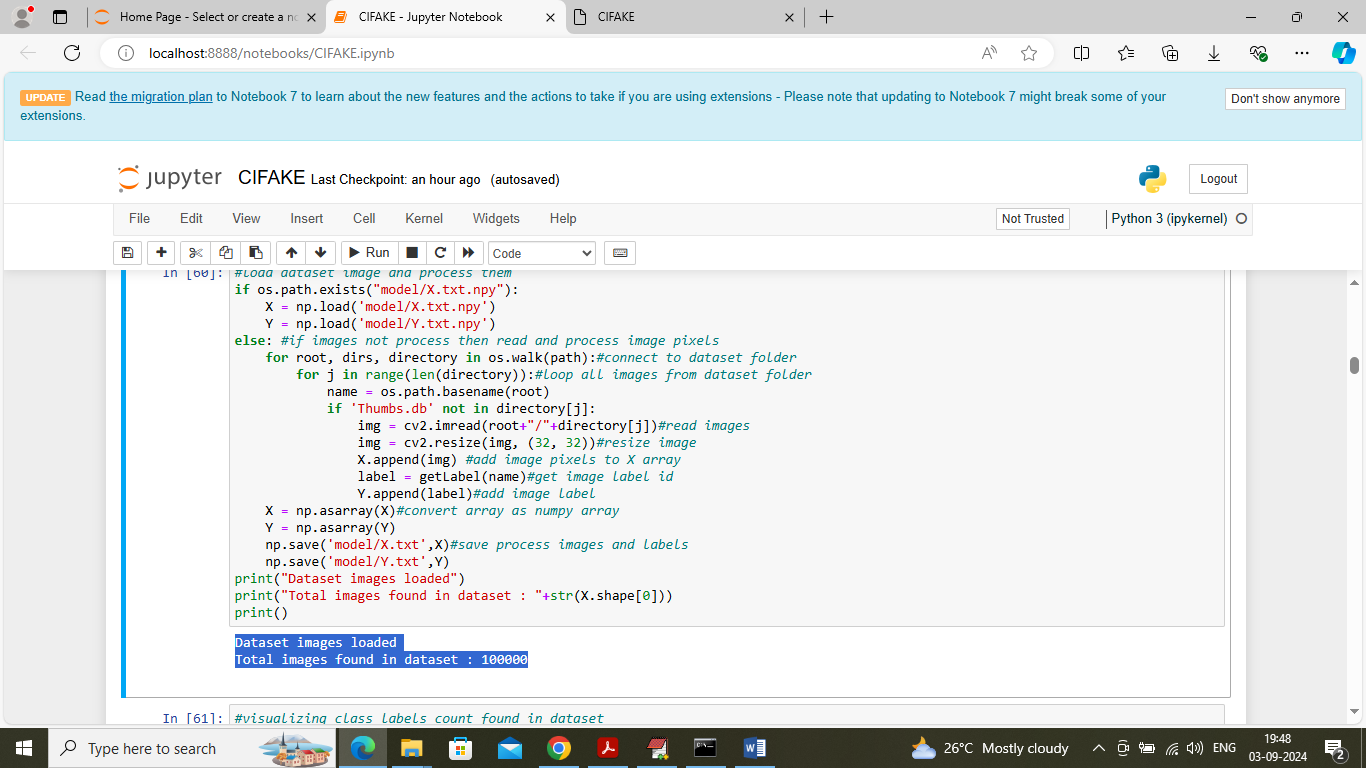
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



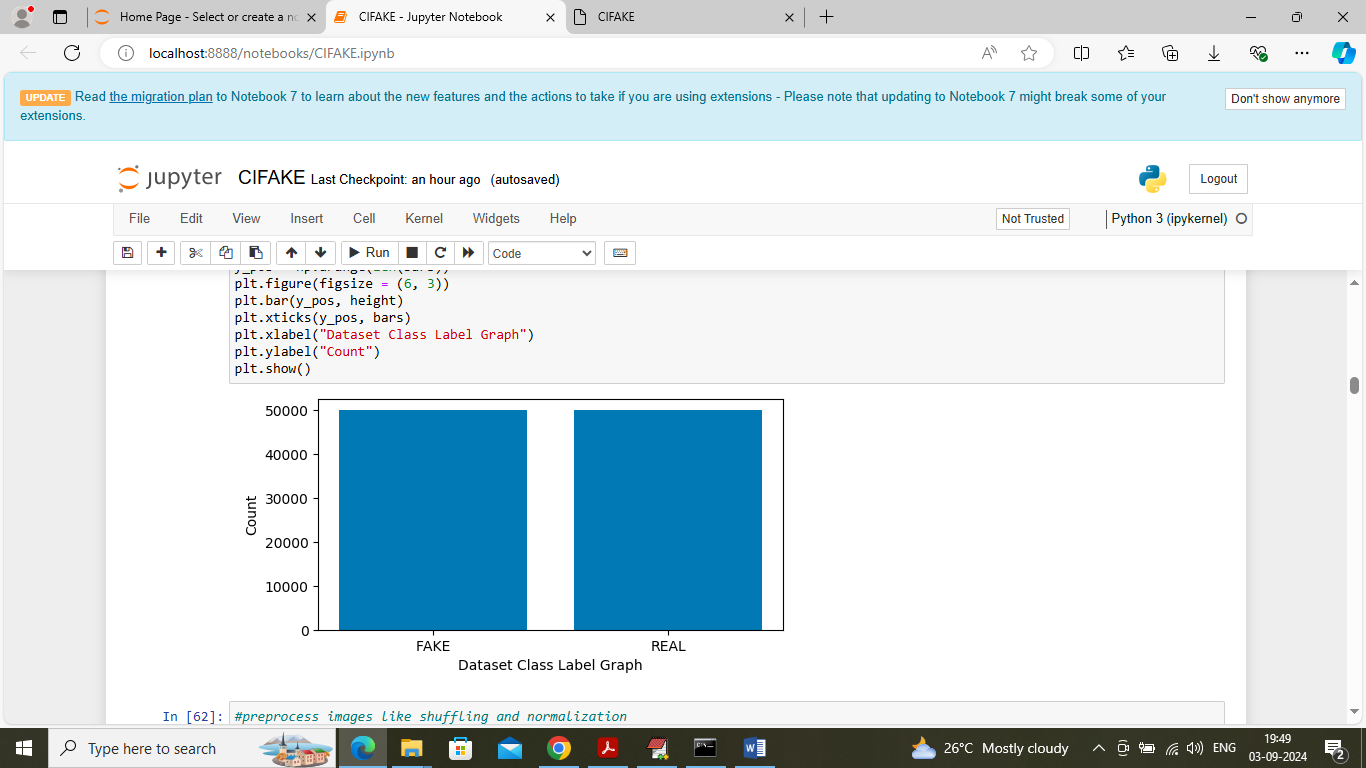
In above screen importing required python classes and packages



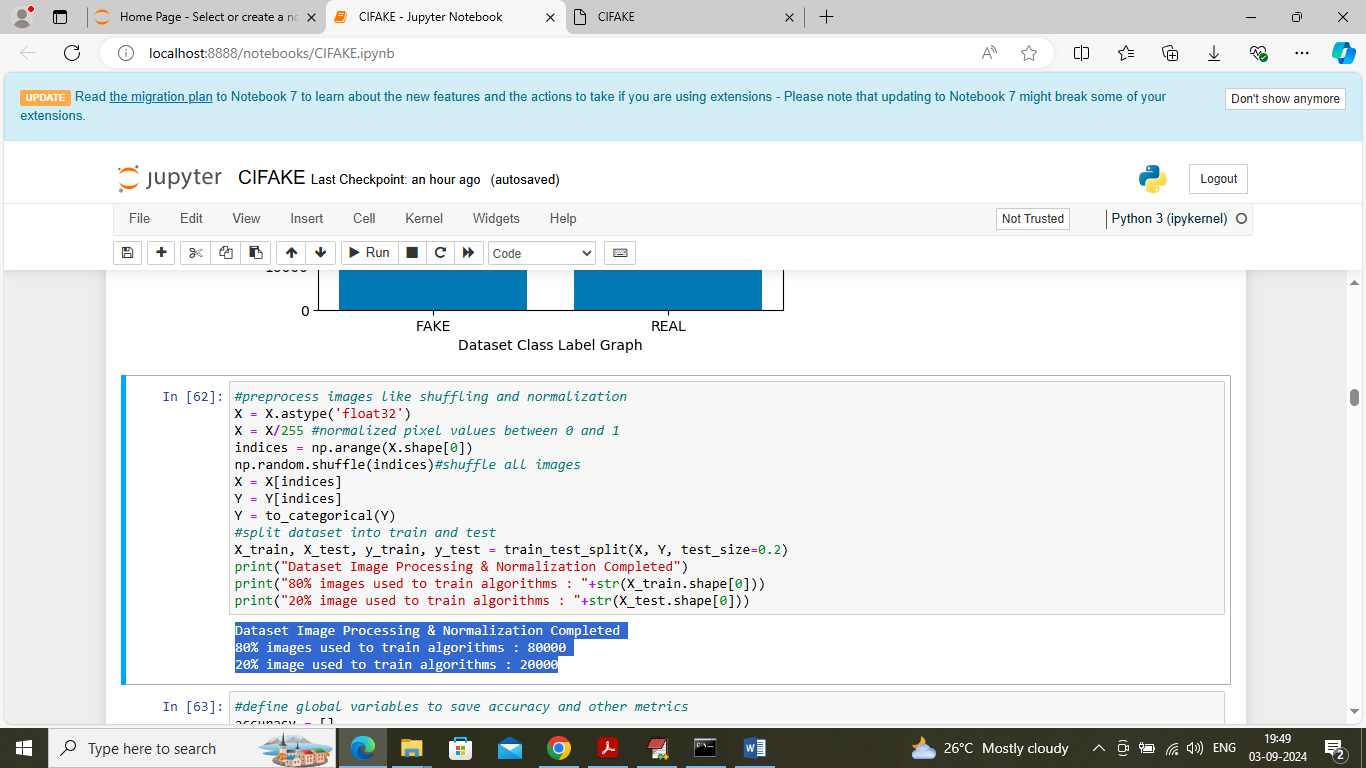
In above screen loading dataset to obtained different class labels found in dataset



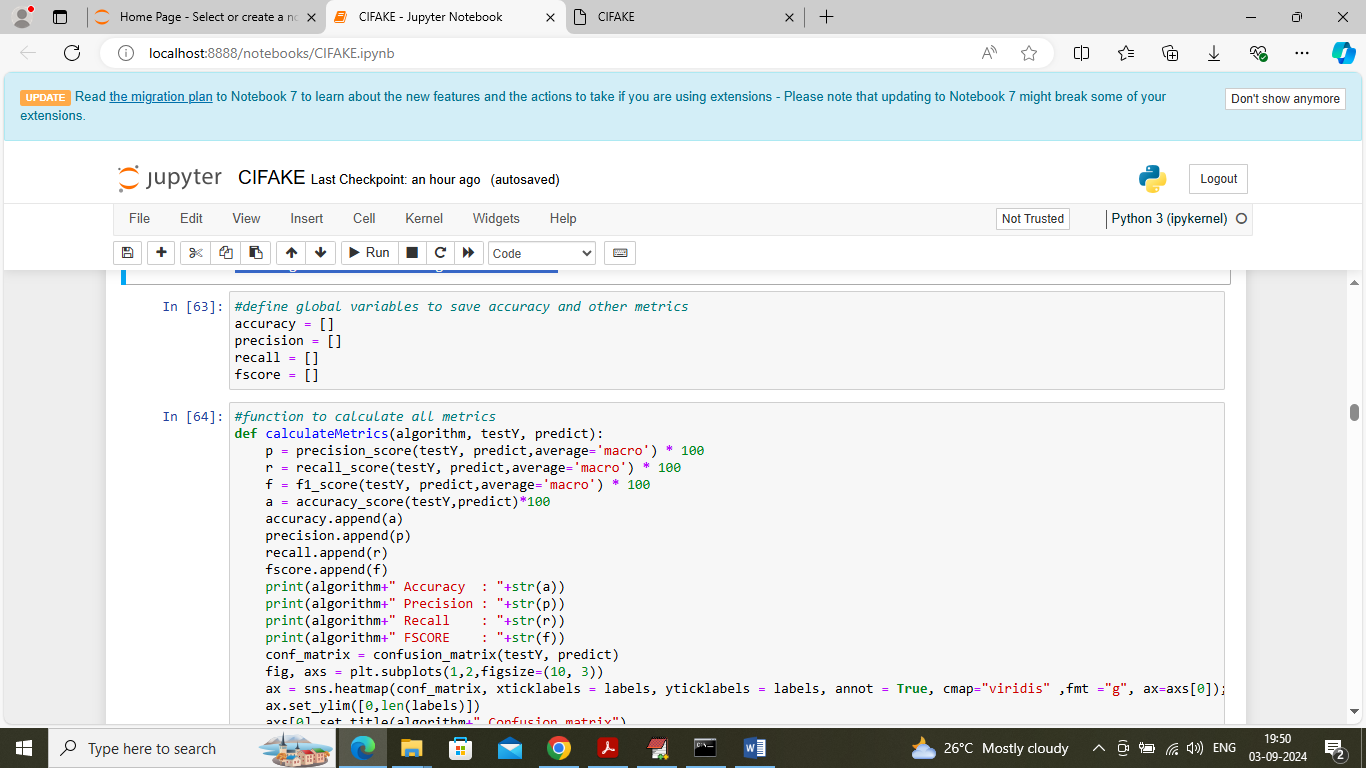
In above screen looping and reading all images from dataset and then resizing all images to equal sizes and then displaying dataset contains 1 lac images



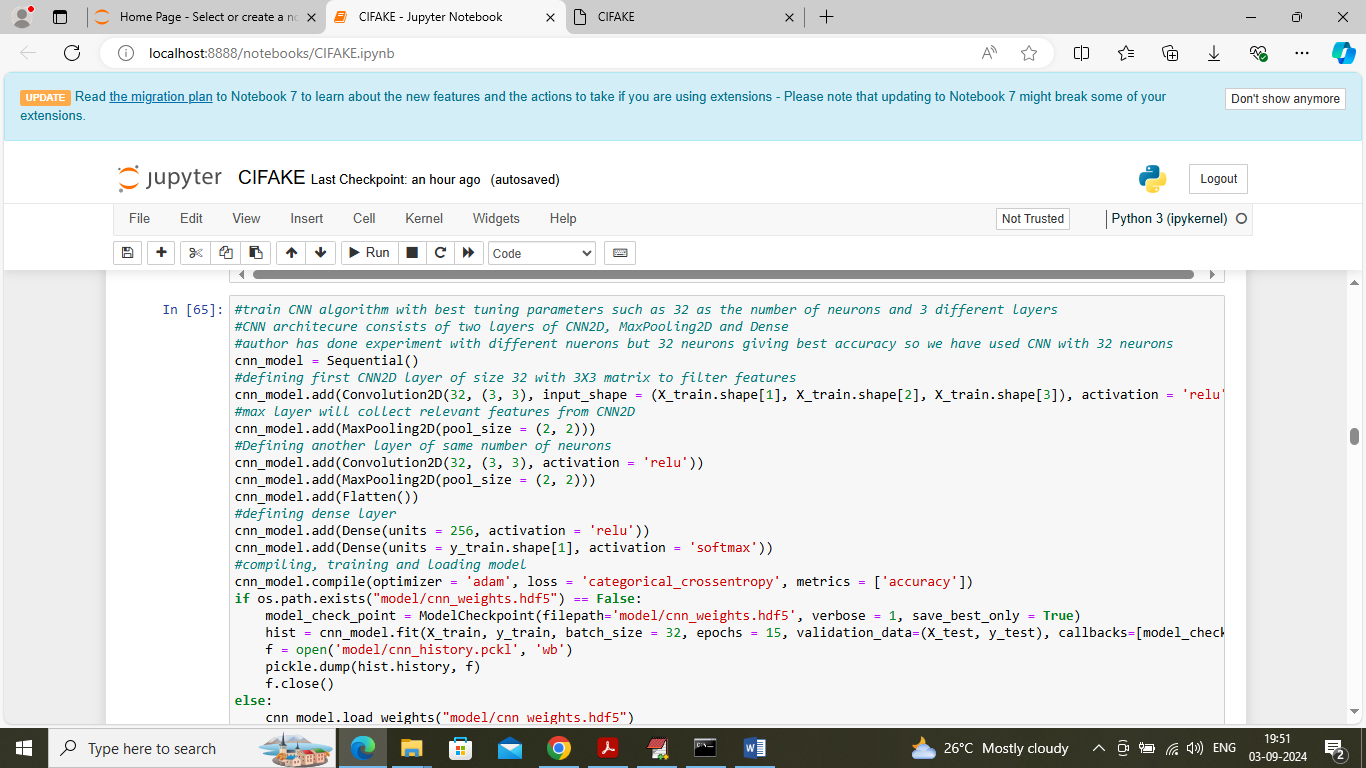
In above graph visualizing number of Fake and Real images exists in dataset



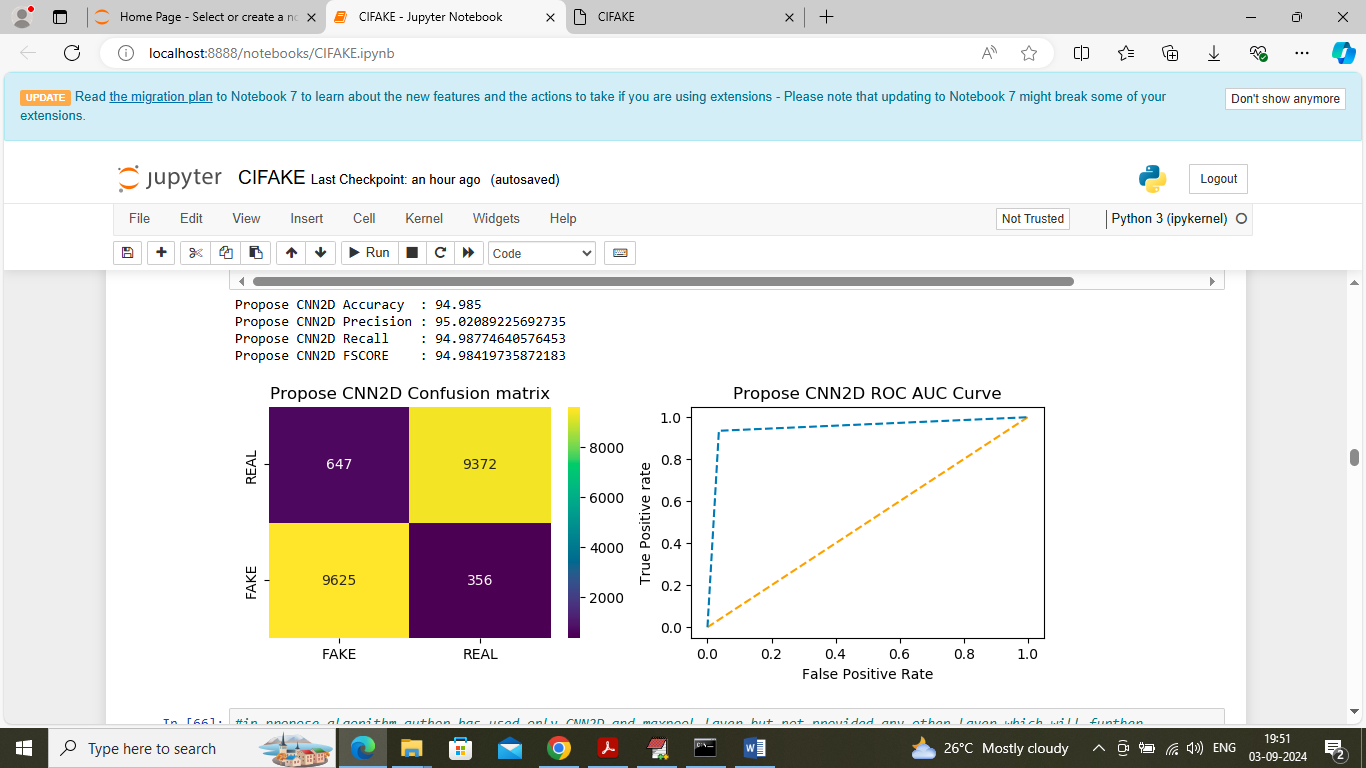
In above screen applying image processing techniques such as shuffling, normalizing and then splitting to train and test where 80% dataset images for training and 20% for testing



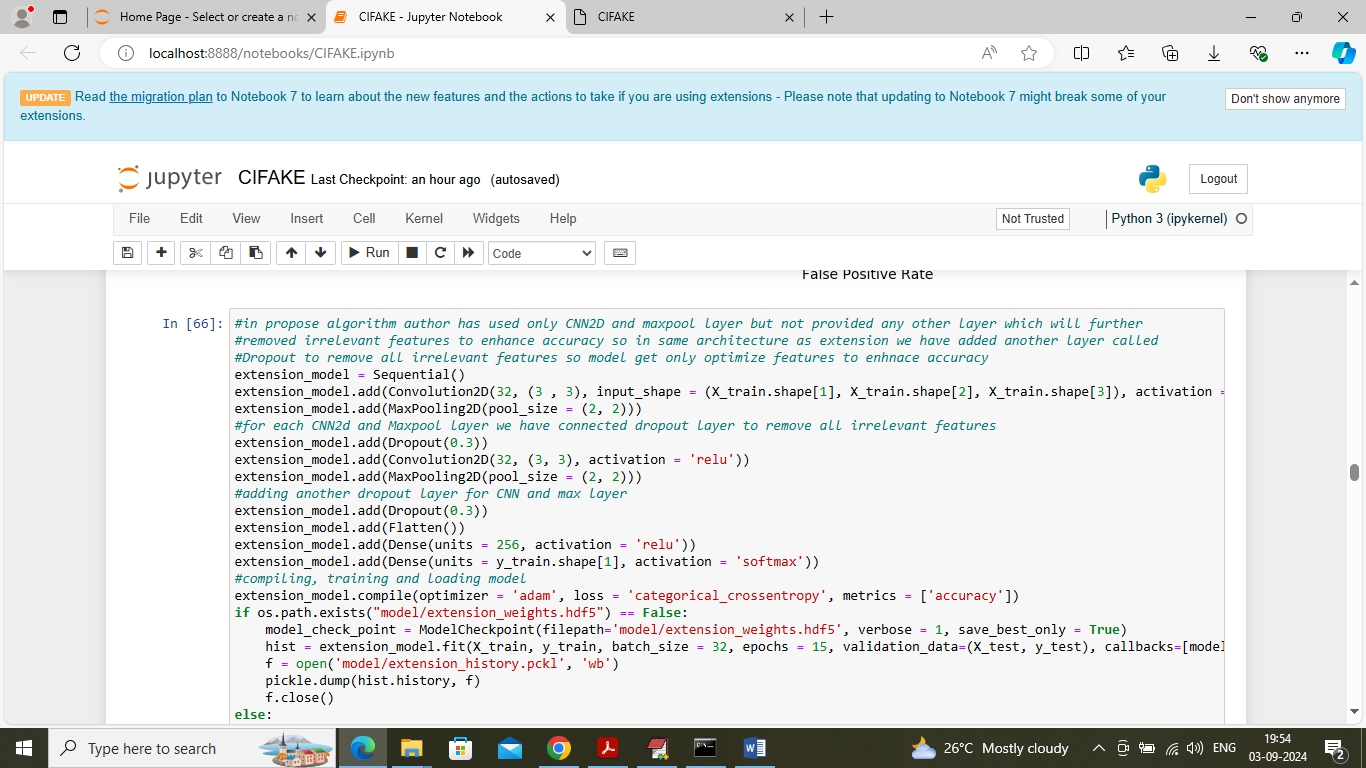
In above screen defining function to calculate accuracy and other metrics



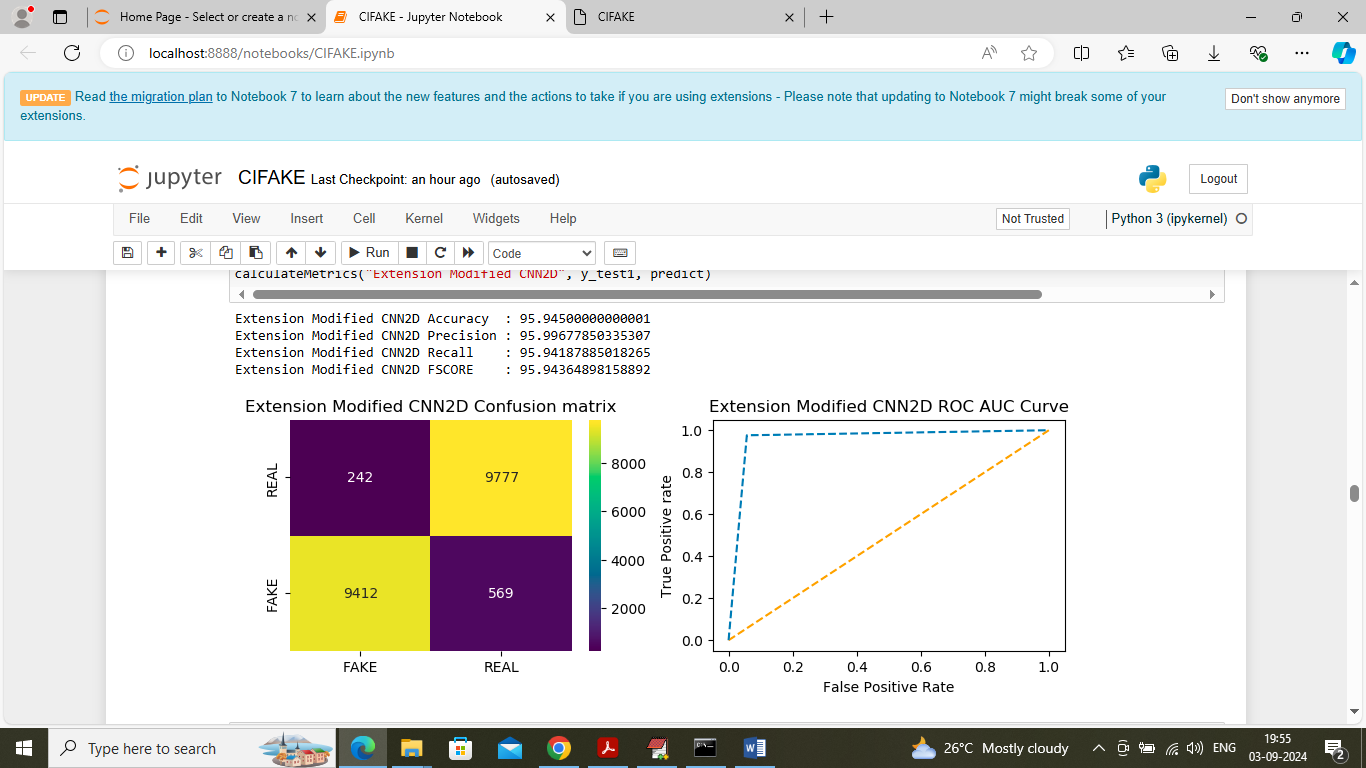
In above screen defining propose CNN2D algorithm and after executing above block will get below output



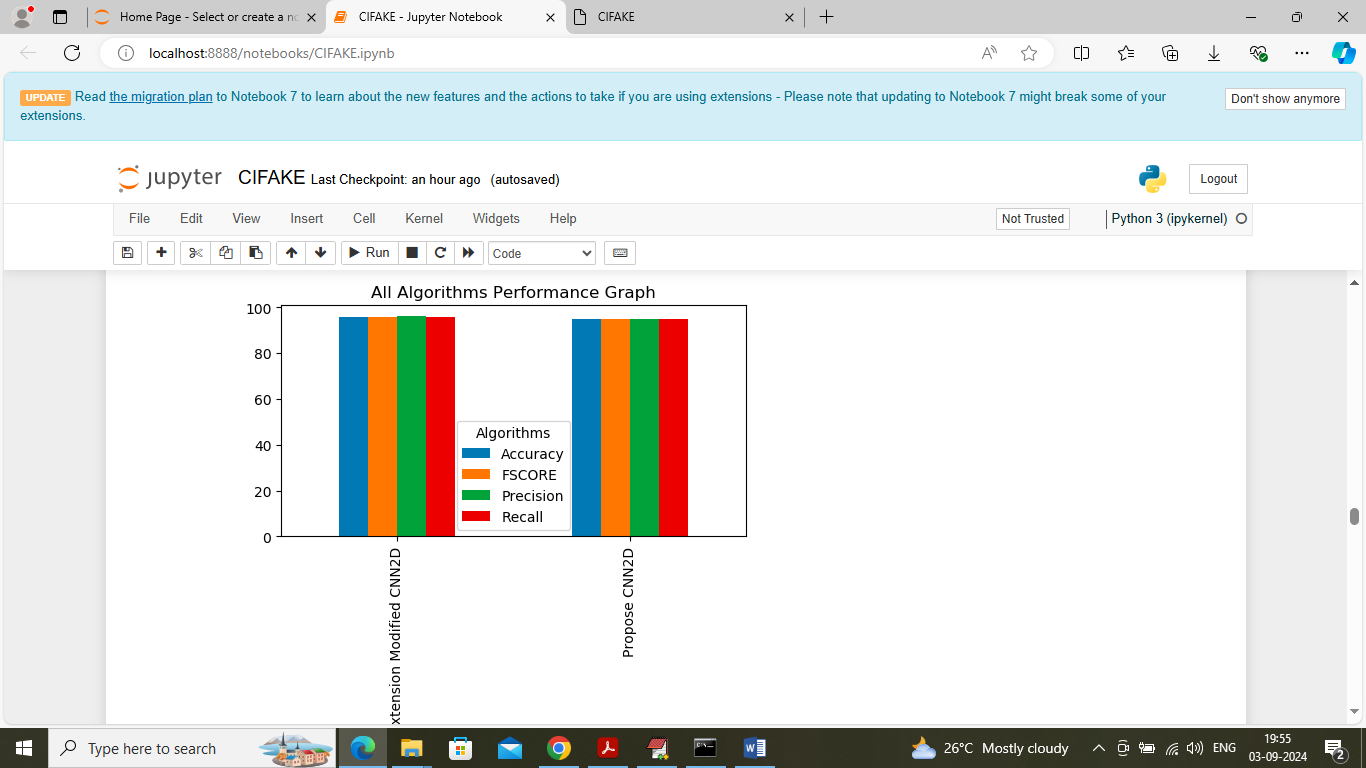
In above screen propose CNN2D with 32 neurons got 94% accuracy and can see other metrics like precision, recall and FSCORE. In above confusion matrix graph x-axis represents ‘Predicted Labels’ and y-axis represents ‘True Labels’ and then all yellow boxes represents correct prediction count and blue boxes represents incorrect prediction count which are very few. In above ROC graph x-axis represents ‘False Positive rate’ and y-axis represents True Positive rate and if blue line comes on top of orange line then all predictions are correct and if goes down below orange line then predictions are incorrect.



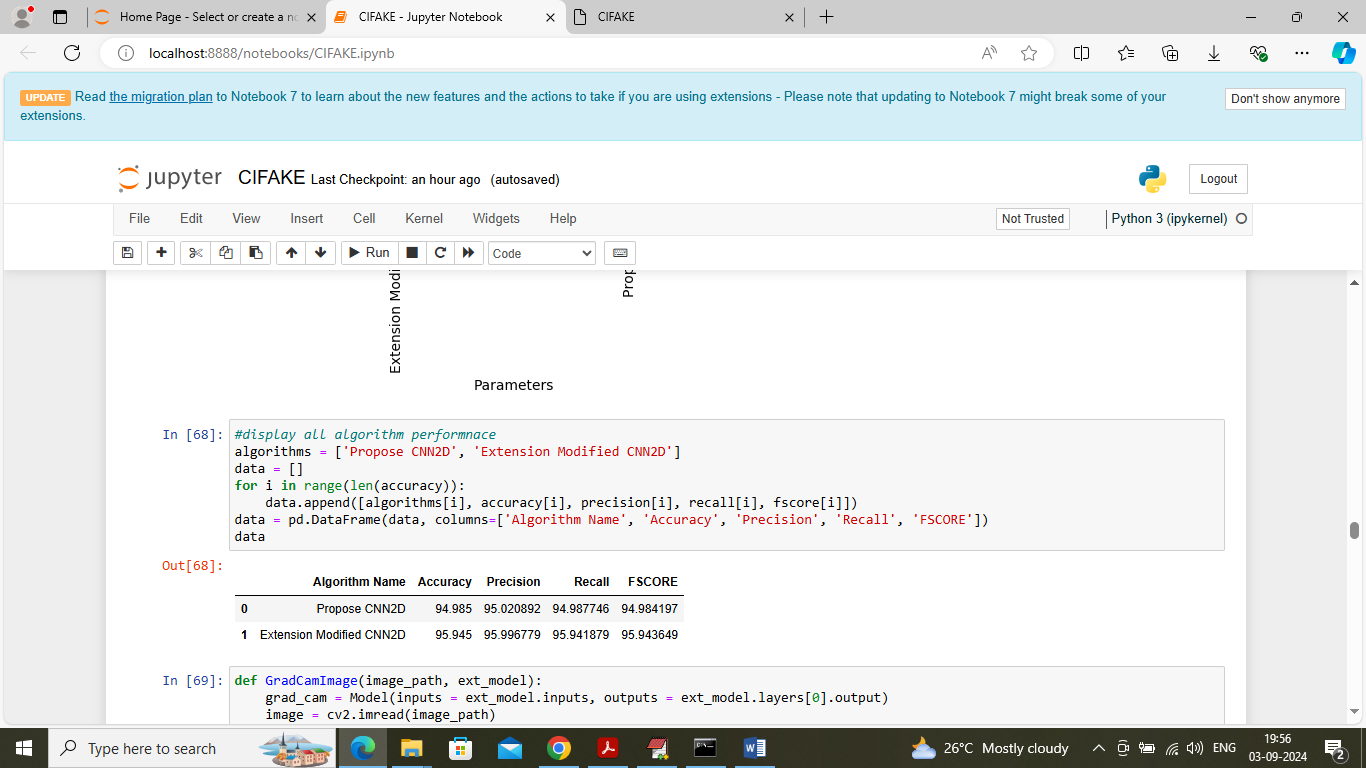
In above screen defining extension model by adding new layers to propose CNN2D model and after modifying propose architecture model will get below output



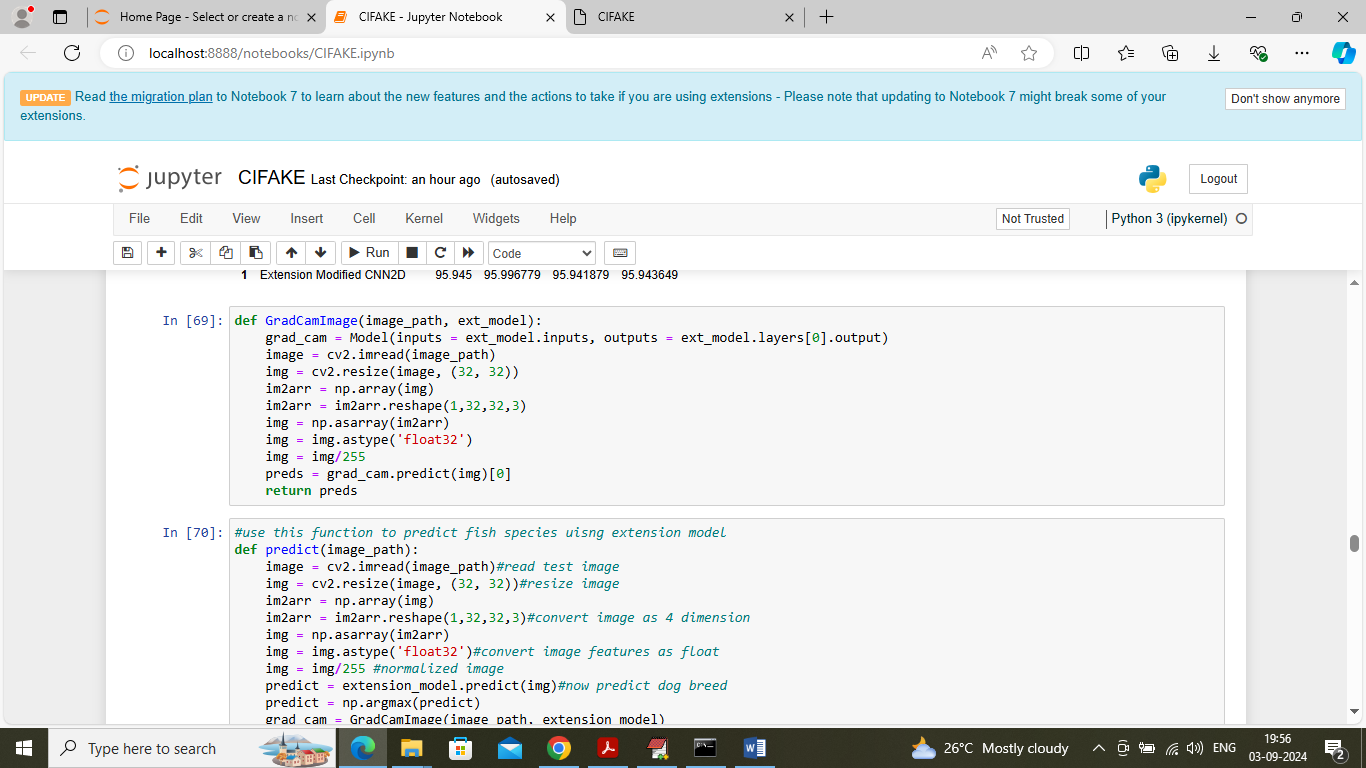
In above screen extension algorithm got 95% accuracy and can see other metrics also



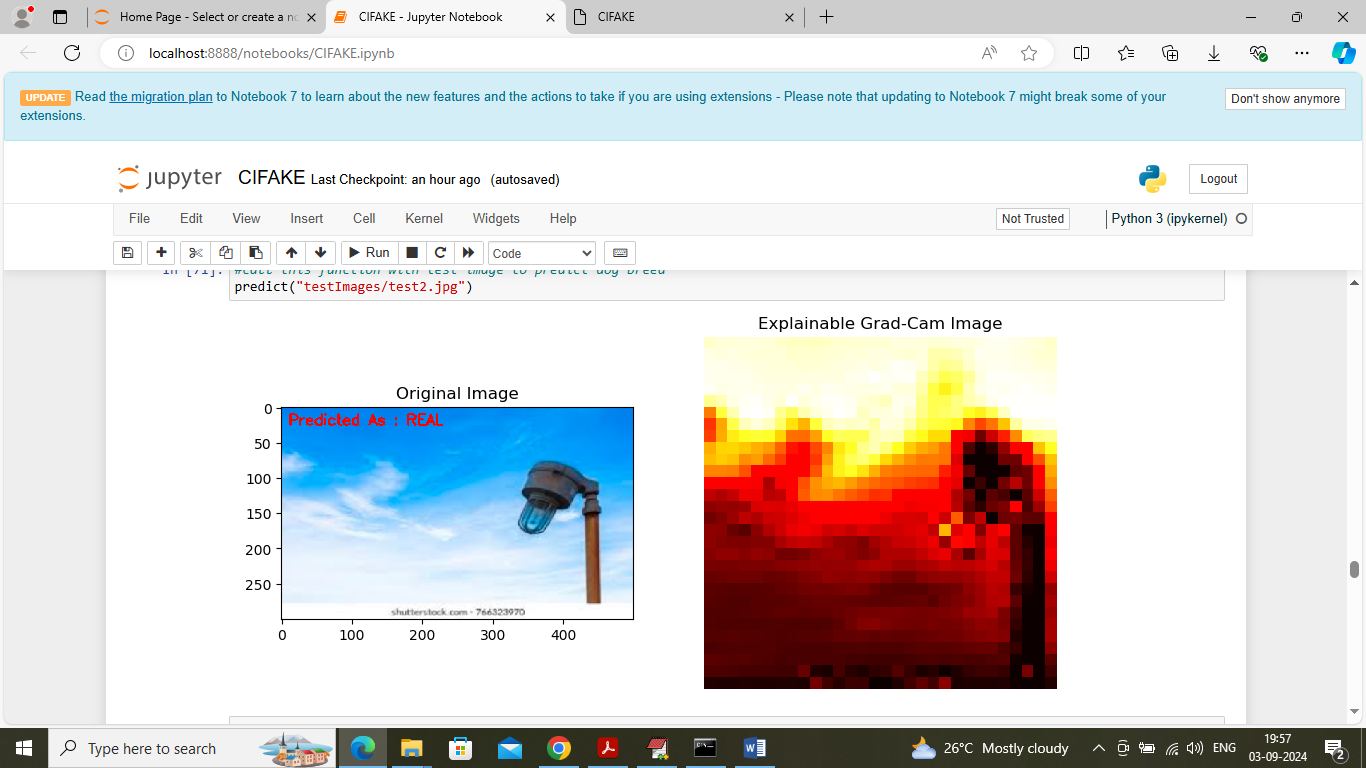
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in both algorithms extension got high accuracy



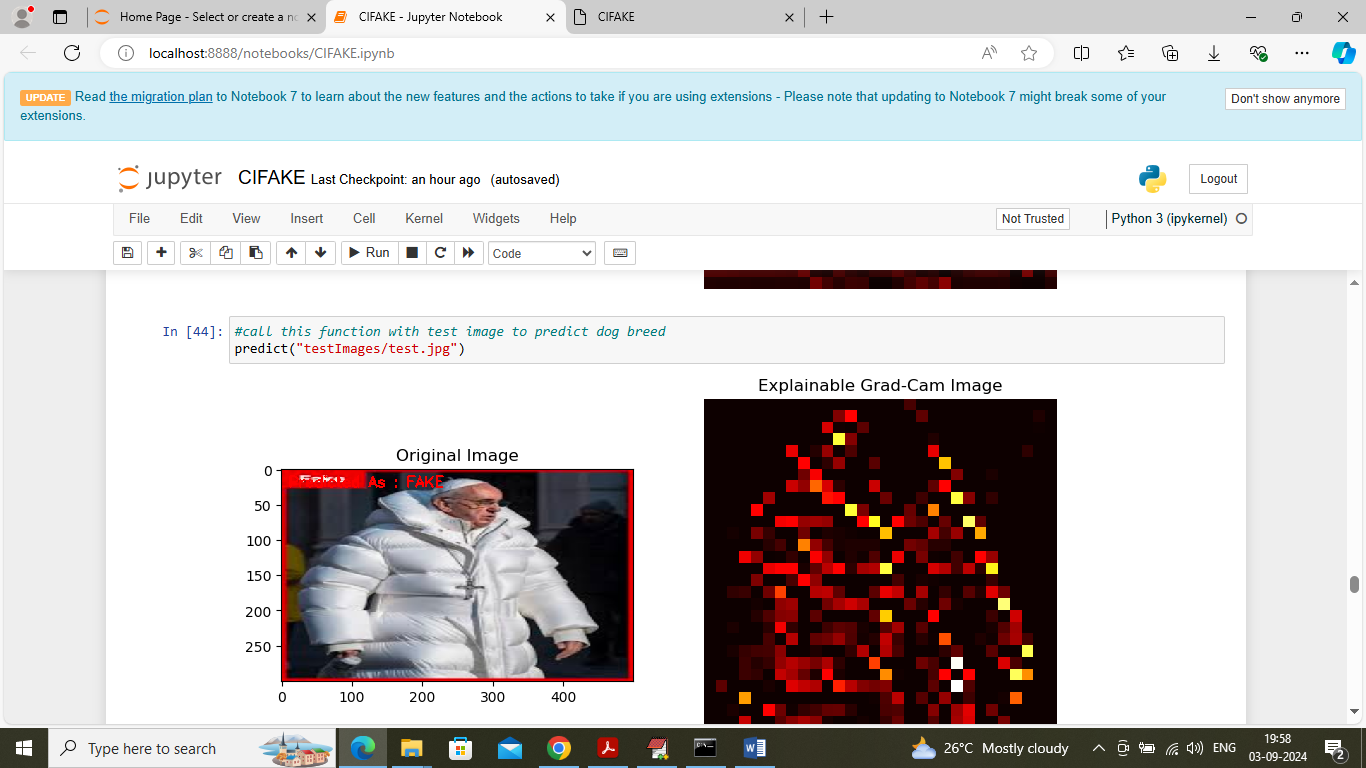
In above screen displaying both algorithm performance tabular format



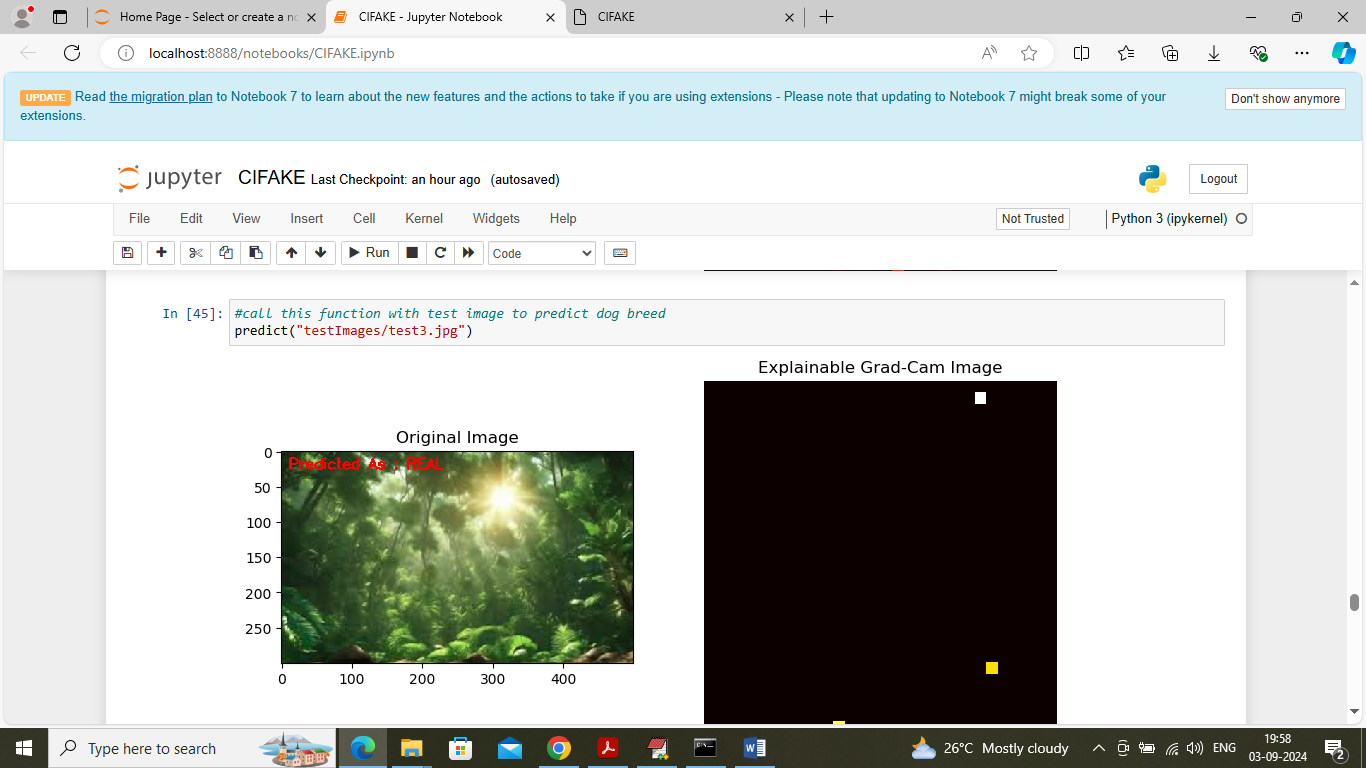
In above screen defining predict function to classify image as Fake or Real and then defining function to get GRAD-CAM explainable image



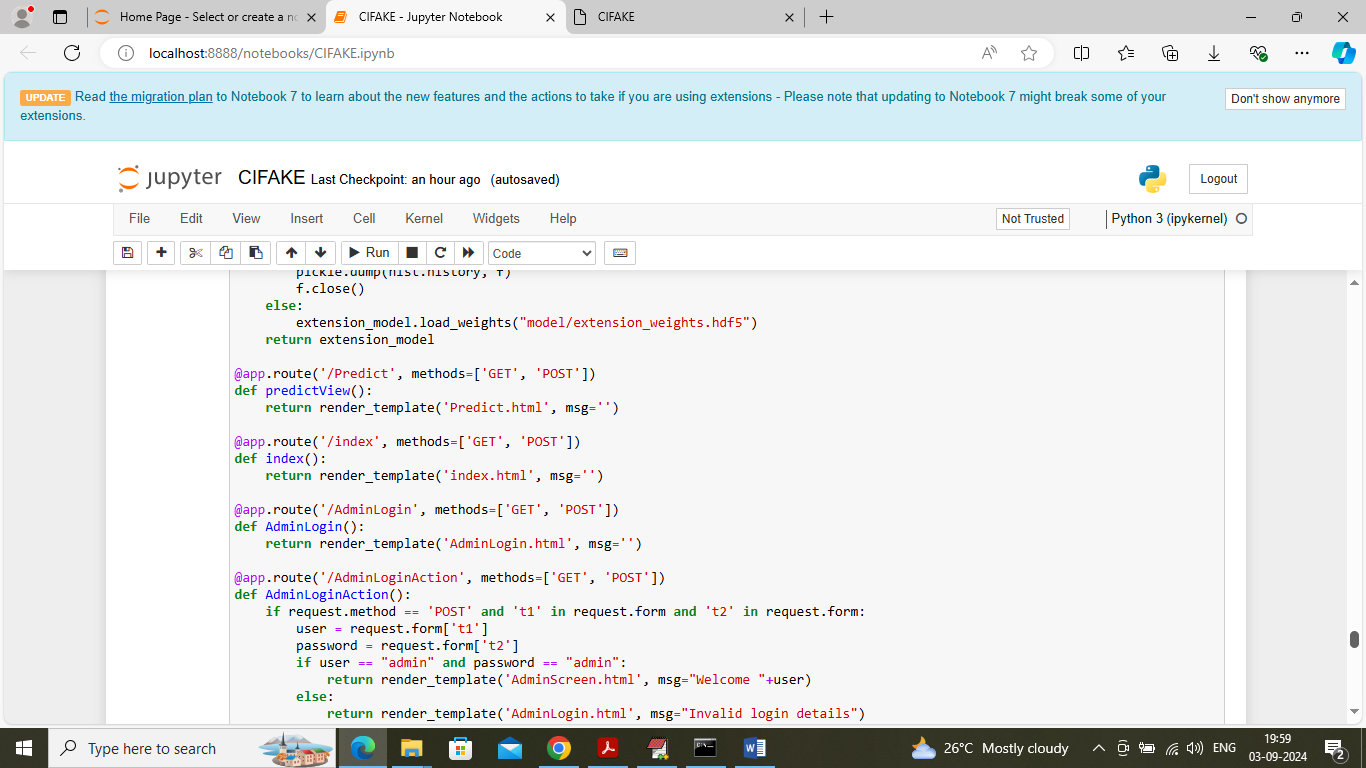
In above screen in red colour text can see given image classify as ‘Real’ and then second image is the explainable GRAD CAM image and the image features which are dark in colour help CNN to classify image as real



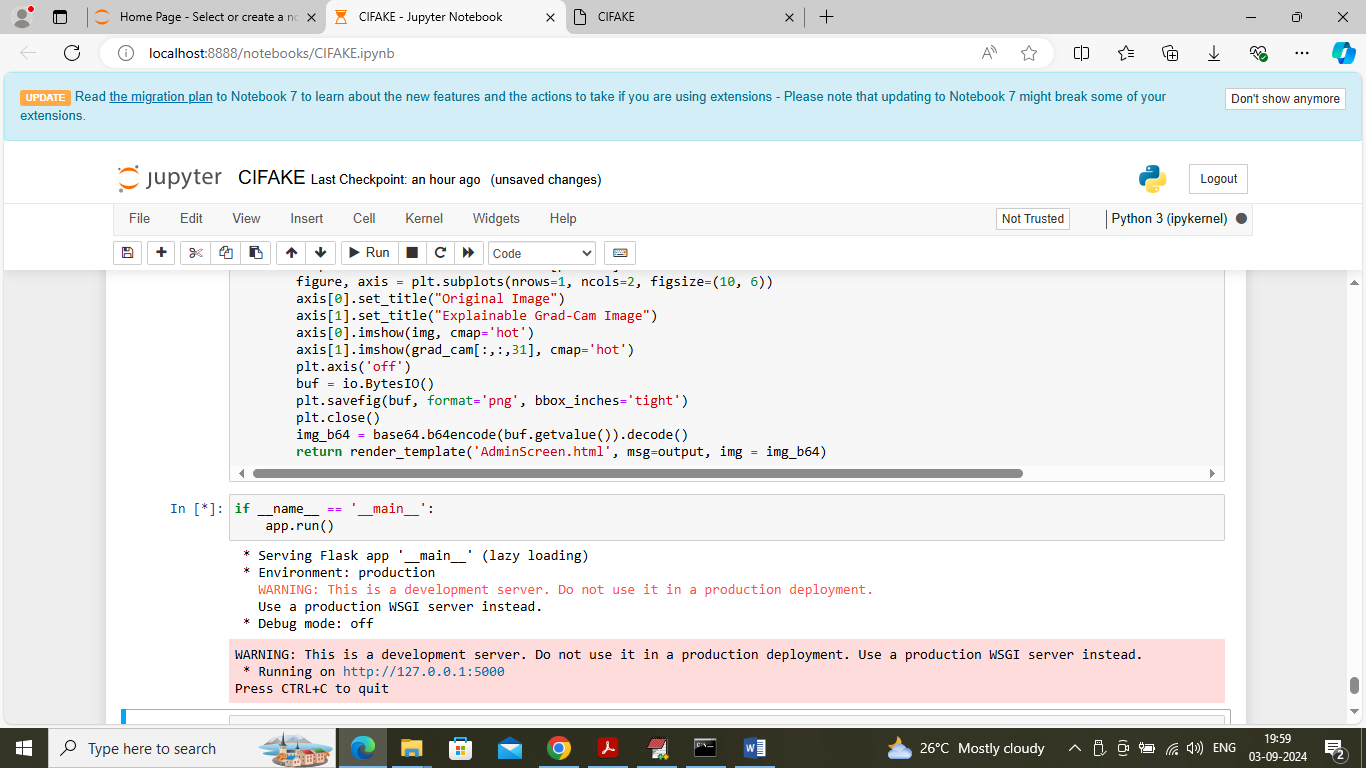
Above image is classify as Fake



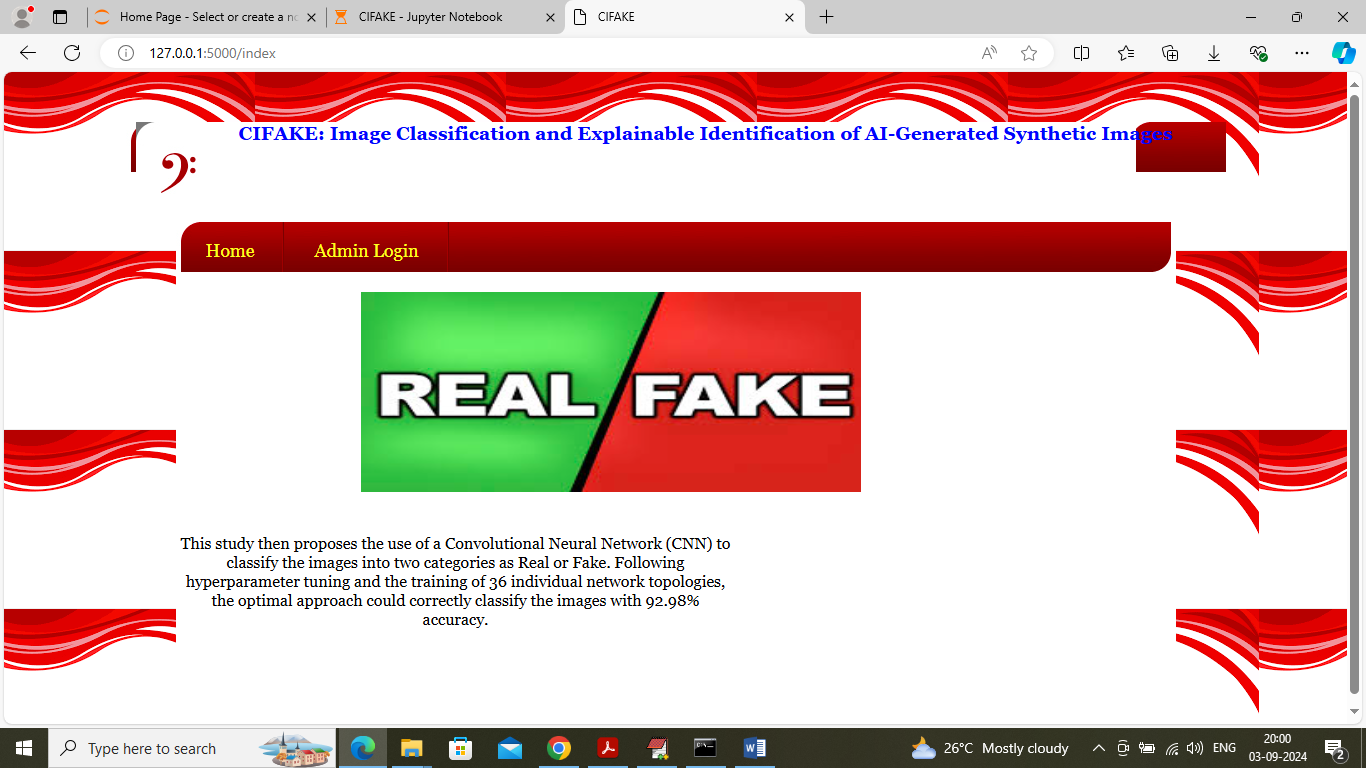
Above image classify as Real and the features which are non-black help CNN in classifying image



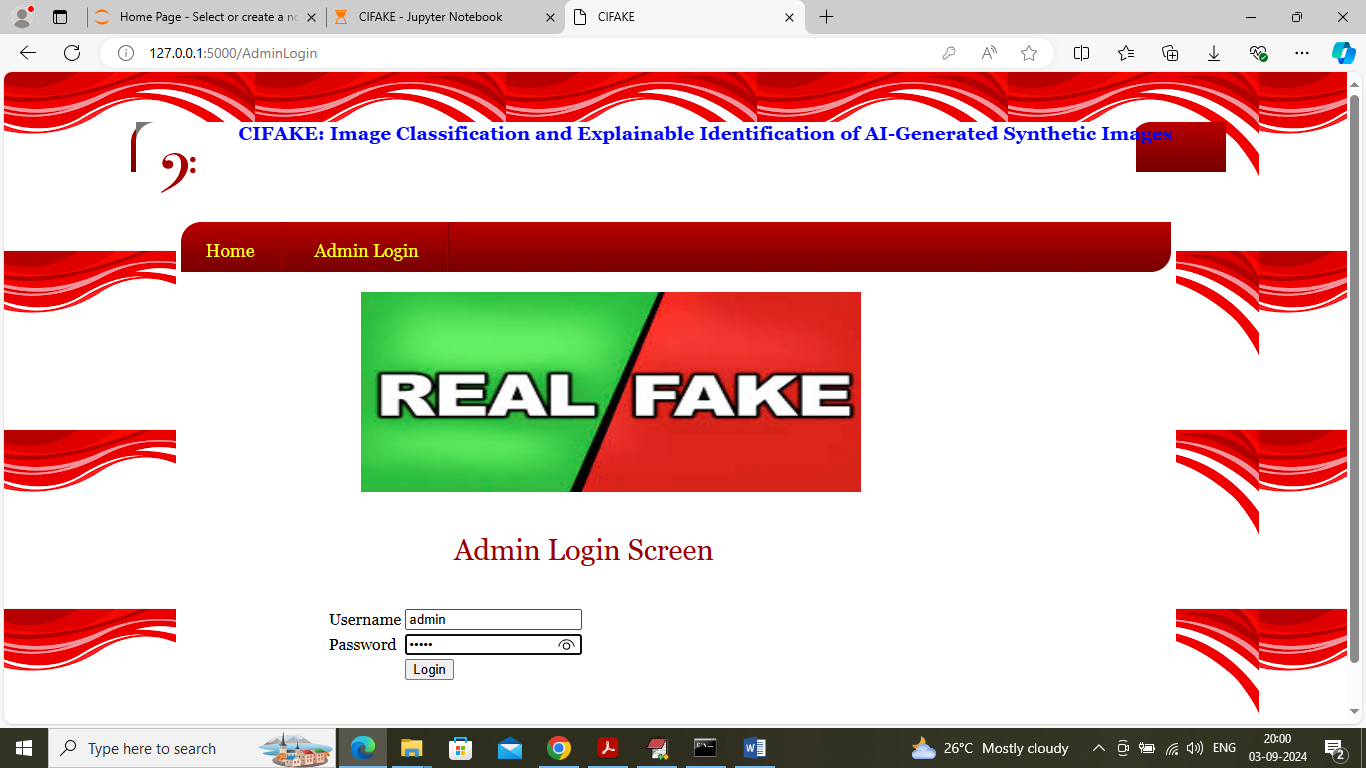
In above screen defining flask code and after running flask block will get below page



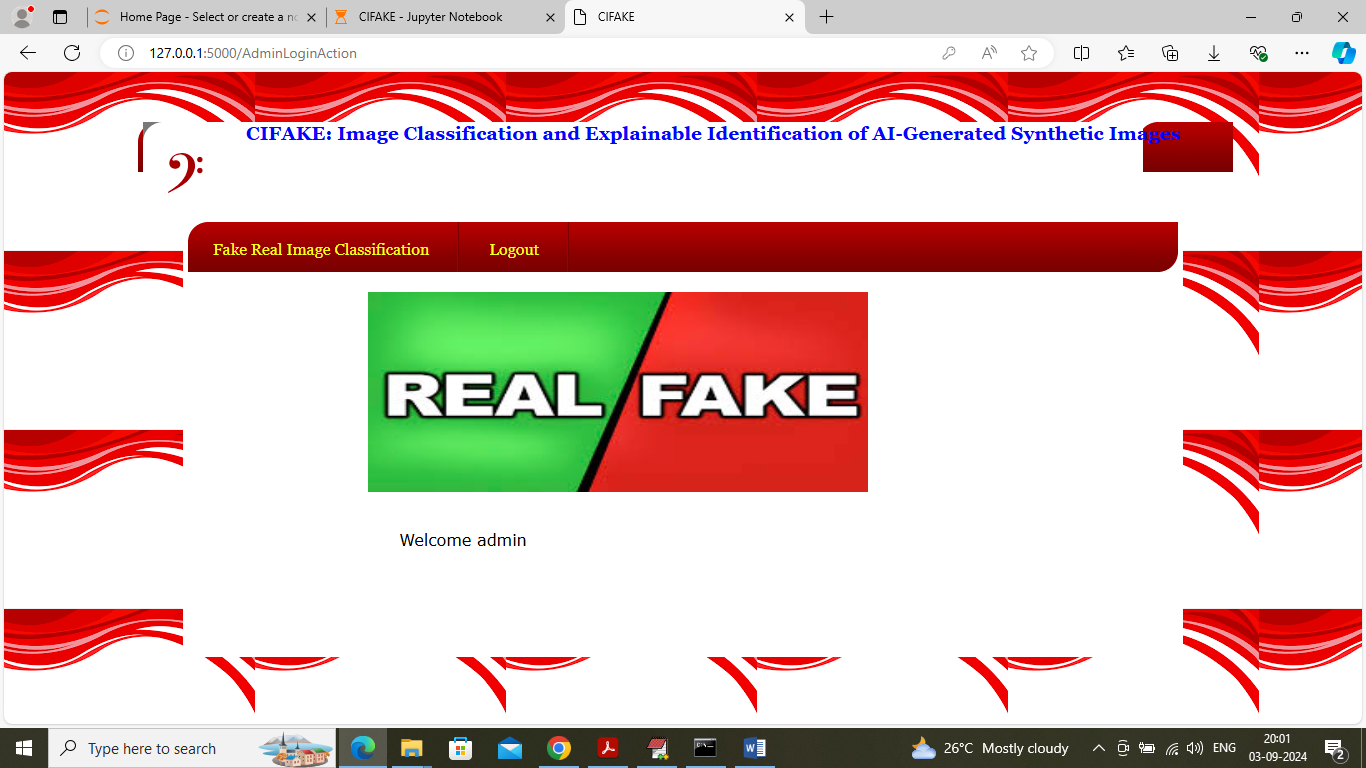
In above screen Flask server started and now open browser and enter URL as <http://127.0.0.1:5000/index> and press enter key to get below page



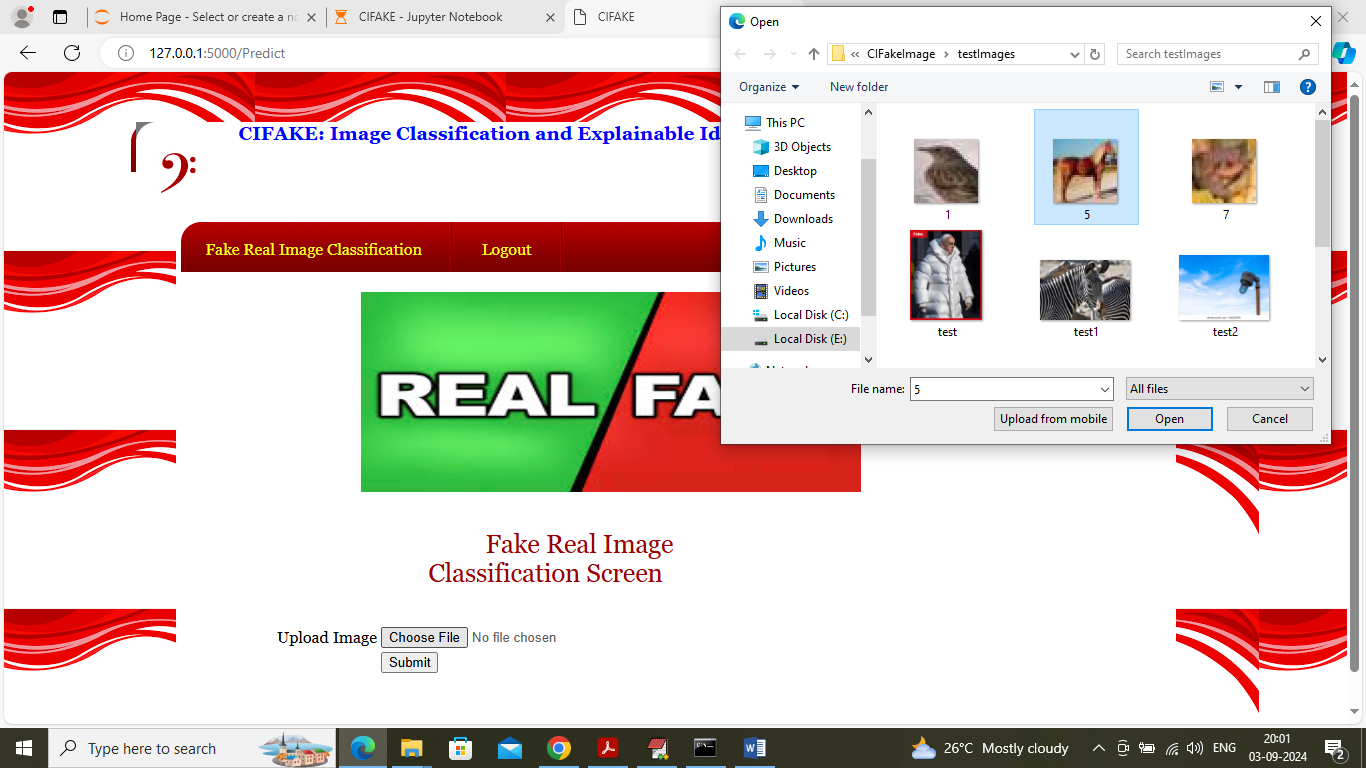
In above screen click on ‘Admin Login’ link to get below page



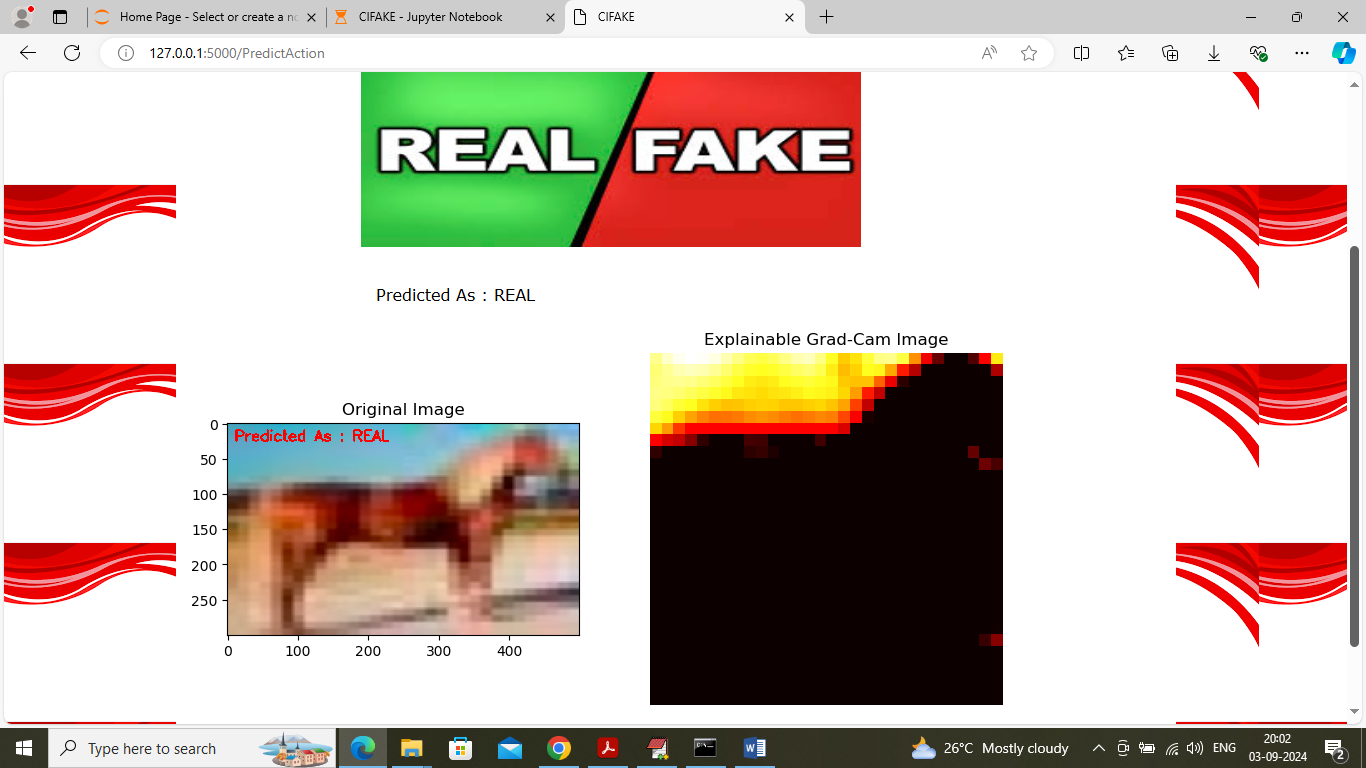
In above screen admin is login and after login will get below page



In above screen click on ‘Fake Real Image Classification’ link to get below page



In above screen selecting and uploading image and then press button to get below page



In above screen can see image classification output along with GRAD CAM explainable image which is showing dark region as the portion of image used by CNN for classification

Similarly by using above screens you can upload and test other images