Remote Sensing Image Analysis using Deep Learning

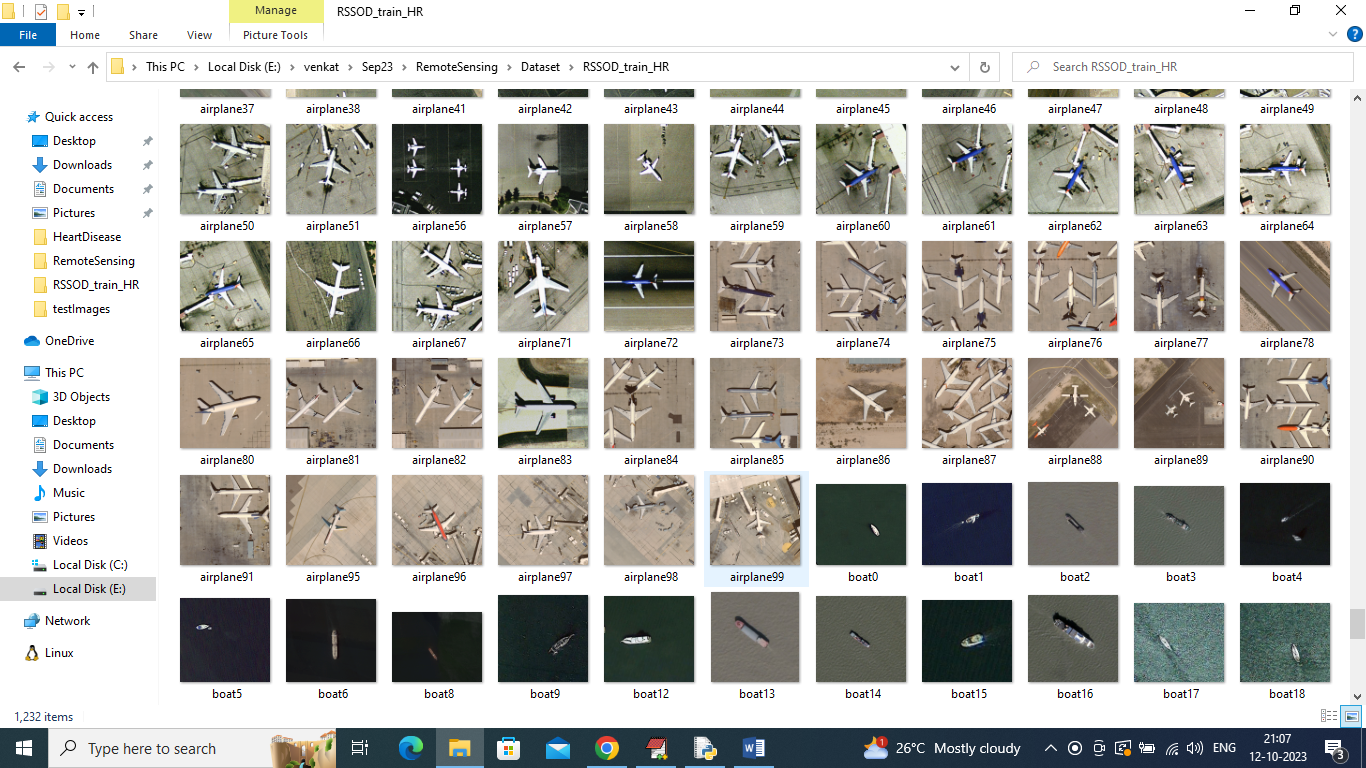
In propose work we are employing deep learning Convolution Neural Network to analyse and detect objects from Remote Sensing or Serial images. In the past many algorithms were introduced to detect objects from normal images and very few work has been done on remote images. So we employed CNN algorithm to detect objects from aerial images.

In propose work CNN algorithm objects detection accuracy is recorded between 89 to 95%. No CNN algorithms are 100% accurate so our algorithm able to identify objects correctly but bounding boxes of detected objects is little less accurate.

To train CNN algorithm we have utilized RSSOD remote sensing images dataset which can be downloaded from below website

<https://data.mendeley.com/datasets/b268jv86tf/1>

In below screen we are showing some remote sensing images from dataset



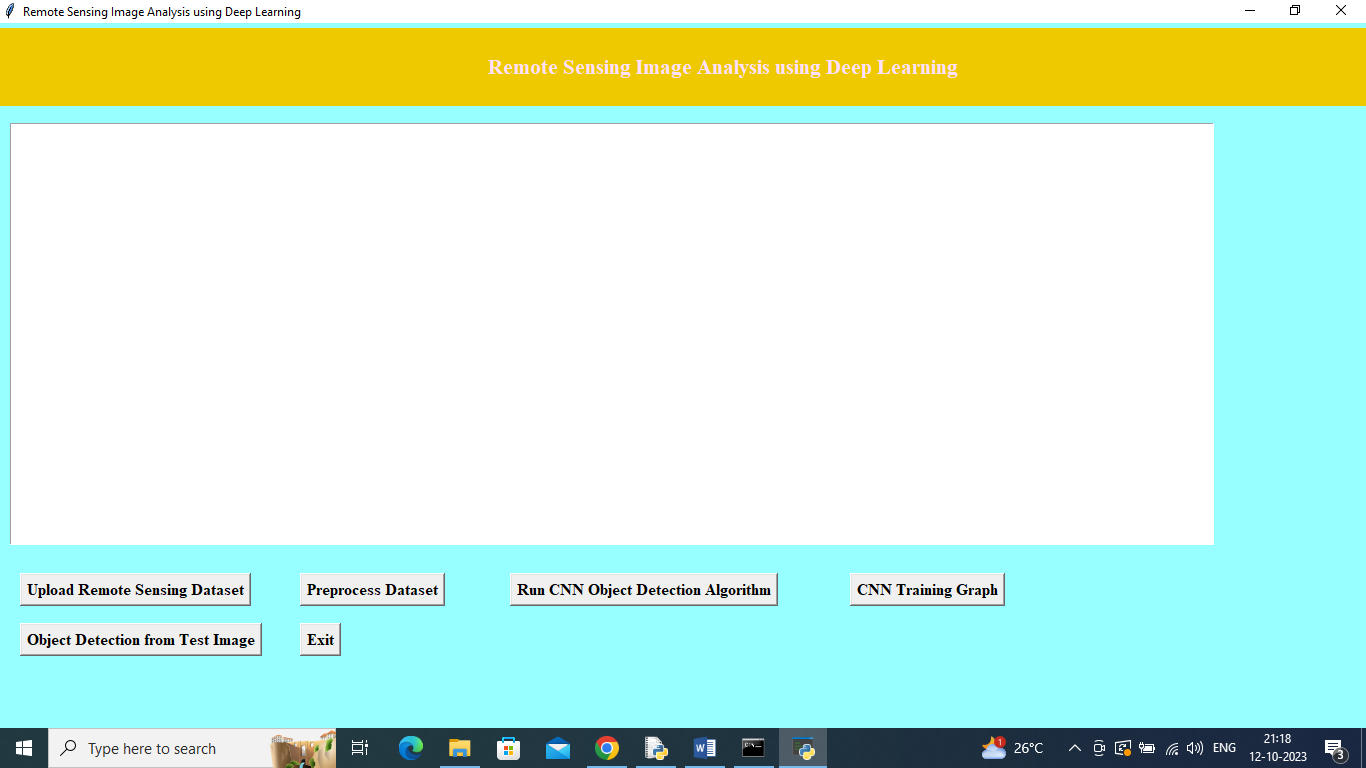
Above dataset consists of 5 different classes such as Car, Aeroplane, Vegetation, Tree and Boat and by using above dataset we will train and test CNN performance on remote sensing images.

To implement this project we have designed following modules

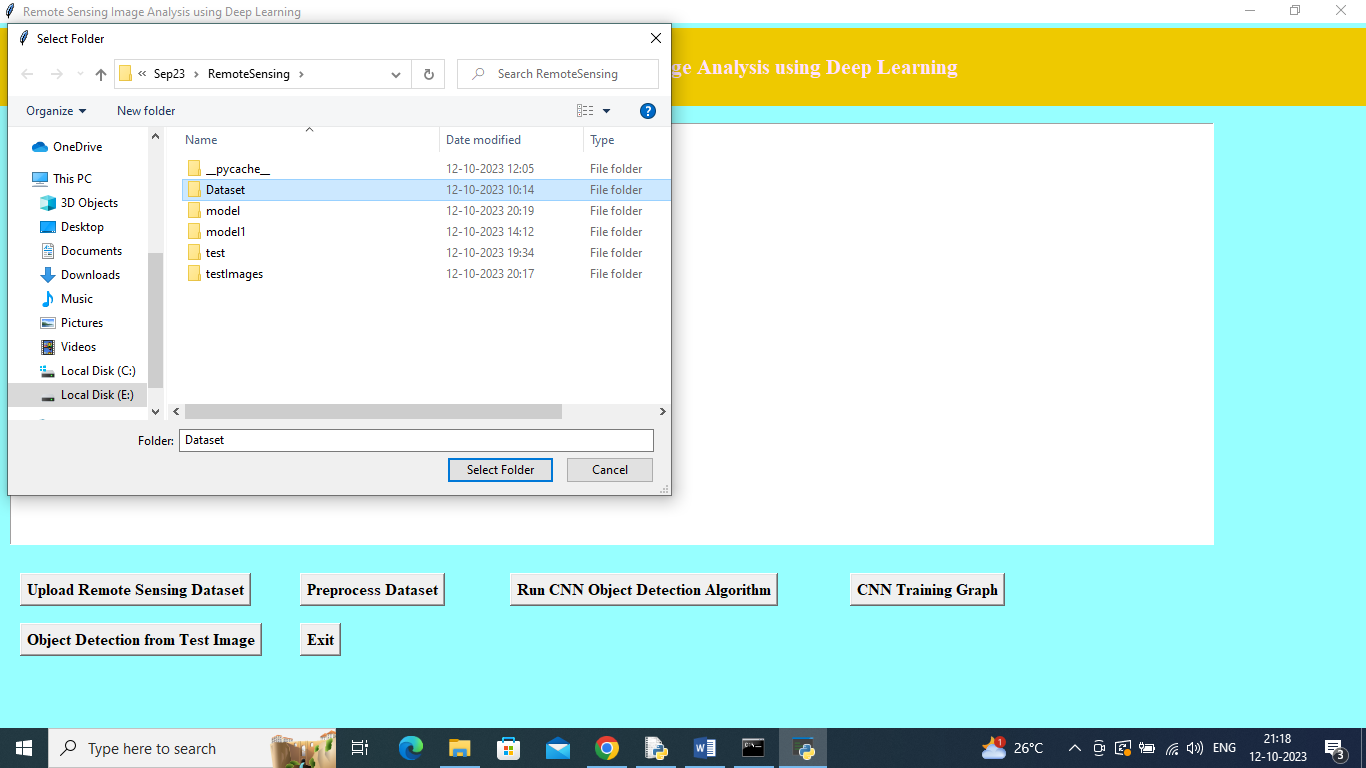
1. Upload Remote Sensing Dataset: using this module we can upload dataset to application and then read all images and bounding boxes to generate training features
2. Pre-process Dataset: using this module we will normalize, shuffle and split dataset into train and test
3. Run CNN Object Detection Algorithm: In this module CNN will get trained on training features and then its performance will be evaluated using test data features. Performance evaluation will be done in terms of accuracy, precision, recall, Confusion Matrix and FSCORE
4. CNN Training Graph: using this module we will plot CNN training and validation accuracy graph
5. Object Detection from Test Image: using this module we will upload test image and then CNN will detect and predict object from input image.

SCREN SHOTS

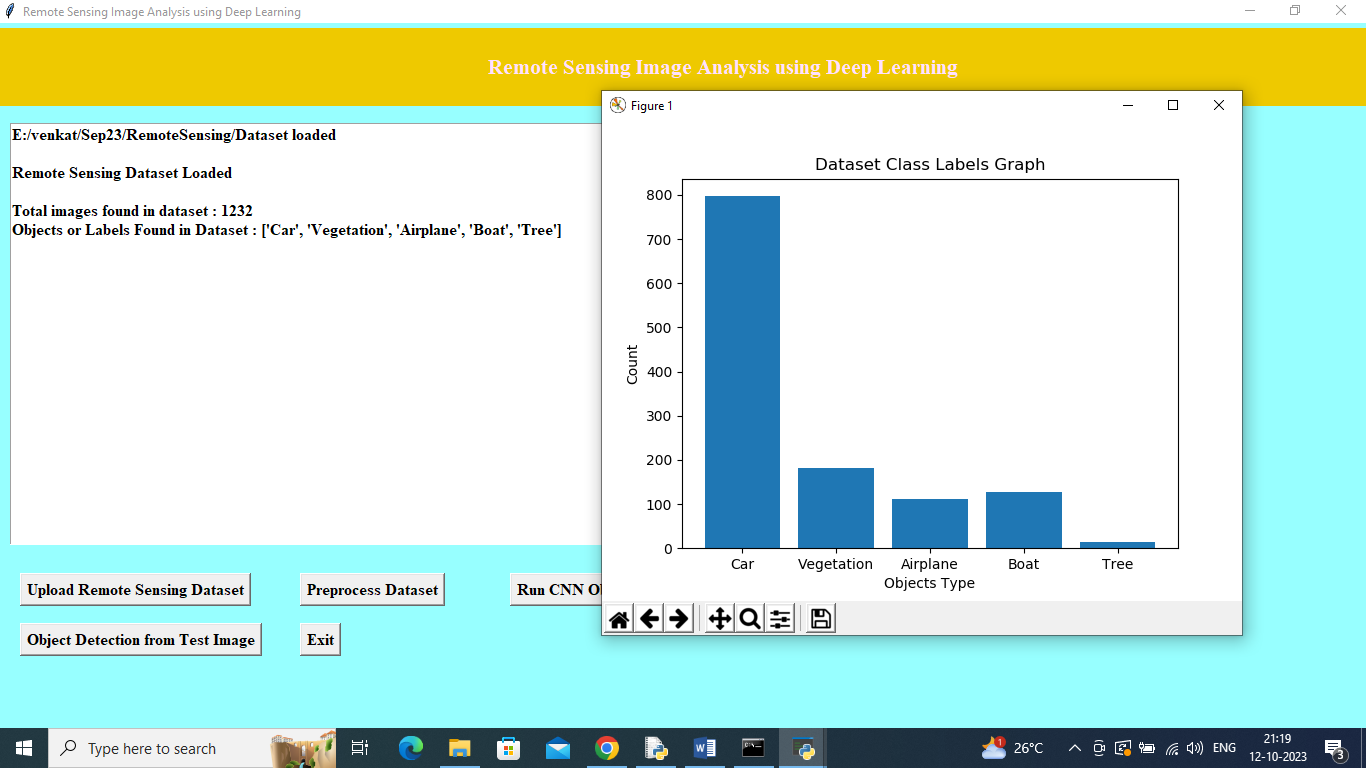
To run project double click on ‘run.bat’ file to get below screen



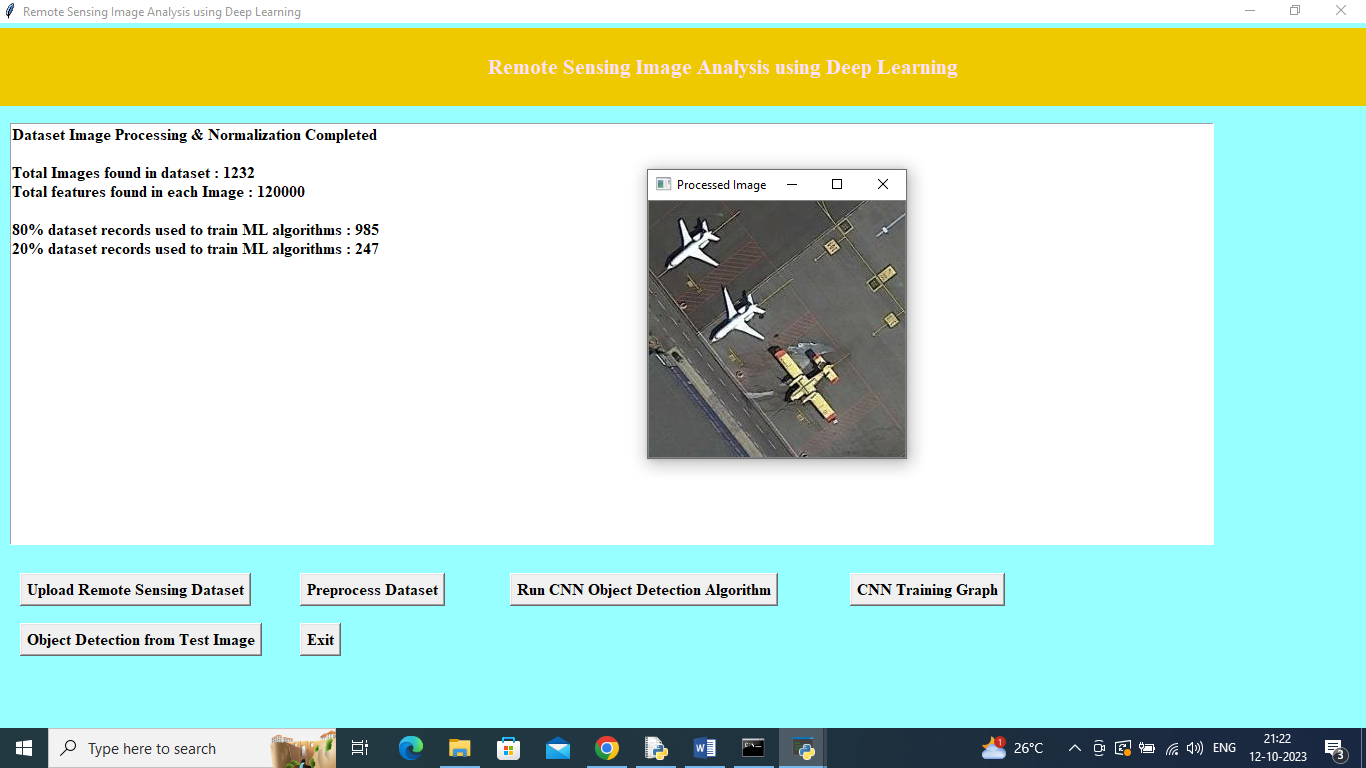
In above screen click on ‘Upload Remote Sensing Dataset’ button to upload dataset and get below output



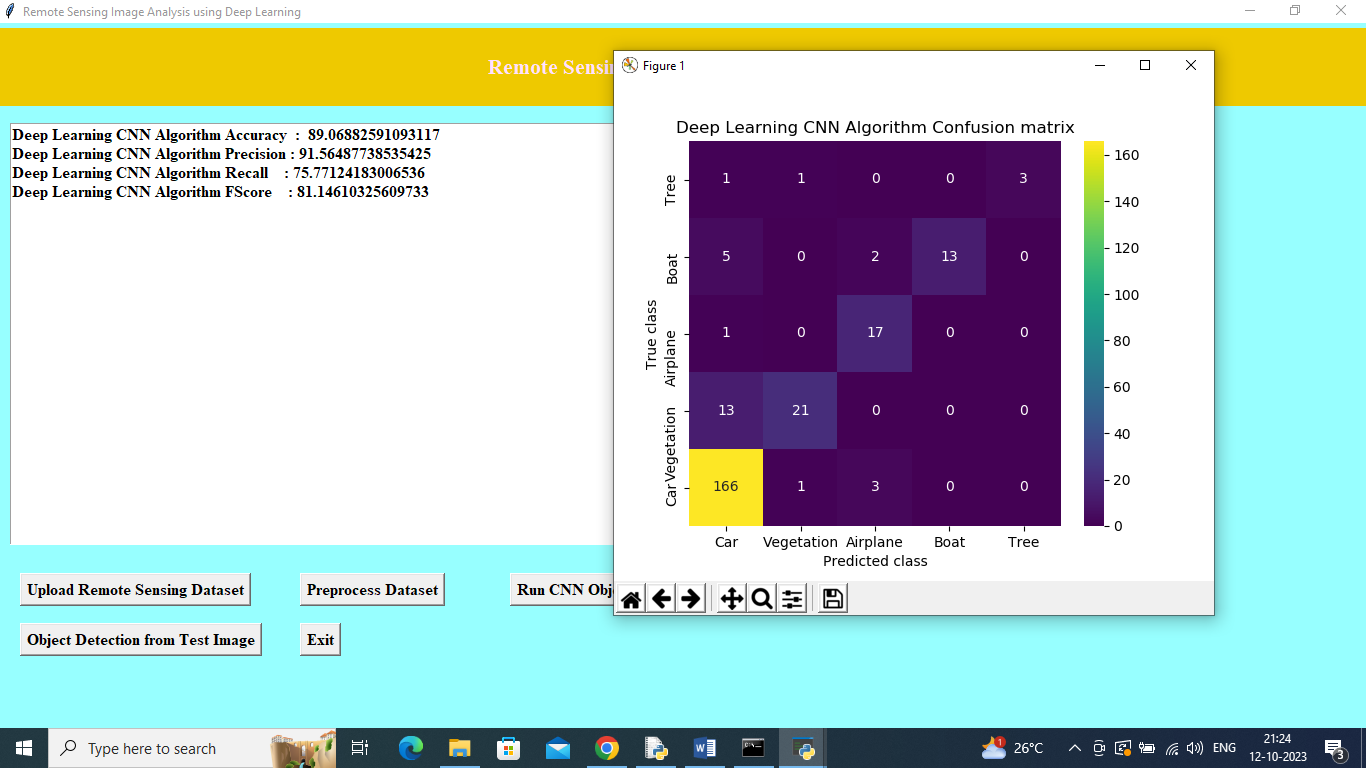
In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and then will get below output



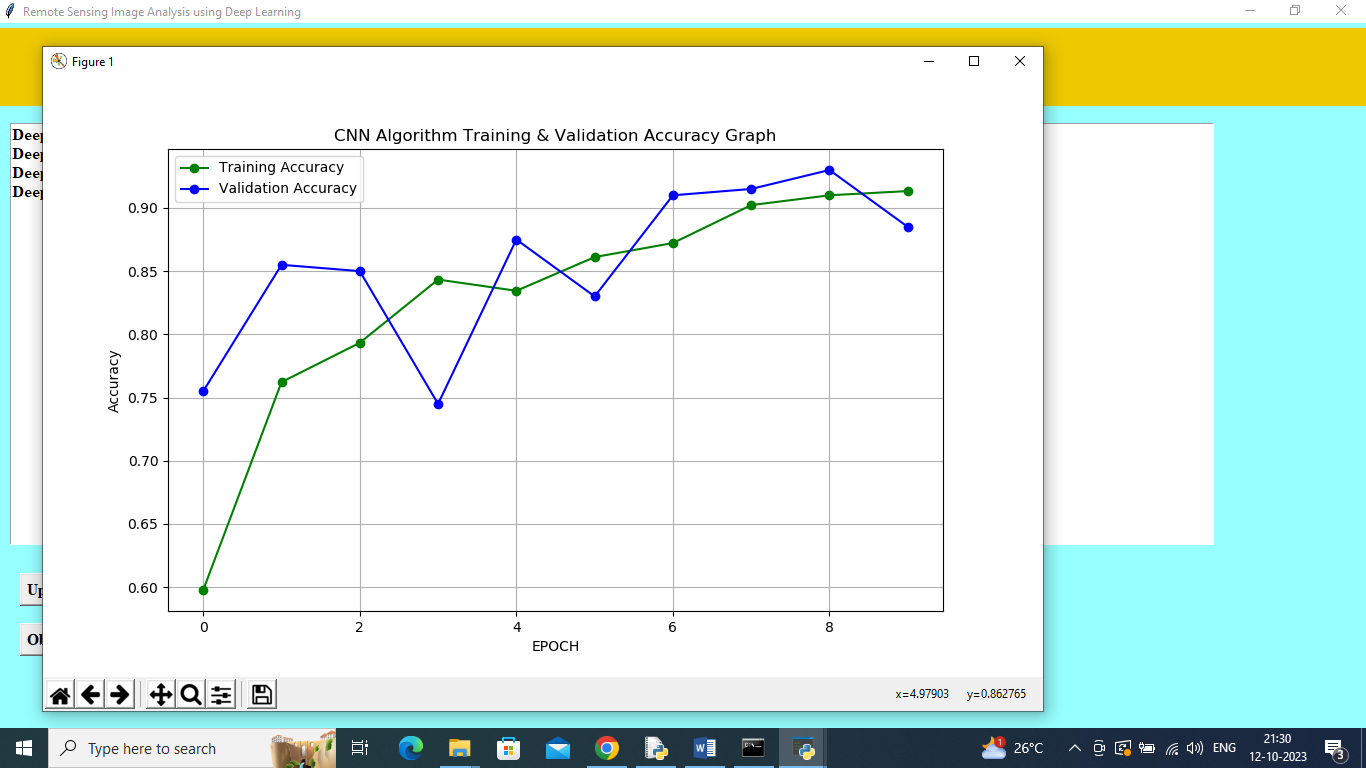
In above screen can see dataset loaded and then can see available labels and in graph x-axis represents Labels and y-axis represents number of images found under that label. Now close above graph and then click on ‘Pre-process dataset’ button to process images and get below output



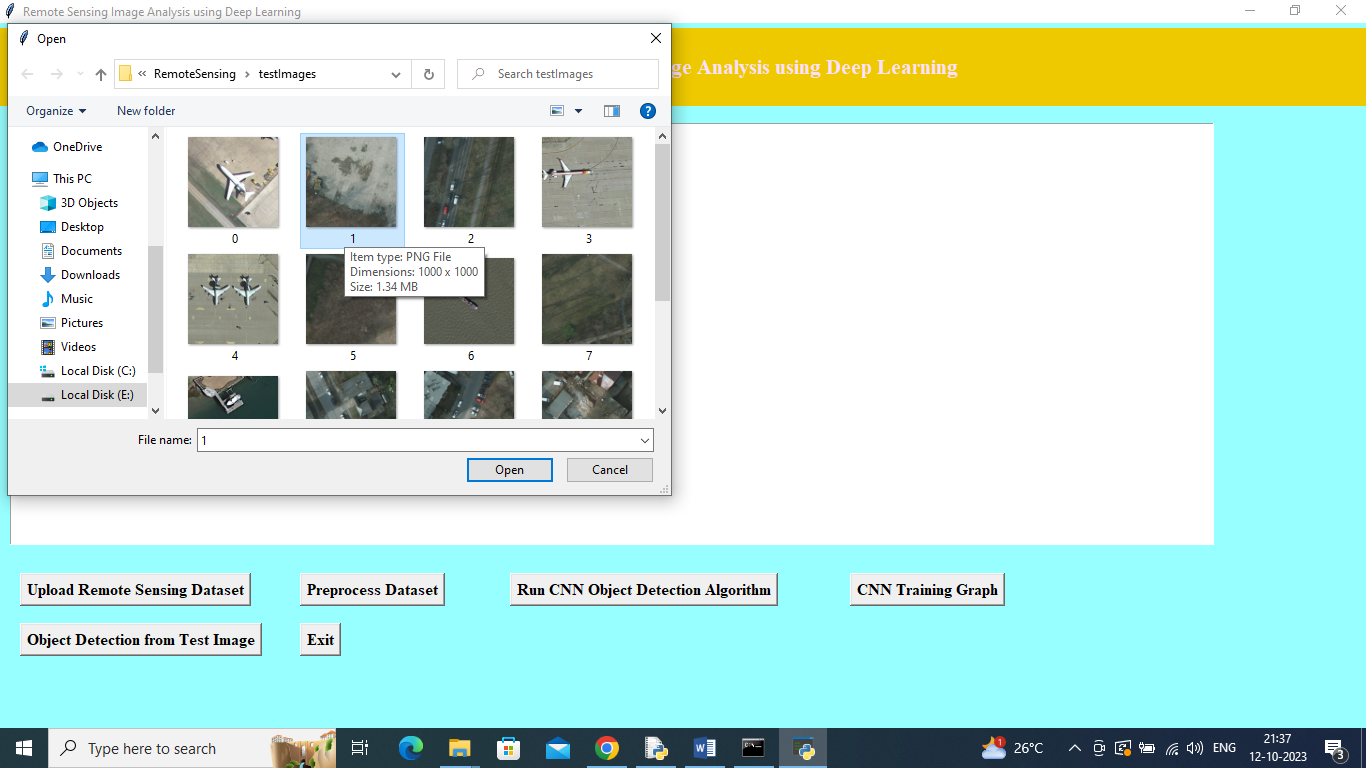
In above screen can see total loaded images and number of features available in each image and then can see training and test size and then can see processed sample image and now close above image and then click on ‘Run CNN Object Detection Algorithm’ button to train CNN and get below output



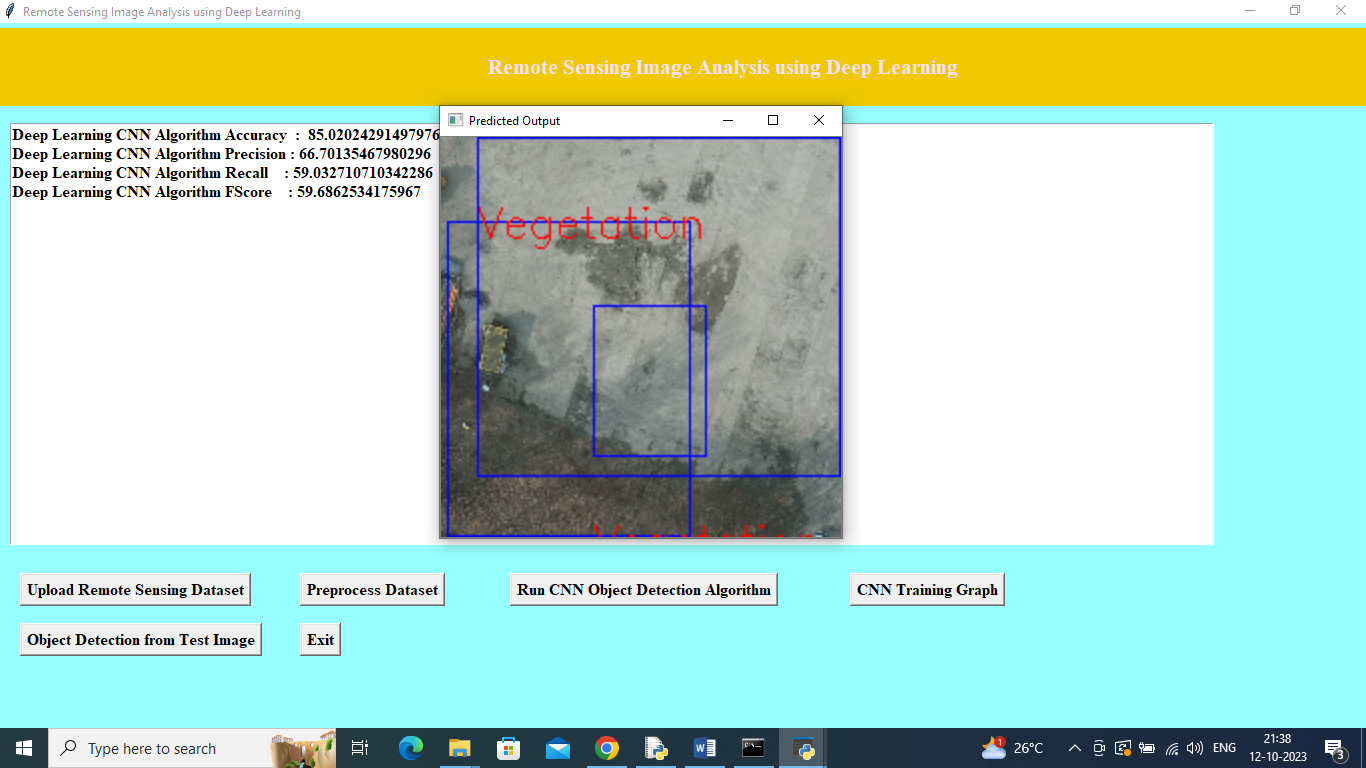
In above screen CNN training complete and it got an accuracy of 89% and can see other metrics like precision, recall and FSCORE. In Confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all counts in diagnol boxes represents correct prediction count and remaining boxes represents incorrect prediction count. Now close above graph and then click on ‘CNN Training Graph’ button to get below graph



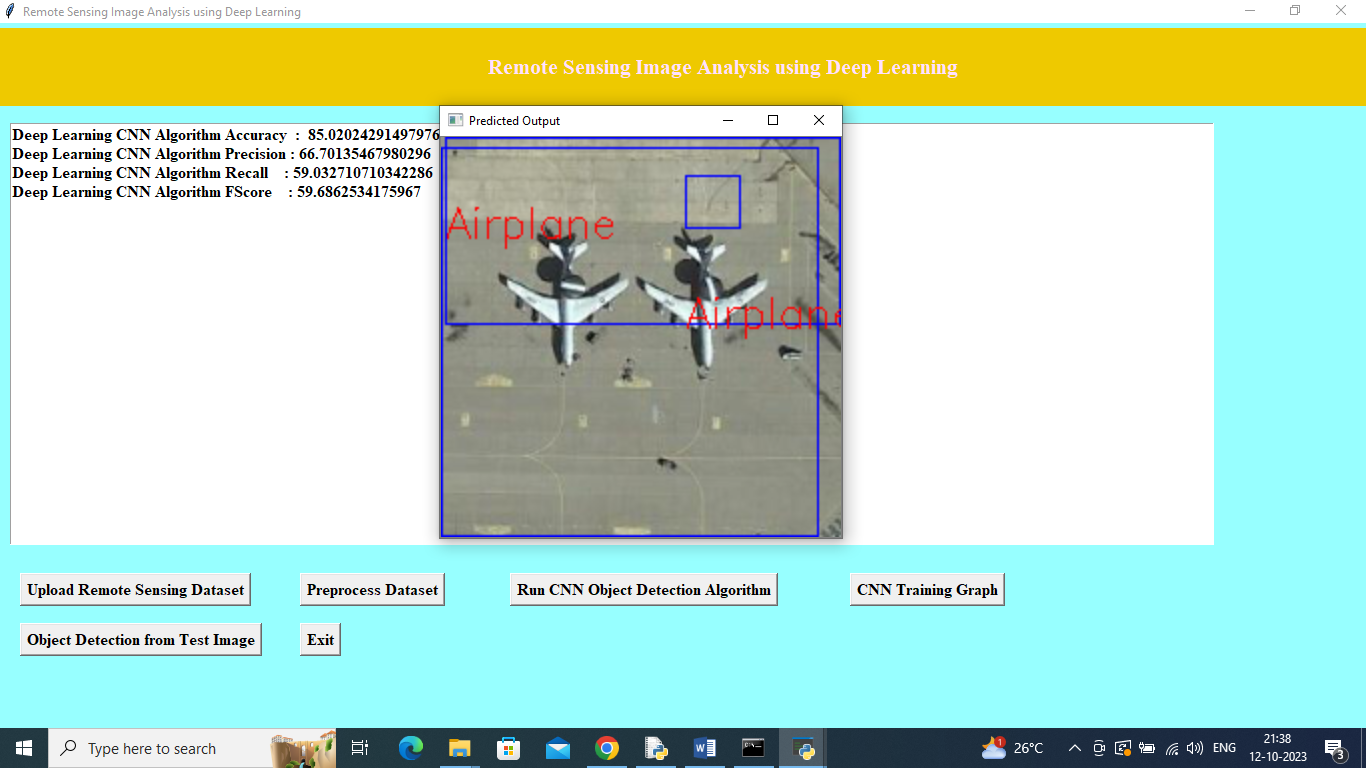
In above graph x-axis represents training epoch and y-axis represents training and validation accuracy where green line is for training accuracy and blue line for validation accuracy and with each increasing epoch both training and validation accuracy got increased and now close above image and then click on ‘Object Detection from Test Image’ button to upload test image like below screen



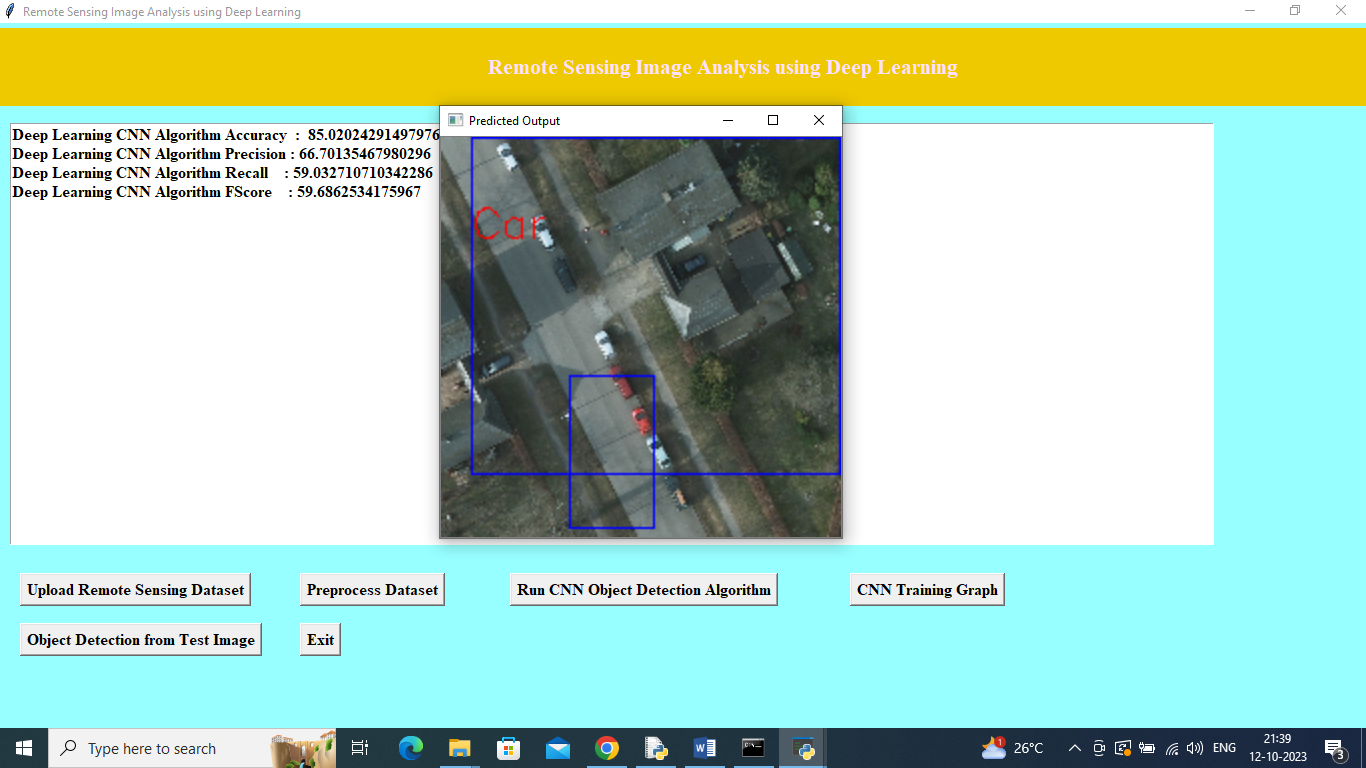
In above screen selecting and uploading ‘1.png’ file and then click on ‘Open’ button to get below output



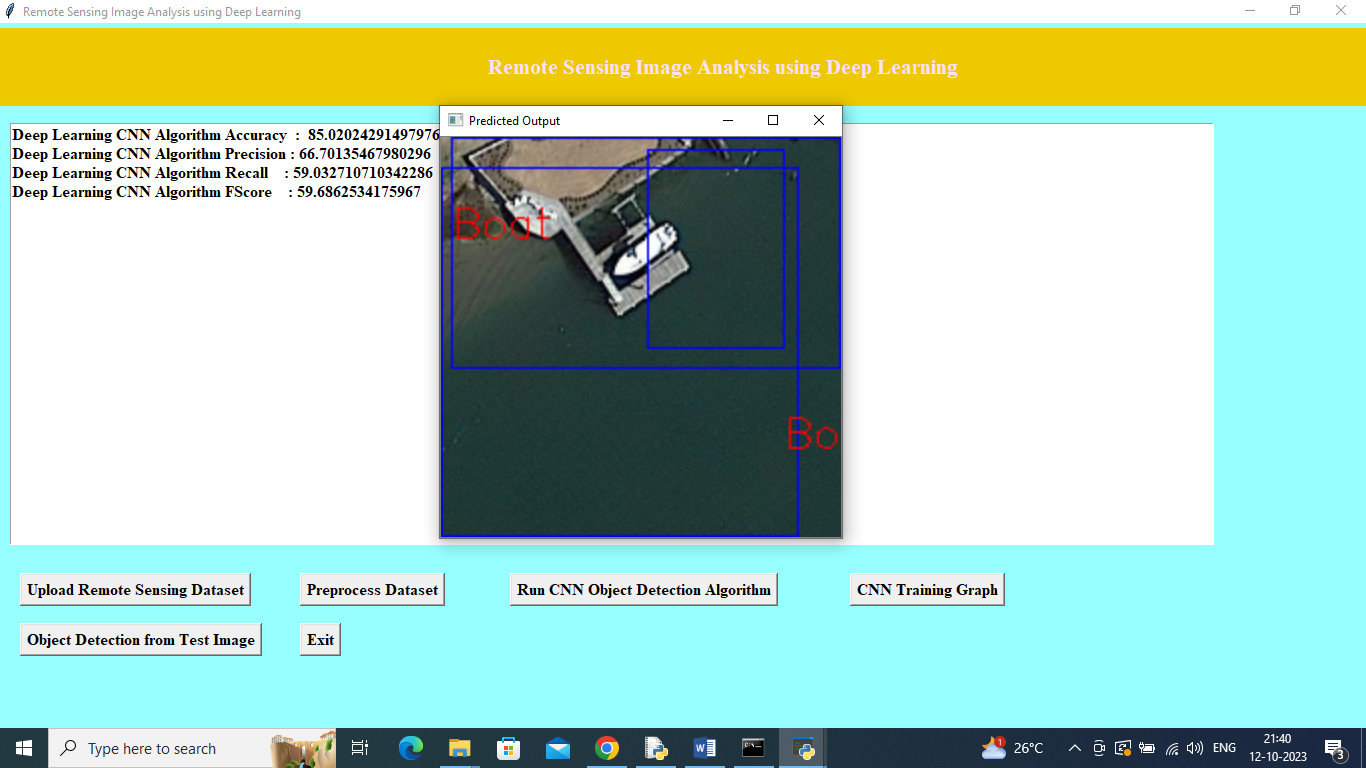
In above output all green parts are identified as ‘Vegetation’ and similarly you can upload and test other images



In above screen object detected and identified as Aeroplane



In above screen object detected as Car



In above screen object detected as Boat