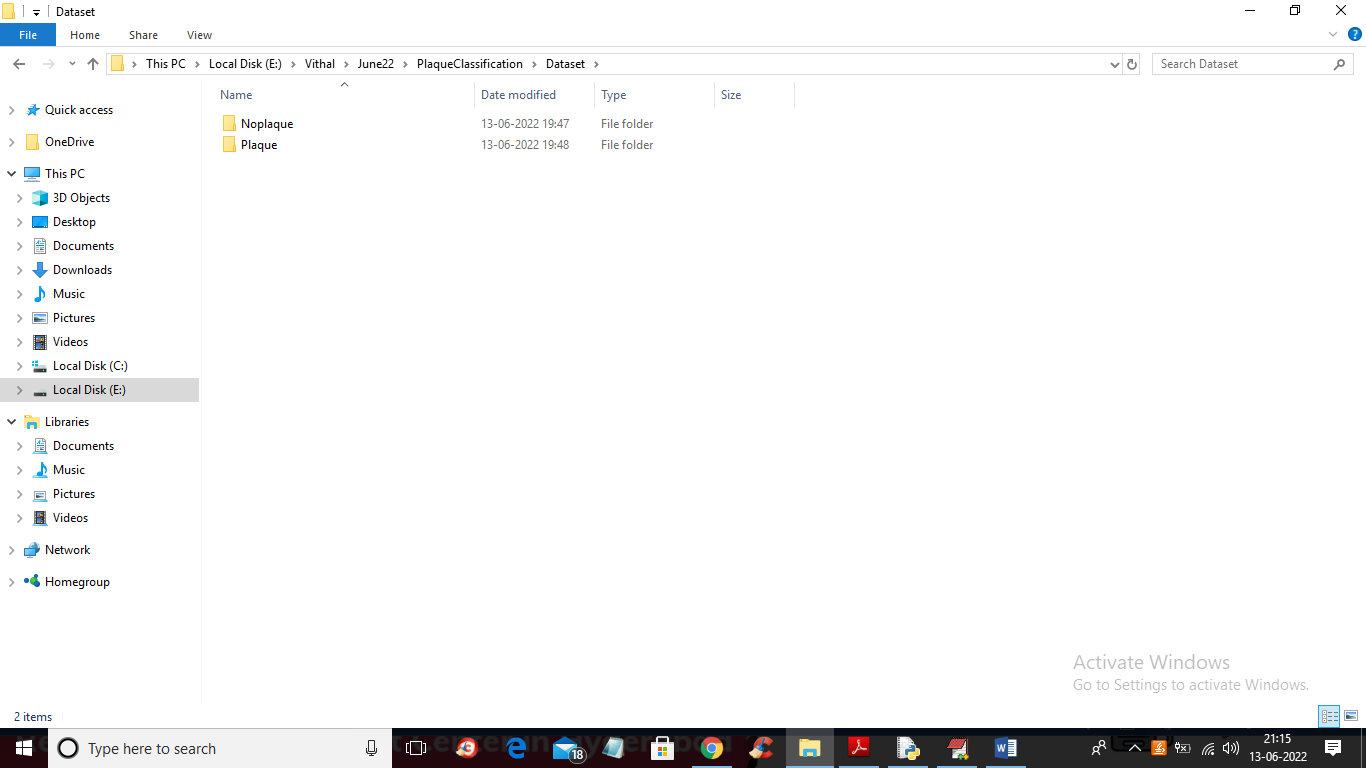
A Recurrent CNN for Automatic Detection and Classification of Coronary Artery Plaque and Stenosis in Coronary CT Angiography

In propose paper author is apply Recurrent Convolution Neural Network (RCNN) to detect plaque and Stenosis from Coronary Artery CCTA (clinically obtained coronary CT angiography) images. All existing algorithms were able to predict presence of Plaque or Not but they will not predict Significance of Stenosis and to overcome from this problem author has introduced RCNN which can predict two task.

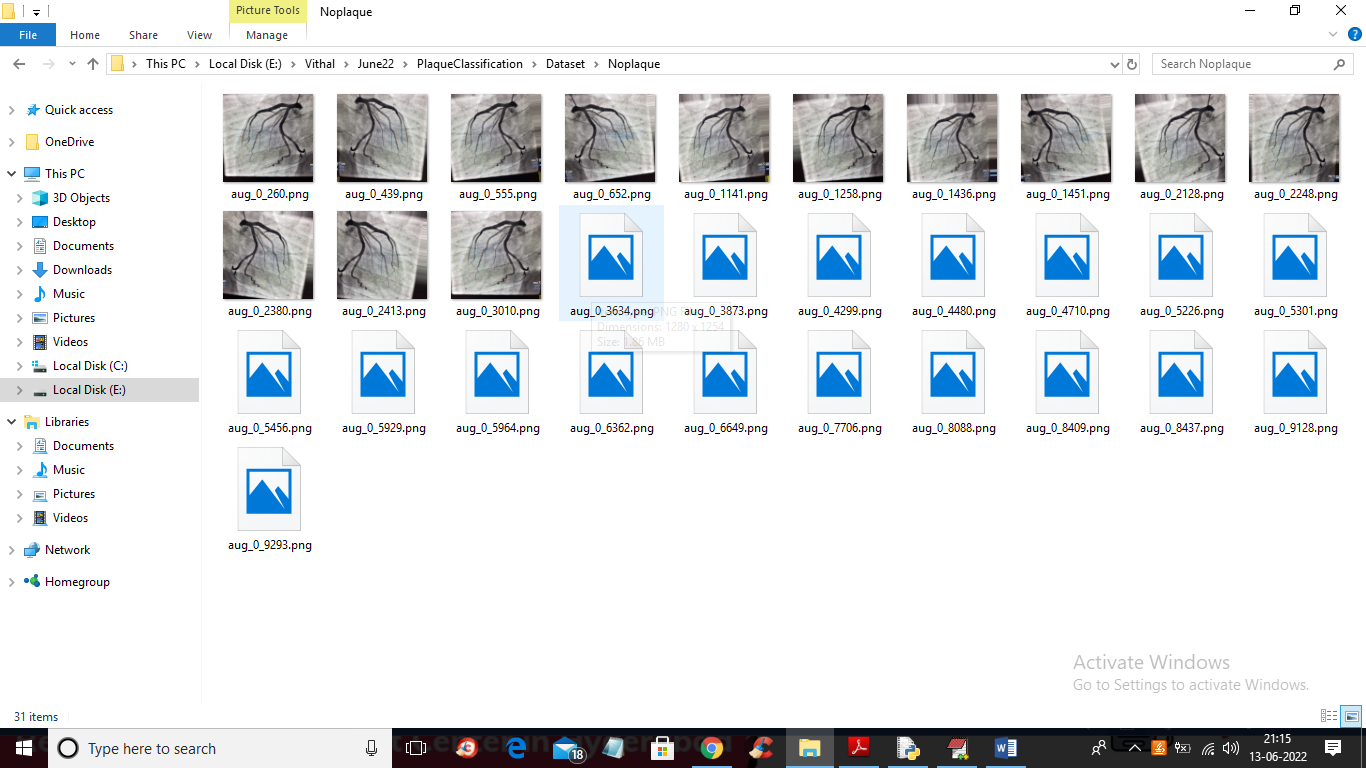
In task one RCNN predict Plaque classification and in task two predict percentage of Stenosis. White line in image indicate Stenosis percentage and if percentage is more than 50 then Stenosis presence detected.

To implement this project author has used annotated images from experts which classify plaque into ‘No-Plaque, Non-Calcified, Mixed and Calcified’ but we are unable to find such dataset so we are predicting Healthy and Plaque images and then calculating percentage of Stenosis.

To implement this project we have used below dataset



In above screen we have 2 folder and just go inside any folder to view CCTA images



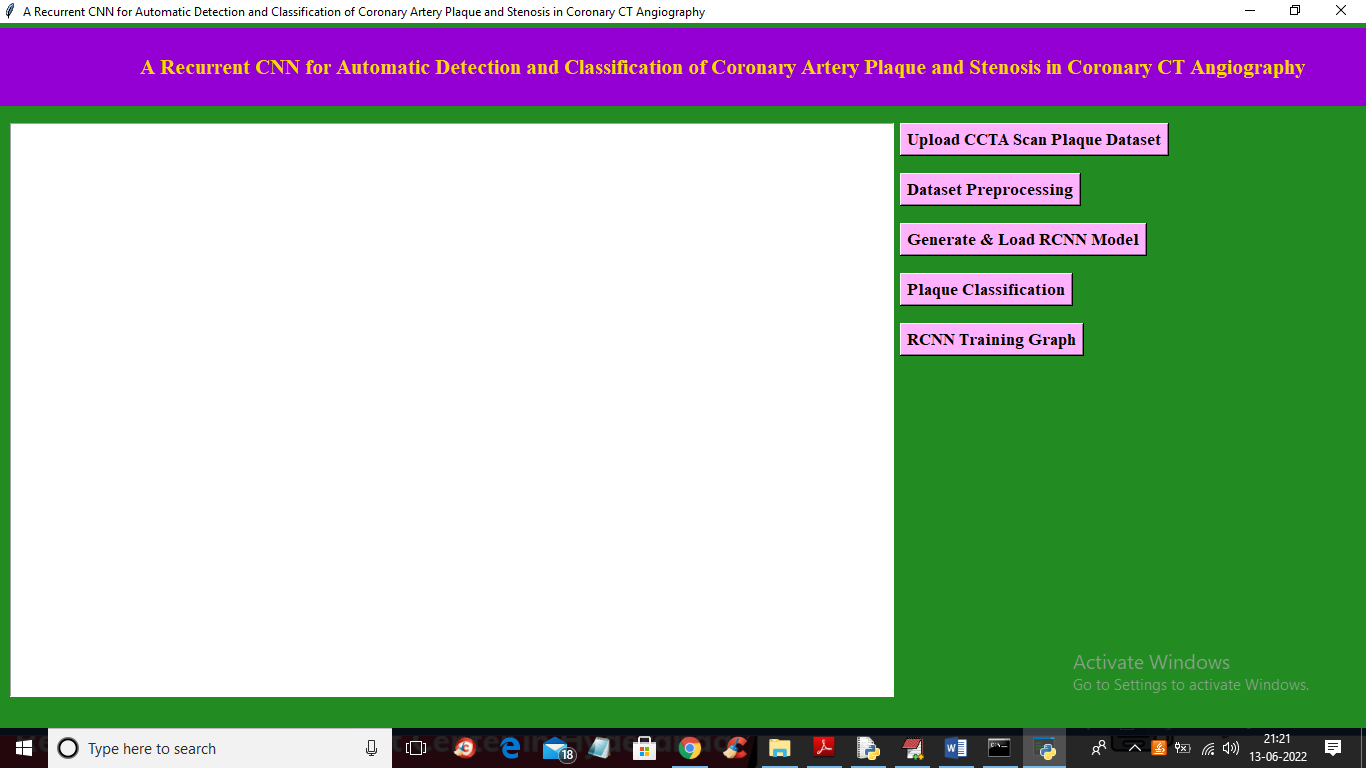
By using above image we are training CNN algorithm

To implement this project we have designed following modules

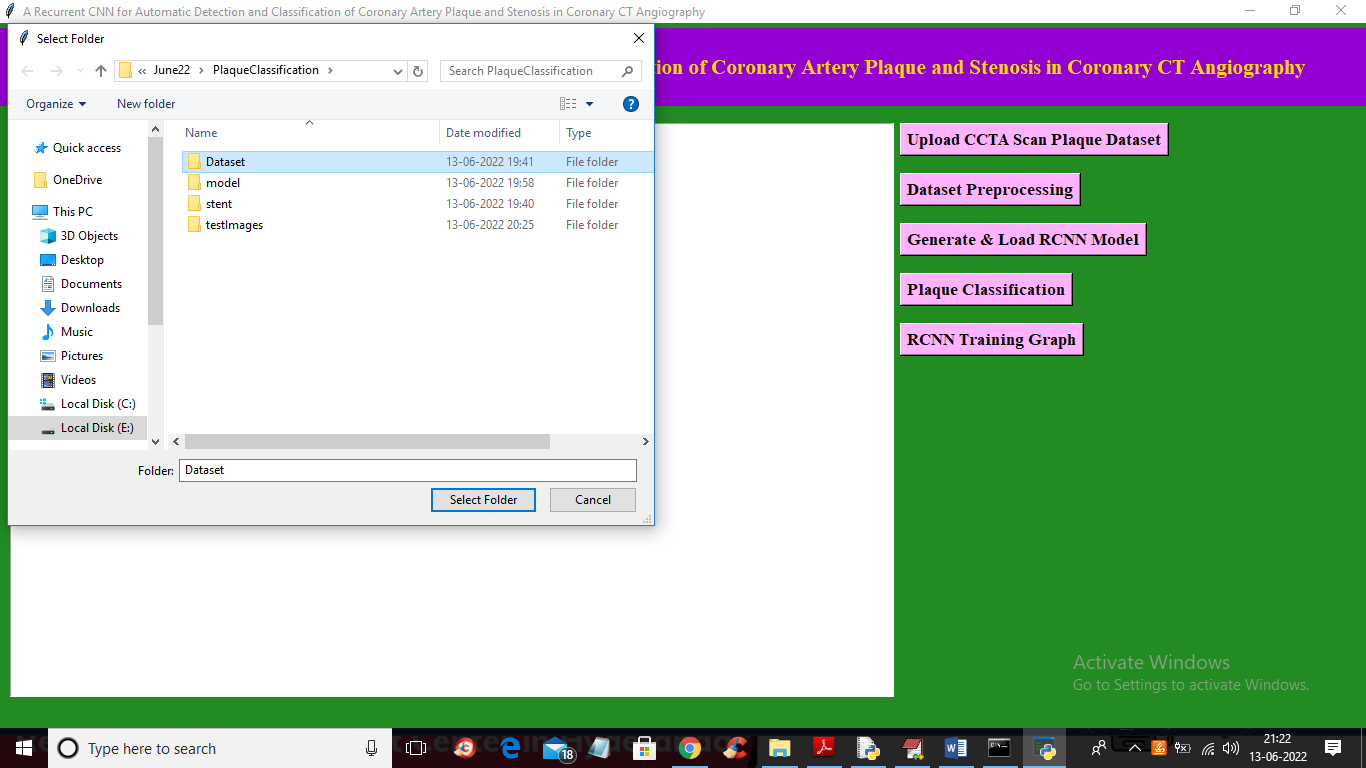
1. Upload CCTA Scan Plaque Dataset: using this module we will upload CCTA images to application
2. Dataset Preprocessing: using this module we will read all images and then resize all images to equal size and then normalize all images values and then split dataset images into train and test where application using 80% images for training 20% for testing
3. Generate & Load RCNN Model: using this module we will train RCNN using 80% images and then test RCNN performance by applying 20% test images prediction
4. Plaque Classification: using this module we will upload test image and then RCNN will predict presence of Plaque and stenosis.
5. RCNN Training Graph: using this module we will plot RCNN training accuracy and loss graph

SCREEN SHOTS

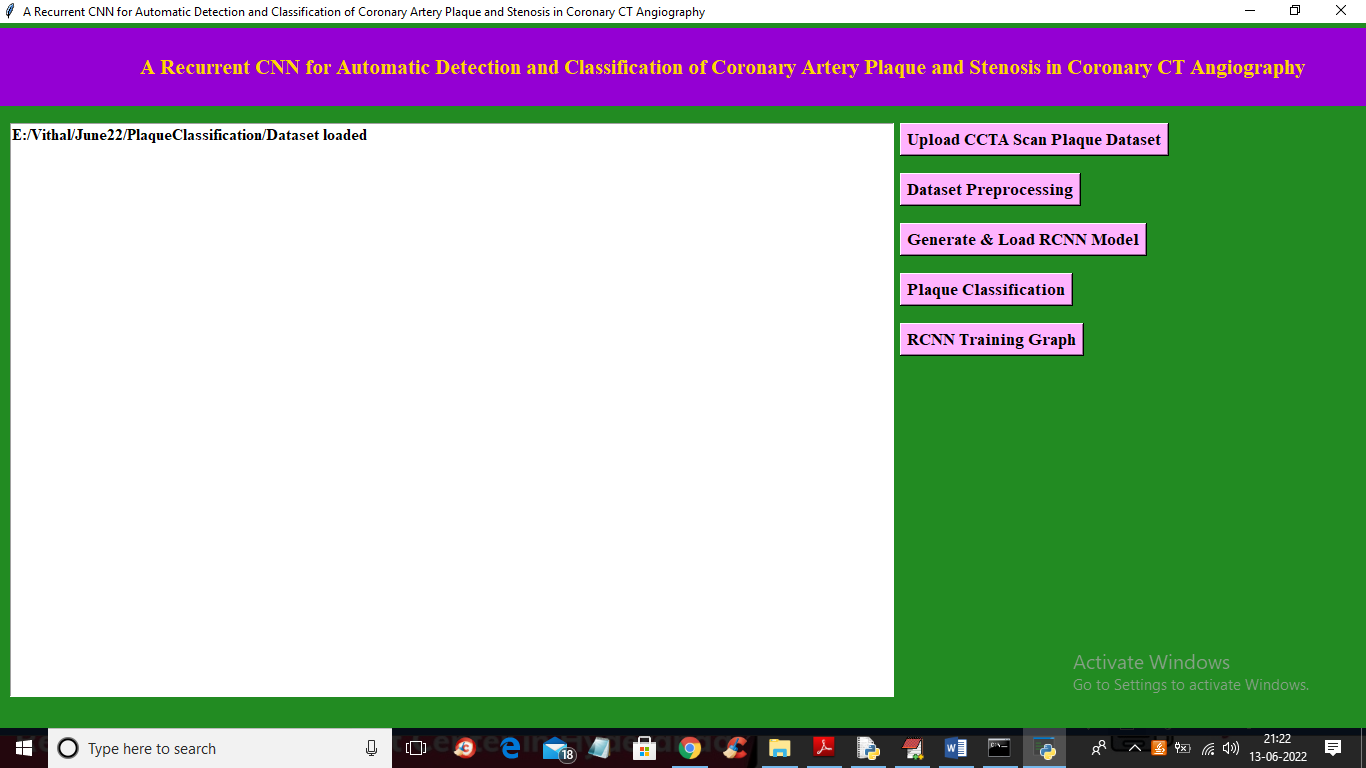
To run project double click on ‘run.bat’ file to get below screen



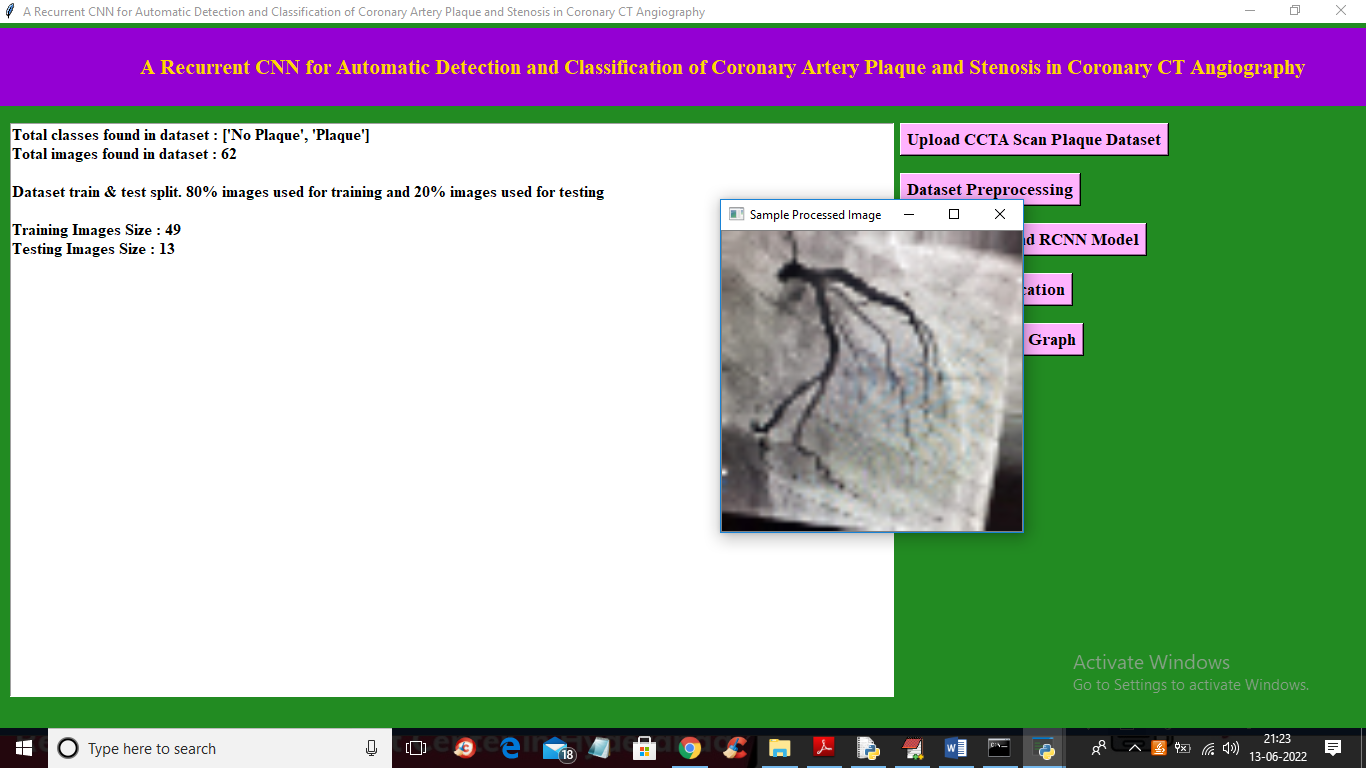
In above screen click on ‘Upload CCTA Scan Plaque Dataset’ button to upload dataset and get below output



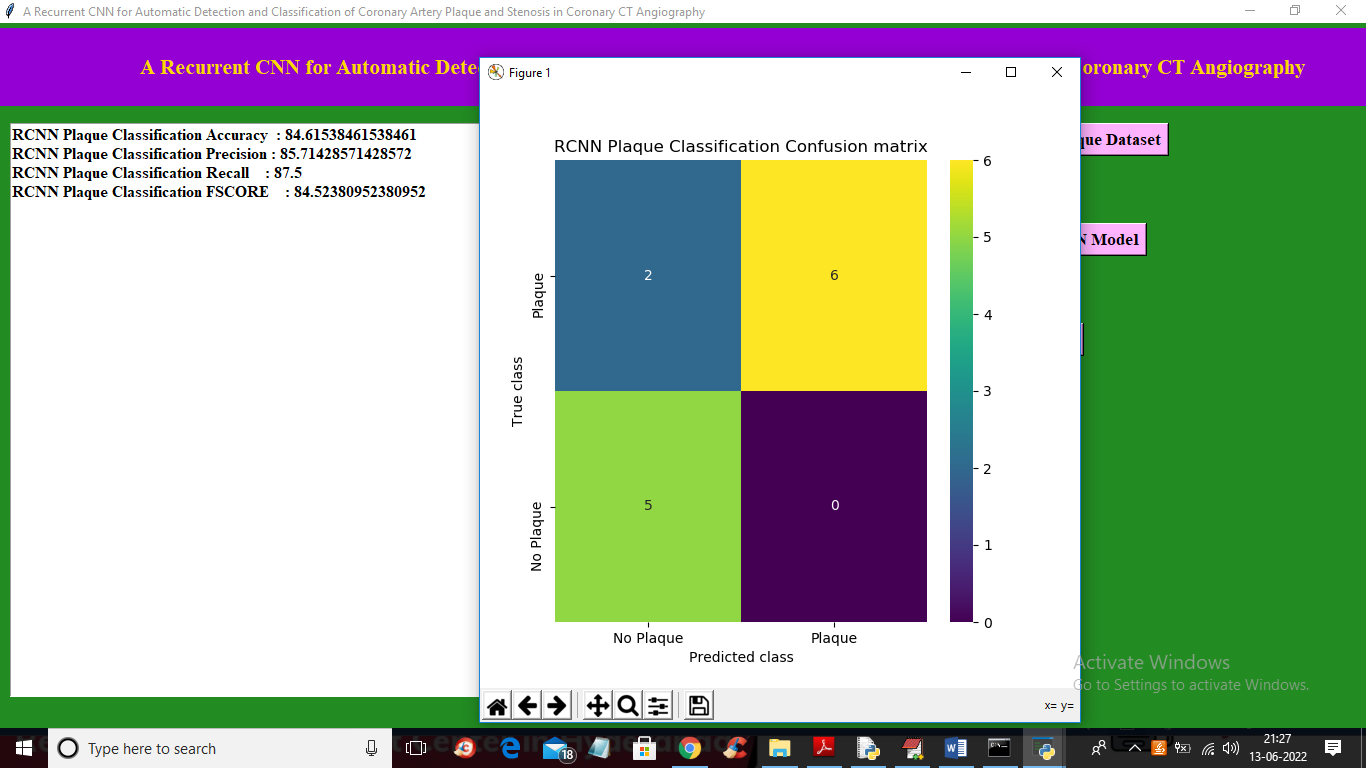
In above screen selecting and uploading dataset and then click on ‘Select Folder’ button to load dataset and get below output



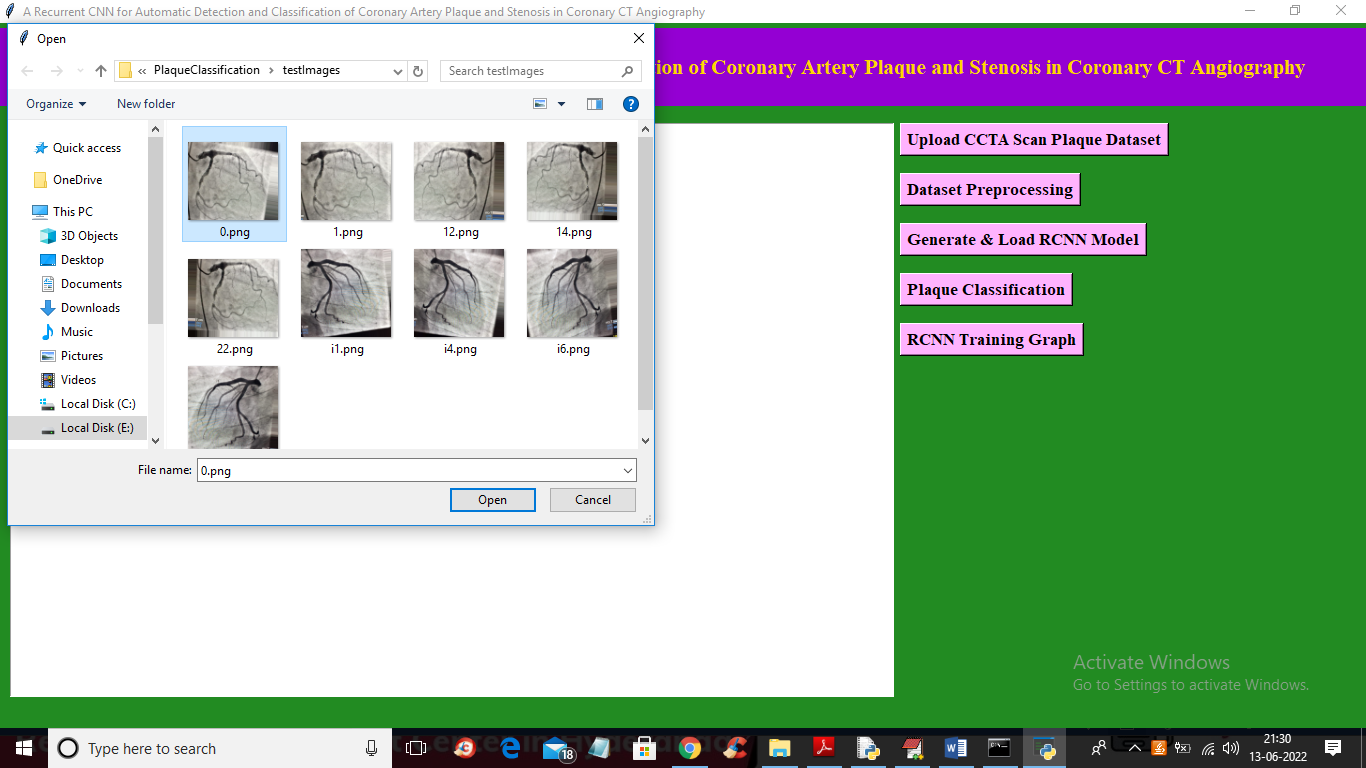
In above screen dataset loaded and now click on ‘Dataset Preprocessing’ button to process dataset and get below output



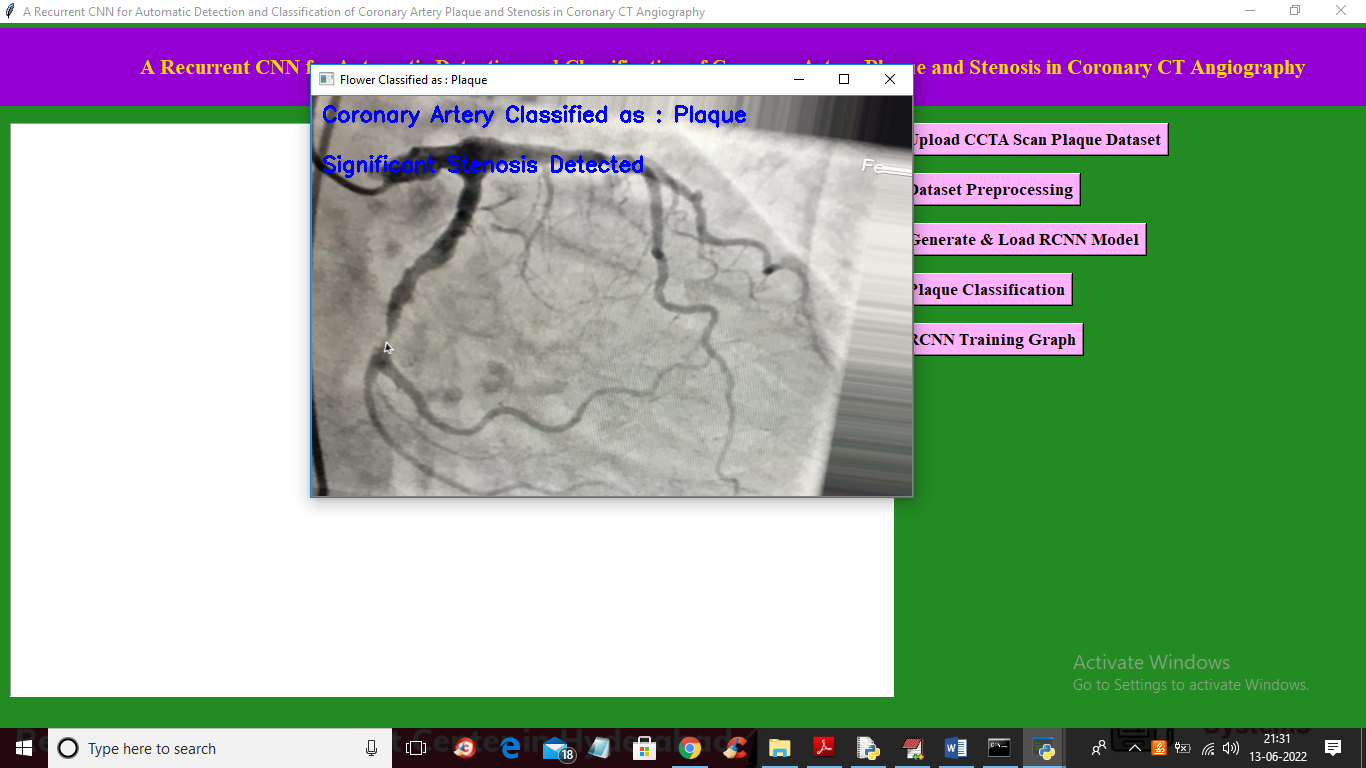
In above screen we can see class labels found in dataset and total images available in dataset with train and test images size and in above screen displaying sample process image to check all images are loaded and process properly and now close above image and then click on ‘Generate & Load RCNN Model’ button to build RCNN model and get below output



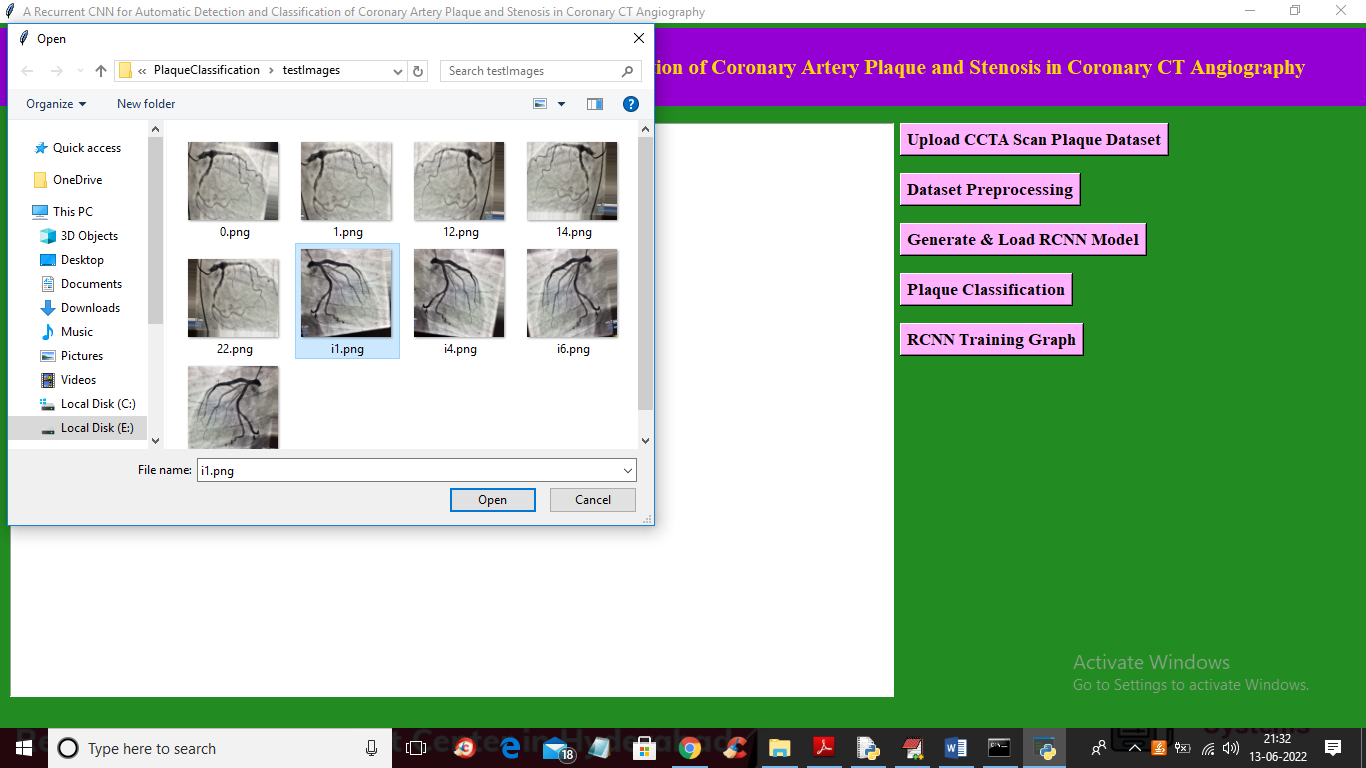
In above screen with RCNN we got accuracy as 84% and in confusion matrix graph x-axis represents Predicted Classes and y-axis represents True classes and in above graph we can see only 2 images of ‘No Plaque’ is incorrectly predicted as ‘Plaque’. Now close above graph and then click on ‘Plaque Classification’ button to upload test image and get below output



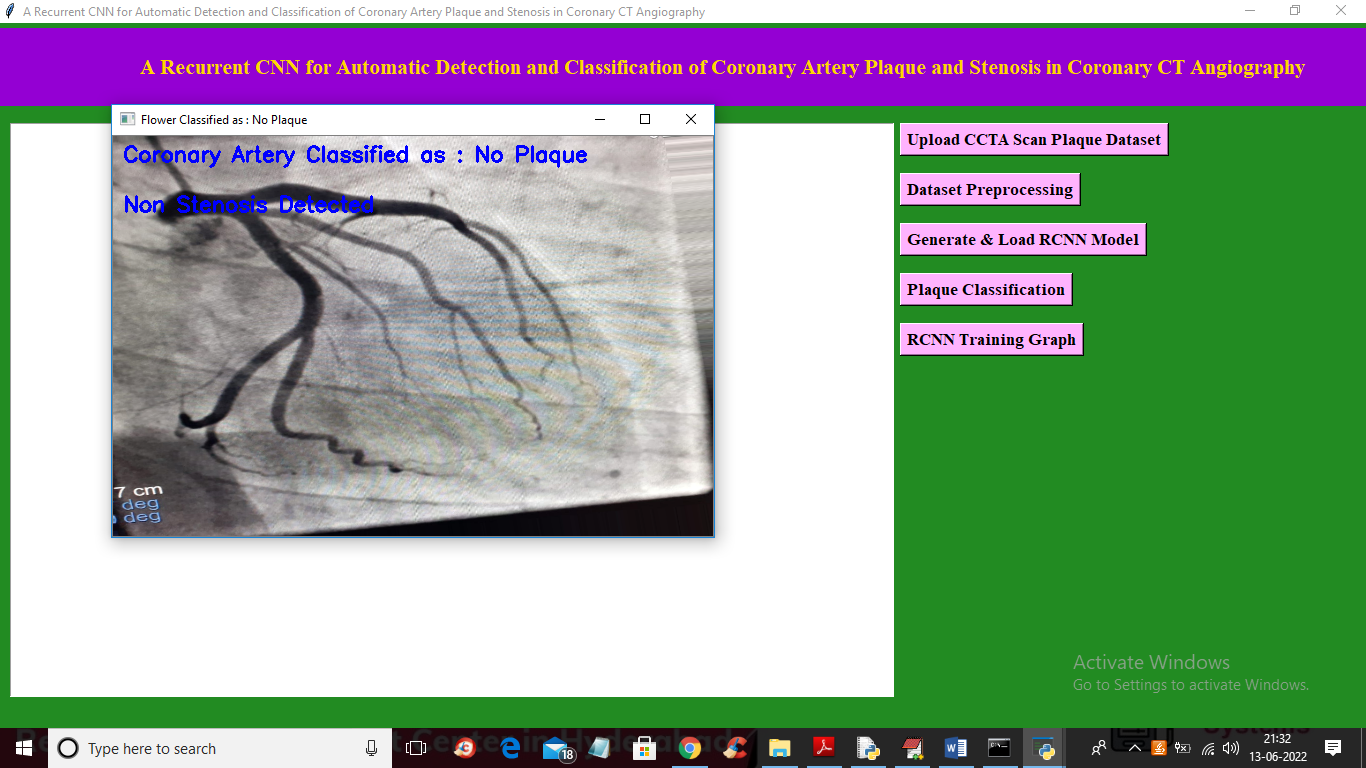
In above screen selecting and uploading ‘0.png’ and then click on ‘Open’ button to get below output



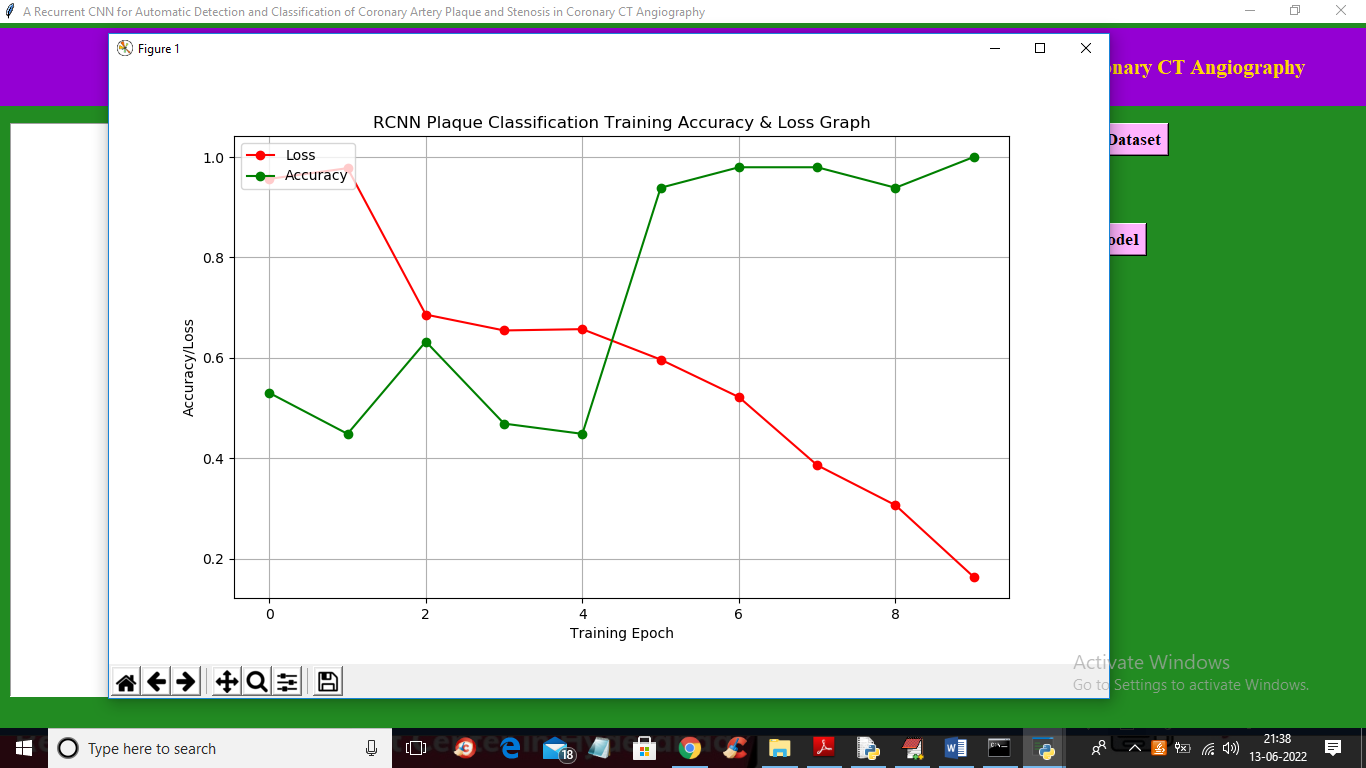
In above screen in image blue colour text we can see classification result as ‘Plaque Detected’ and Stenosis also detected and now upload and test other images



In above screen uploading ‘i1.png’ and then click on ‘Open’ button to get below output



In above screen no plaque detected and similarly you can upload and test other images and now click on ‘RCNN Training Graph’ button to get below graph



In above graph x-axis represents epochs and y-axis represents accuracy and loss values and in above graph red line represents LOSS and green line represents accuracy and in above graph we can see with each increasing epoch accuracy got increase and loss got decrease