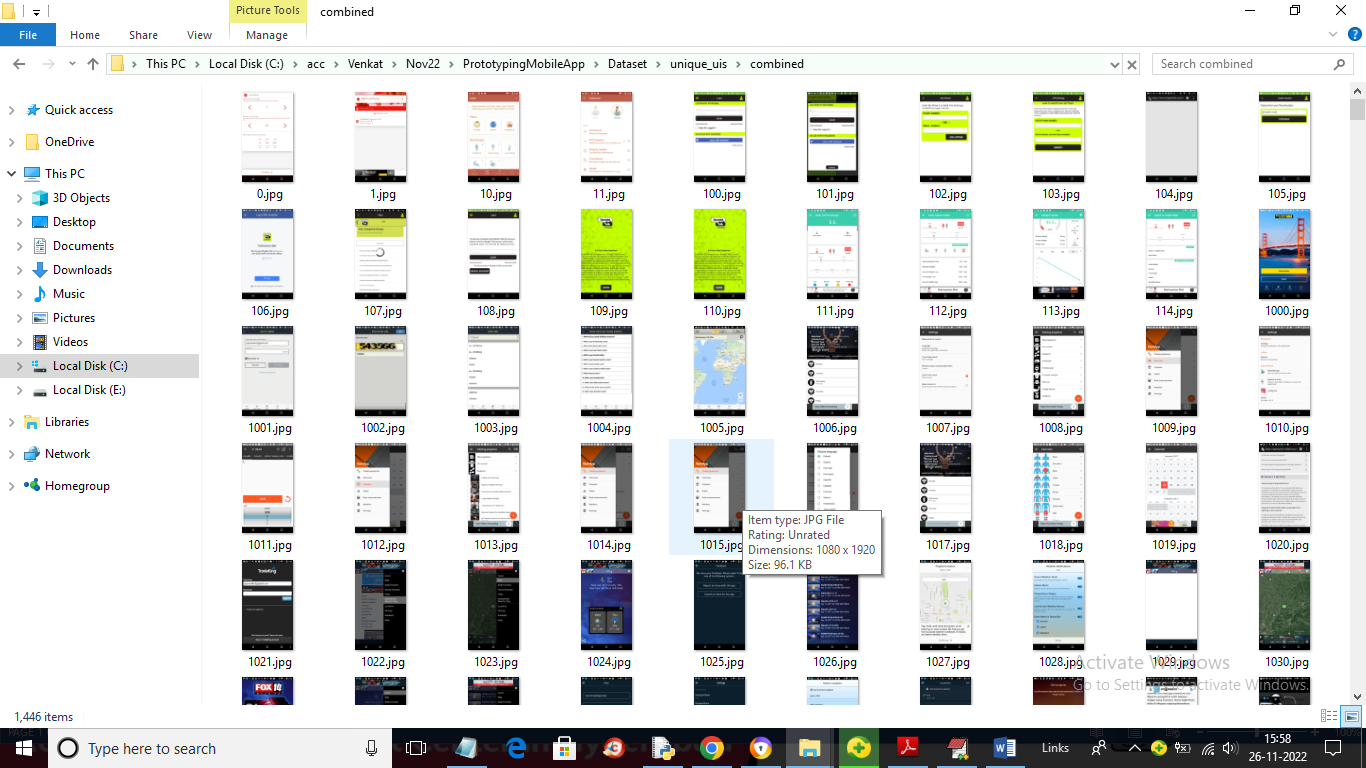
Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps

In propose paper author is applying CNN Neural Network algorithm to predict code for given Android GUI screen. It’s common for developer to generate code for given graphical user interface but this process will take lot of experience and manual work. To overcome from above issue author is training CNN with RICO dataset which consists of CODE in JSON format and GUI images. After training we can apply this CNN model on any android screen to generate or predict code.

Predicted code will be in the form of JSON and we can use below ANDROID APP to convert that JSON code to ANDROID layout

<https://github.com/flipkart-incubator/proteus>

In above link we can see by giving JSON code we can get android code. In below screen I am showing images used to train CNN



Above images are from RICO dataset which can I downloaded from KAGGLE by typing RICO dataset on Google.

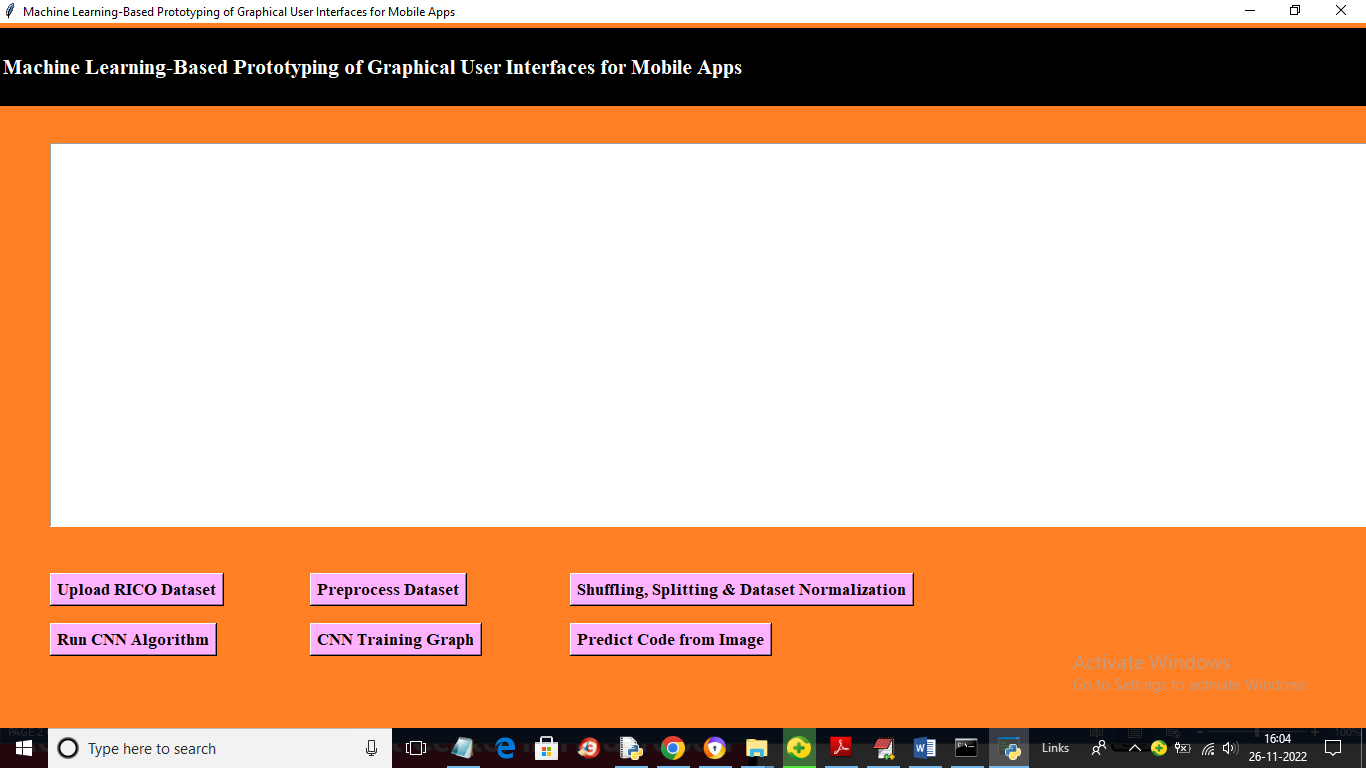
To implement this project we have designed following modules

1. Upload RICO Dataset: using this module we will upload dataset to application
2. Preprocess Dataset: using this module we will read each image and then resize and normalize all pixel values from the image
3. Shuffling, Splitting & Dataset Normalization: using this module we will shuffle dataset and then split dataset into train and test where application will used 80% dataset for training and 20% for testing
4. Run CNN Algorithm: now 80% train data will be input to train CNN and then apply 20% test data to calculate CNN prediction accuracy confusion matrix
5. CNN Training Graph: using this module we will plot CNN training accuracy and loss graph
6. Predict Code from Image: using this module we will upload test GUI screen and then CNN will predict android code in JSON format.

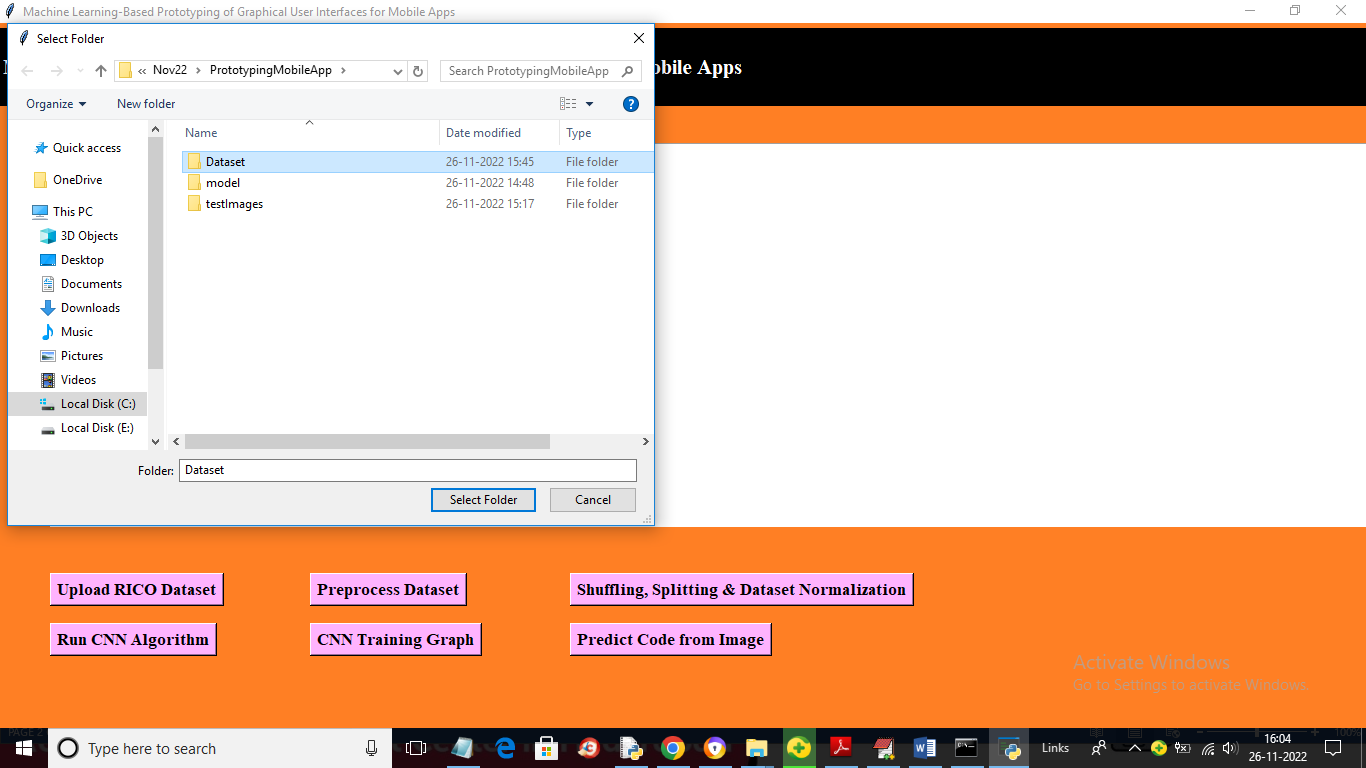
NOTE: in paper also author used RICO dataset and this dataset given training code in JSON format so we can get generated code also in JSON format

SCREEN SHOTS

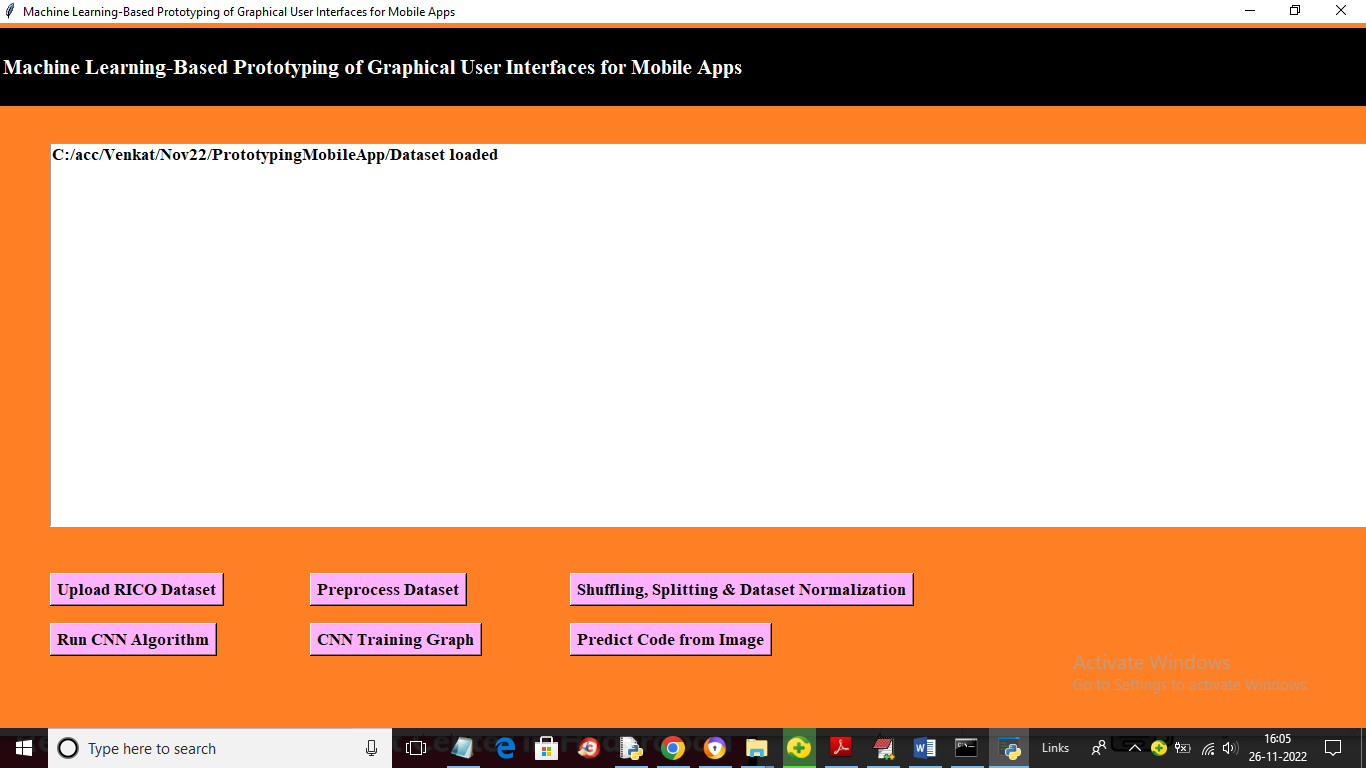
To run project double click on ‘run.bat’ file to get below screen



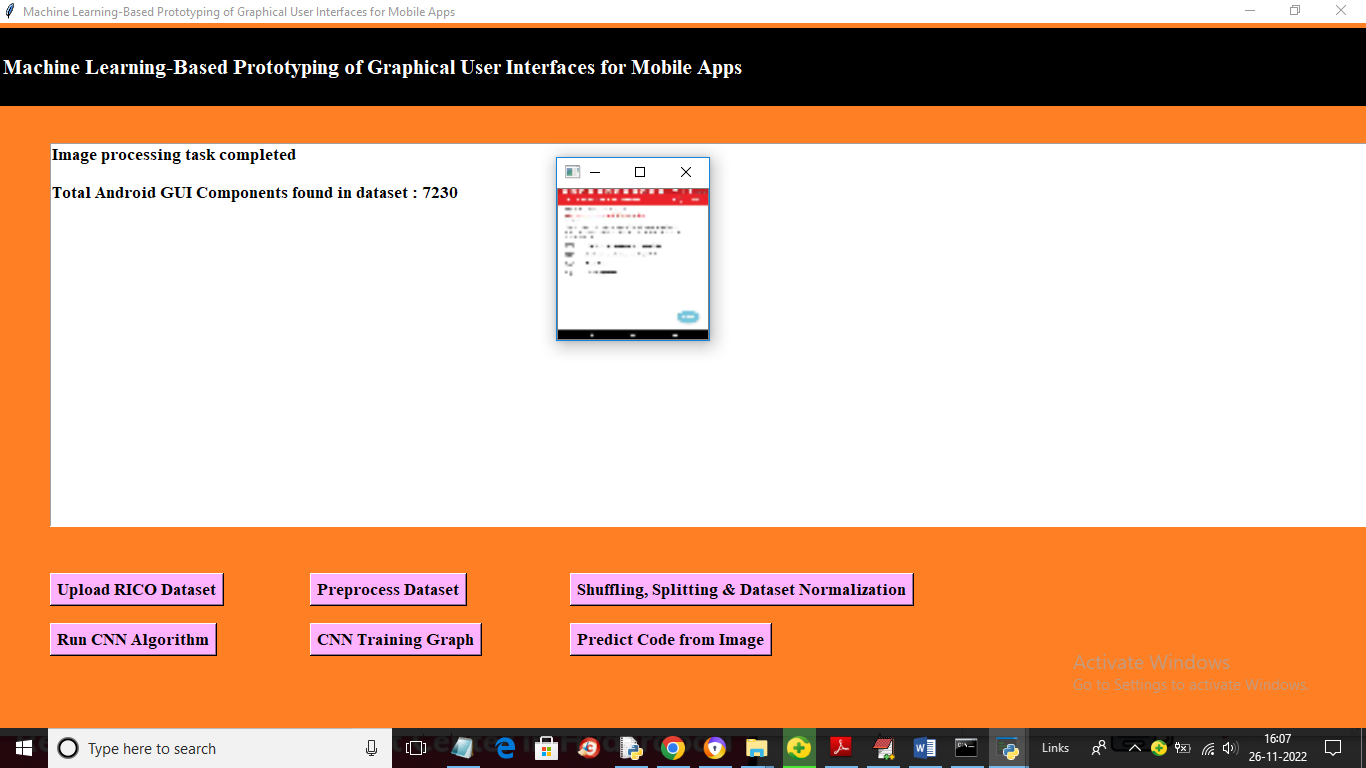
In above screen click on ‘Upload RICO Dataset’ button to upload dataset and get below output



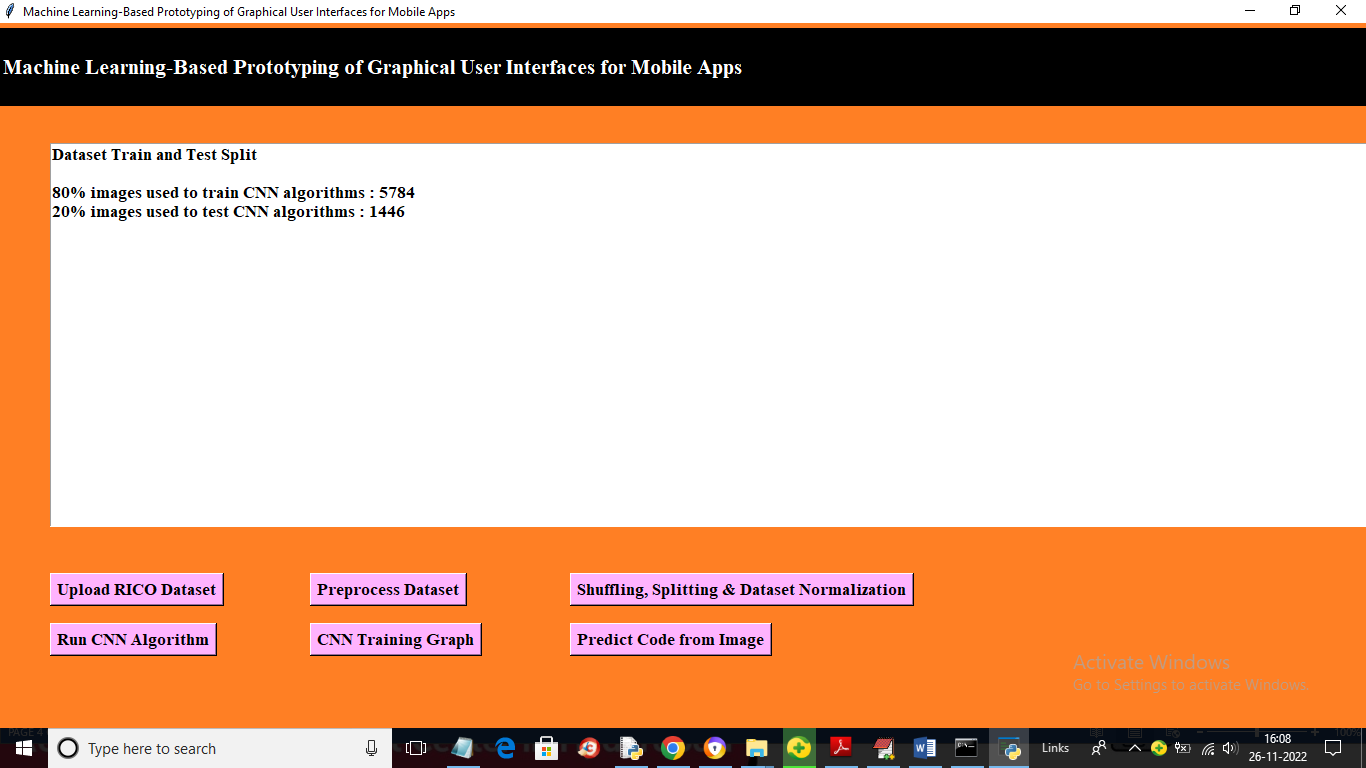
In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



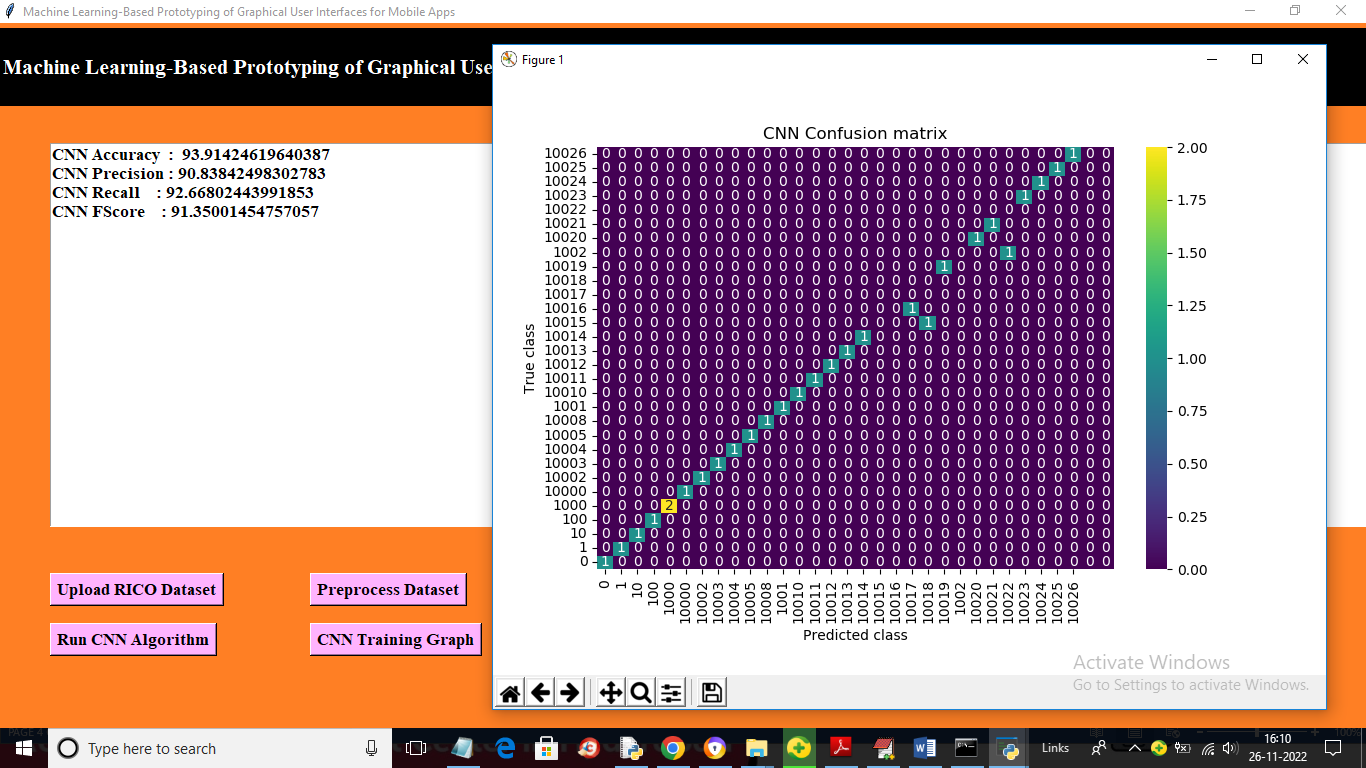
In above screen click on ‘Preprocess Dataset’ button to read and process each image and get below output



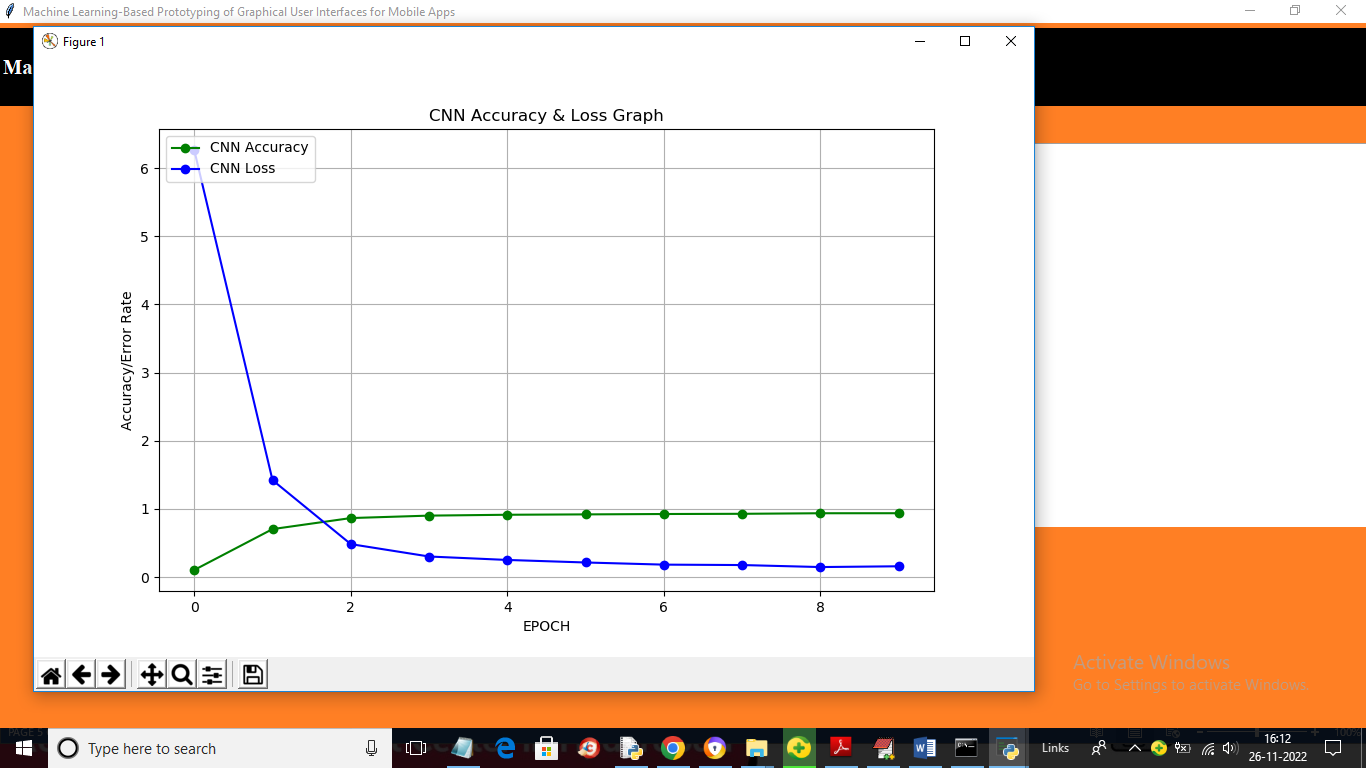
In above screen we can see all 7230 images are processed and loaded and we can see one sample processed image and now close above image and then click on ‘Shuffling, Splitting & Dataset Normalization’ button to shuffle and normalize images and get below output



In above screen we can see application using 5784 images for training and 1446 images for testing and now click on ‘Run CNN Algorithm’ button to get below output

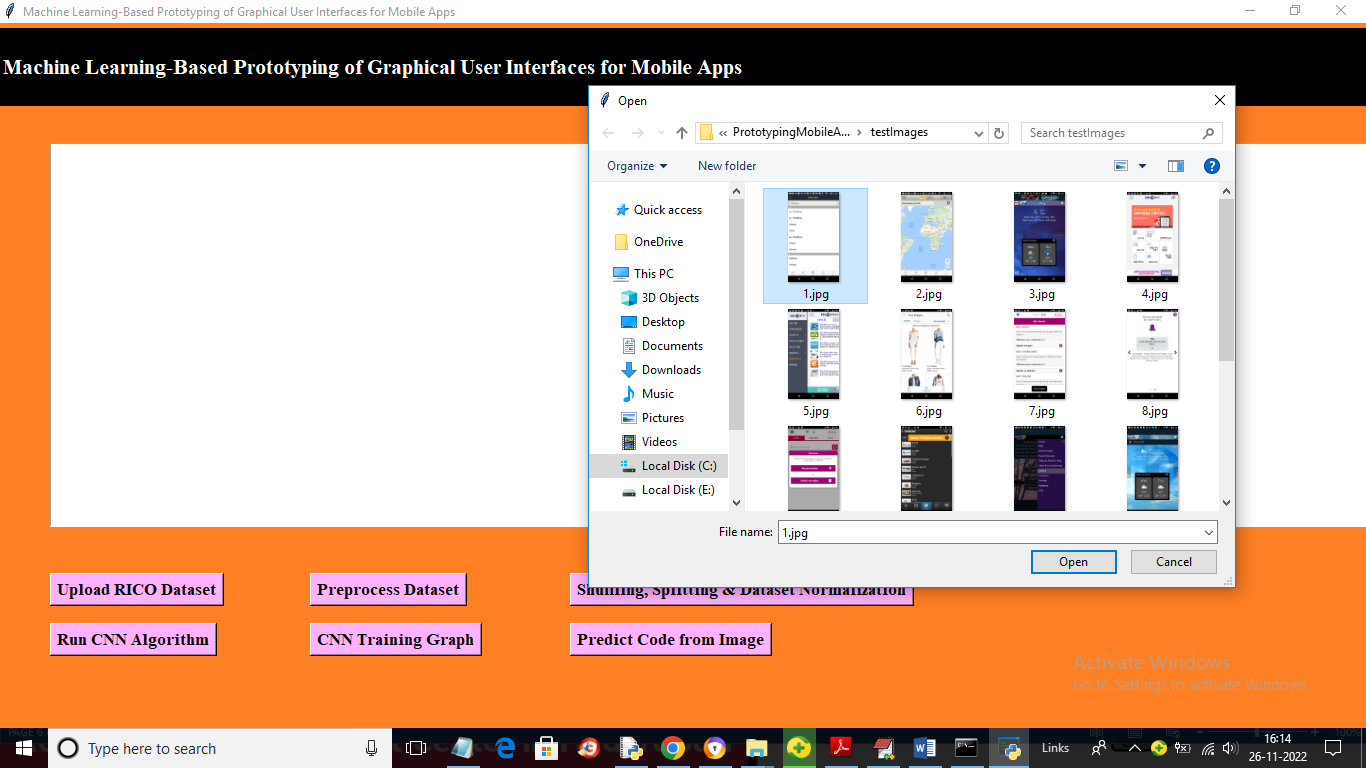


In above screen CNN training completed and we got its accuracy as 93% and then in confusion matrix x-axis represents predicted android CODE ID and y-axis represents TRUE code id and all numbers in blue colour boxes are wrong prediction and numbers in different colour boxes represents correct prediction and we can see CNN predict very few images incorrectly. Now close above image and then click on ‘CNN Training Graph’ button to get below graph

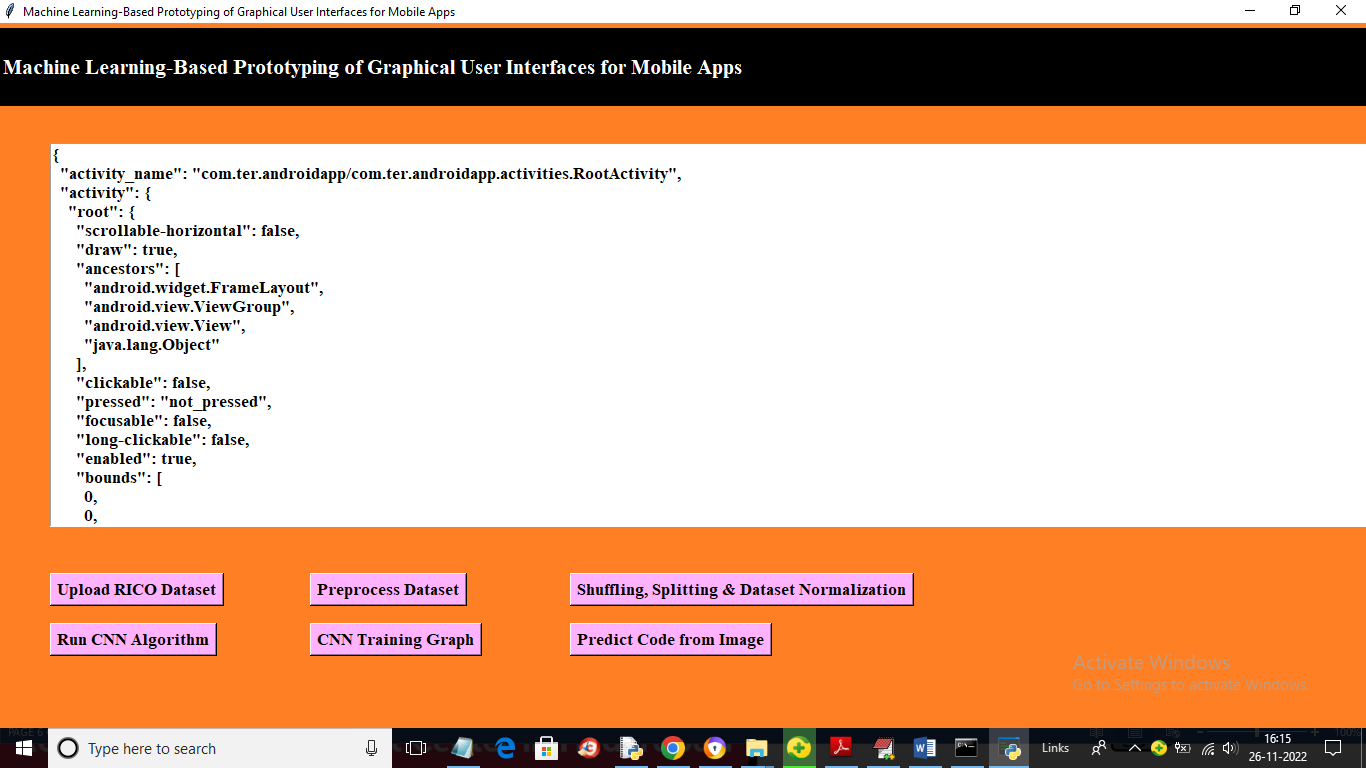


In above graph x-axis represents training EPOCH and y-axis represents accuracy & loss values and green line represents accuracy and blue line represents LOSS and in above graph with each increasing epoch accuracy got increase and reached to 1 and loss got decrease and reached closer to 0. For any CNN model accuracy must increase and loss must get decrease.

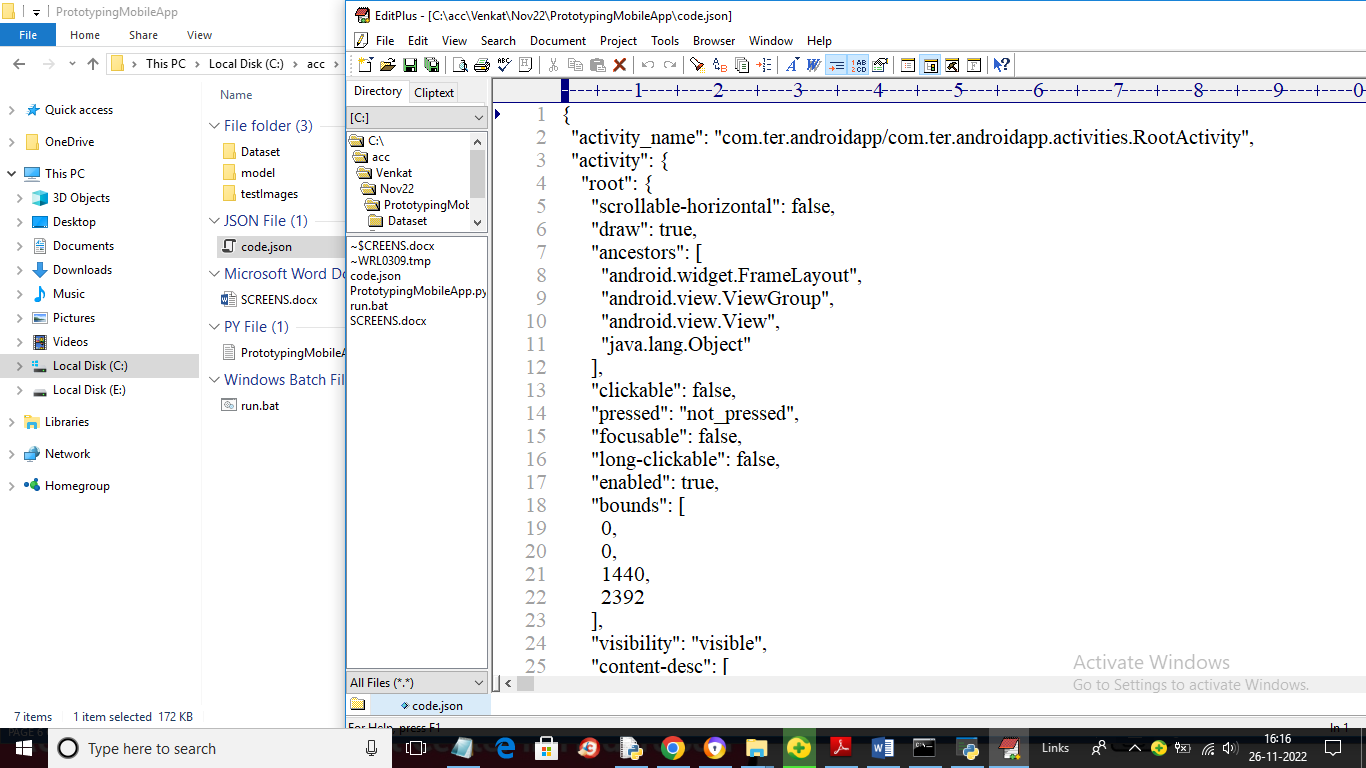
Now close above graph and then click on ‘Predict Code from Image’ button to upload android GUI and get below output



In above screen selecting and uploading 1.jpg and then click on ‘Open’ button to load image and get below predicted code



In above screen we can see predicted code and the same code will saved inside ‘code.json’ file which you can see in below screen



In above screen you can see generated code file and similarly you can upload test image and get code. Another example

