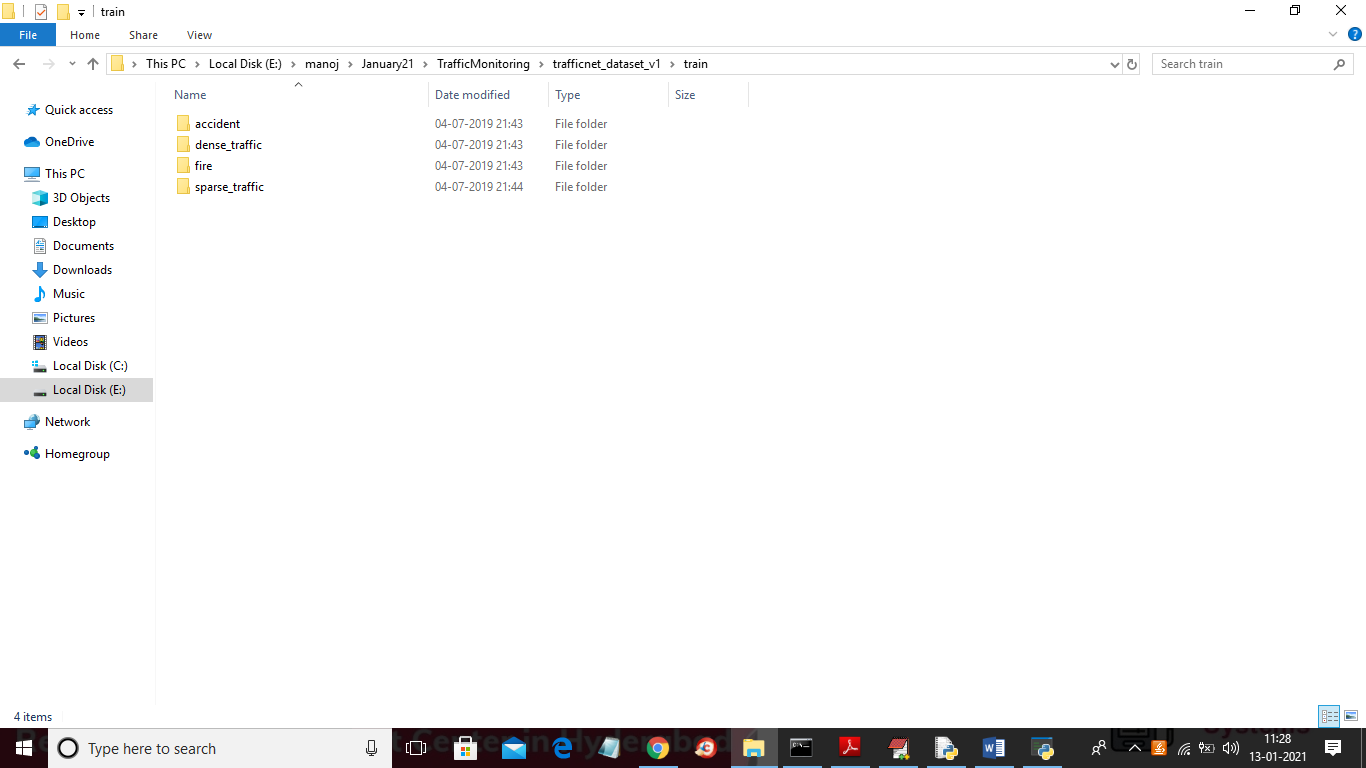
**Road Traffic Condition Monitoring using Deep Learning**

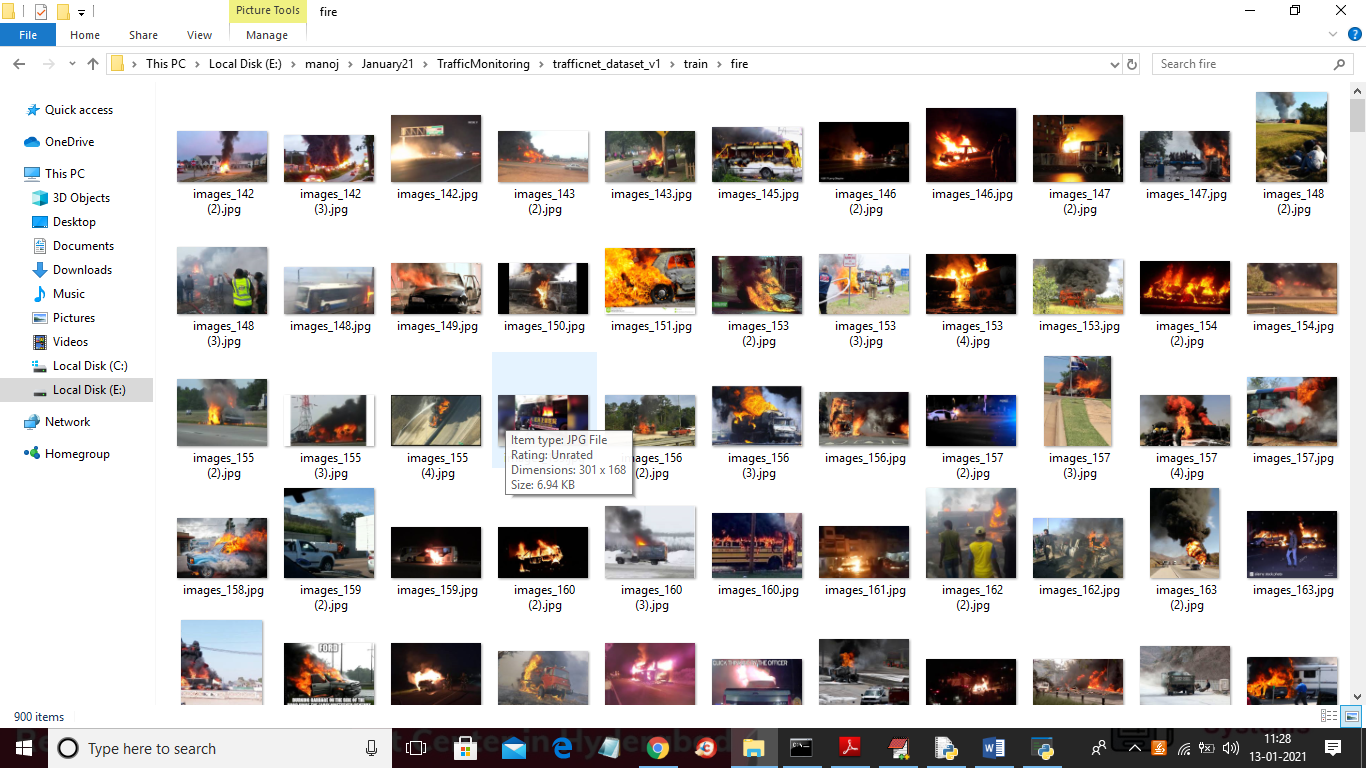
In this project author is using Deep Learning Convolution2D Neural Network to detect traffic conditions such as Heavy Traffic, Low Traffic, Accident or Fire Accident. To implement this project, we have used TRAFFICNET dataset available on GITHUB. We have trained CNN model with TRAFFICNET dataset.

In existing technique to detect traffic Horn’s sound and audio recording technique were used but this technique is not efficient so we need fast processing technique to detect traffic and now a days deep learning is gaining lots of popularity due to its successful detection in less time. So author using CNN deep learning model to detect traffic.

Below is the dataset screen shots which contains various images



In above screen each folder contains its own images



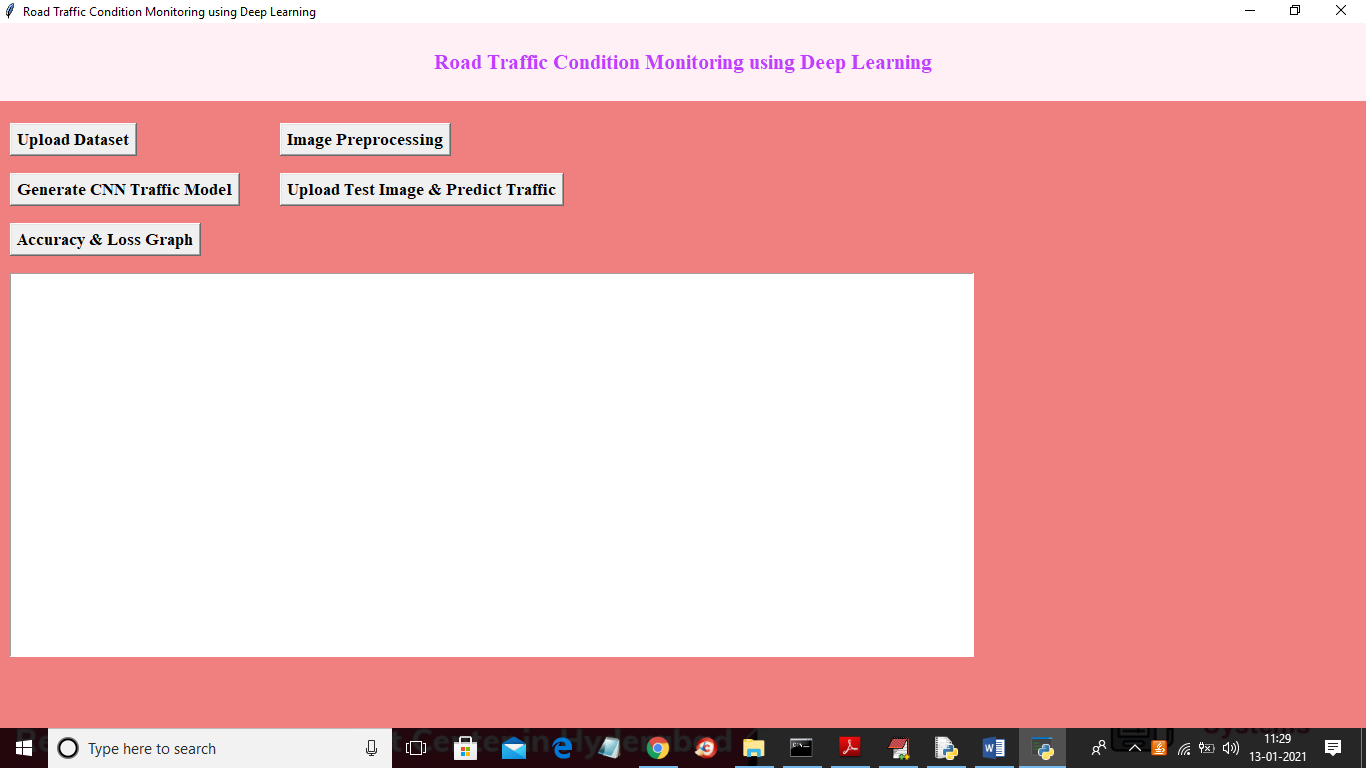
Above screen showing some images from fire accident

This project consists of following modules

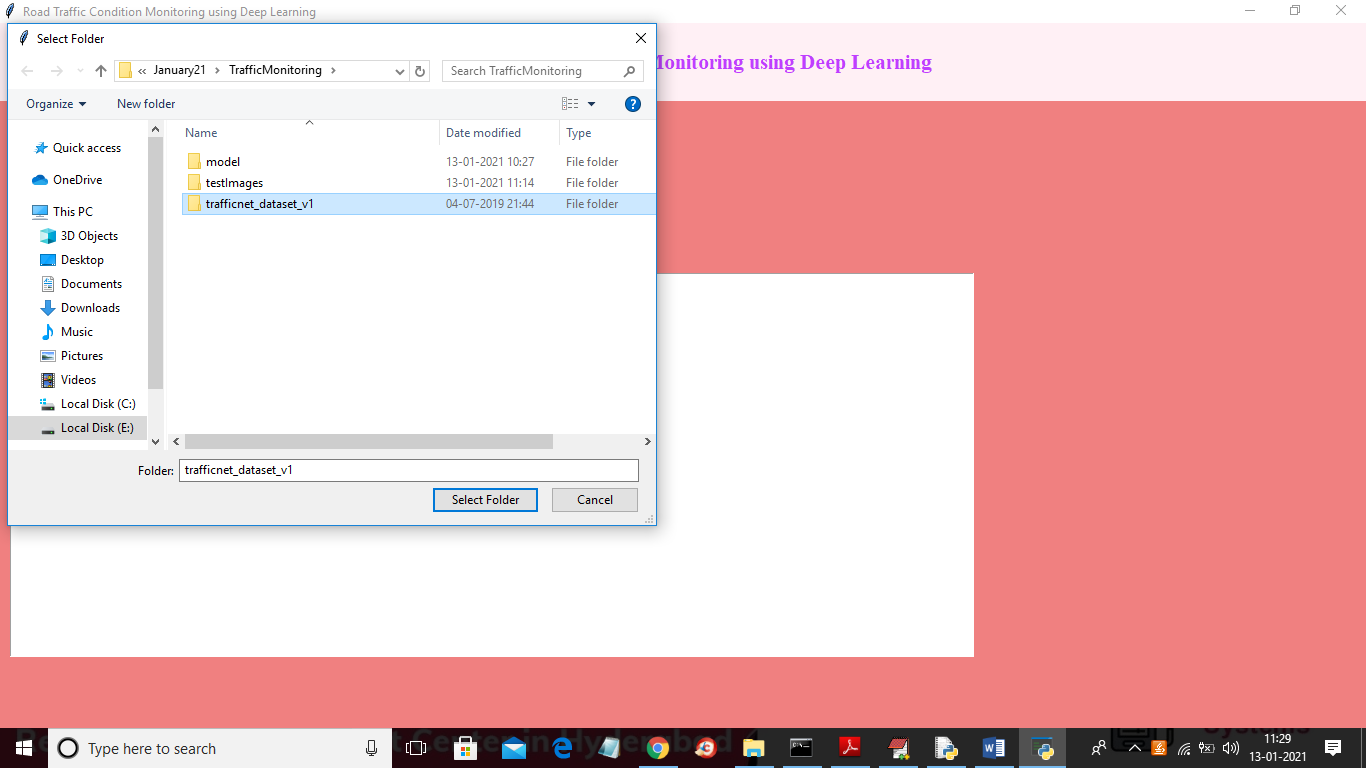
1. Dataset Collection: To implement this project we have downloaded traffic dataset from GITHUB website. <https://github.com/OlafenwaMoses/Traffic-Net/releases/tag/1.0>
2. Preprocessing Image: Using this module we will read images from dataset and then resize all images and then extract pixels from images and convert all images to CNN compatible format
3. CNN Model Generation: using this module we will train CNN model with above processed images
4. Recognition Module: Using this module we will upload traffic image and then CNN model identify whether image contains heavy traffic, low traffic, fire accident and accident.

SCREEN SHOTS

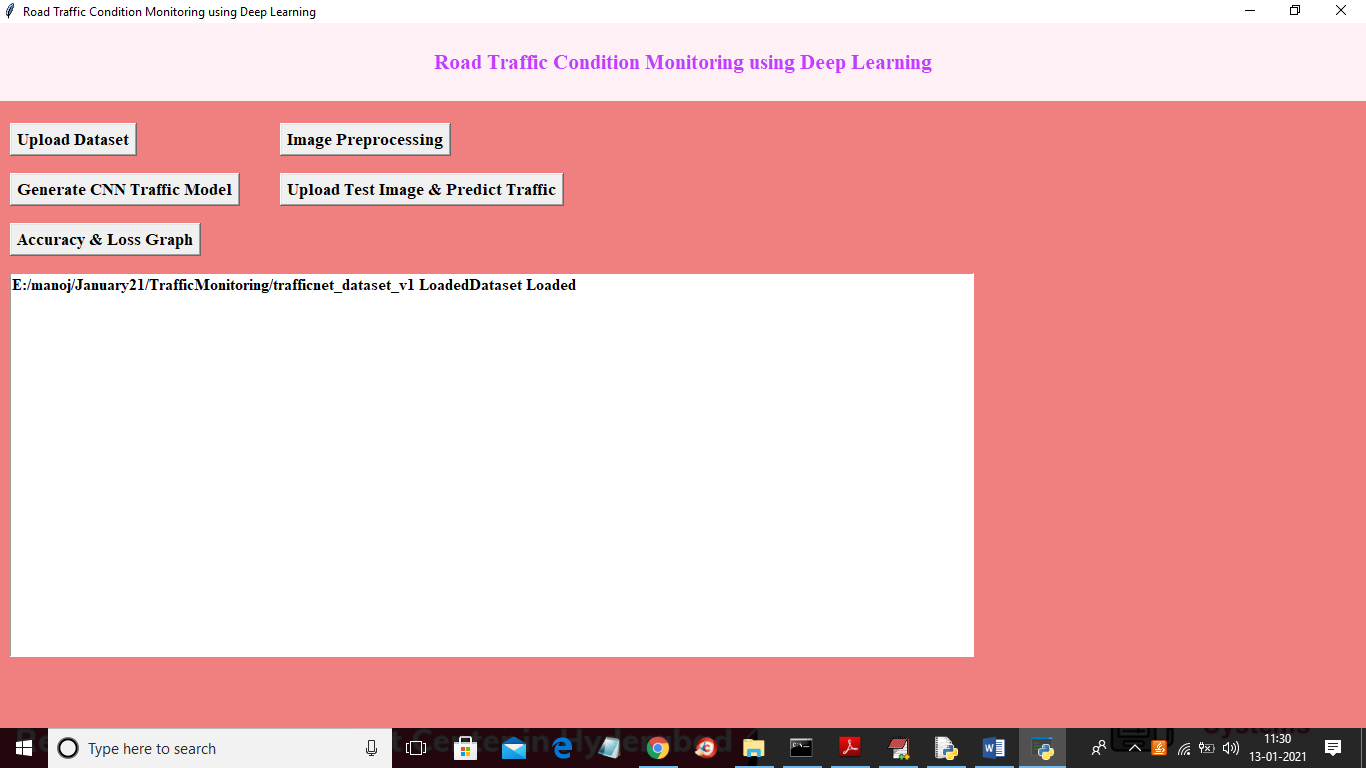
To run project double click on ‘run.bat’ file to get below screen



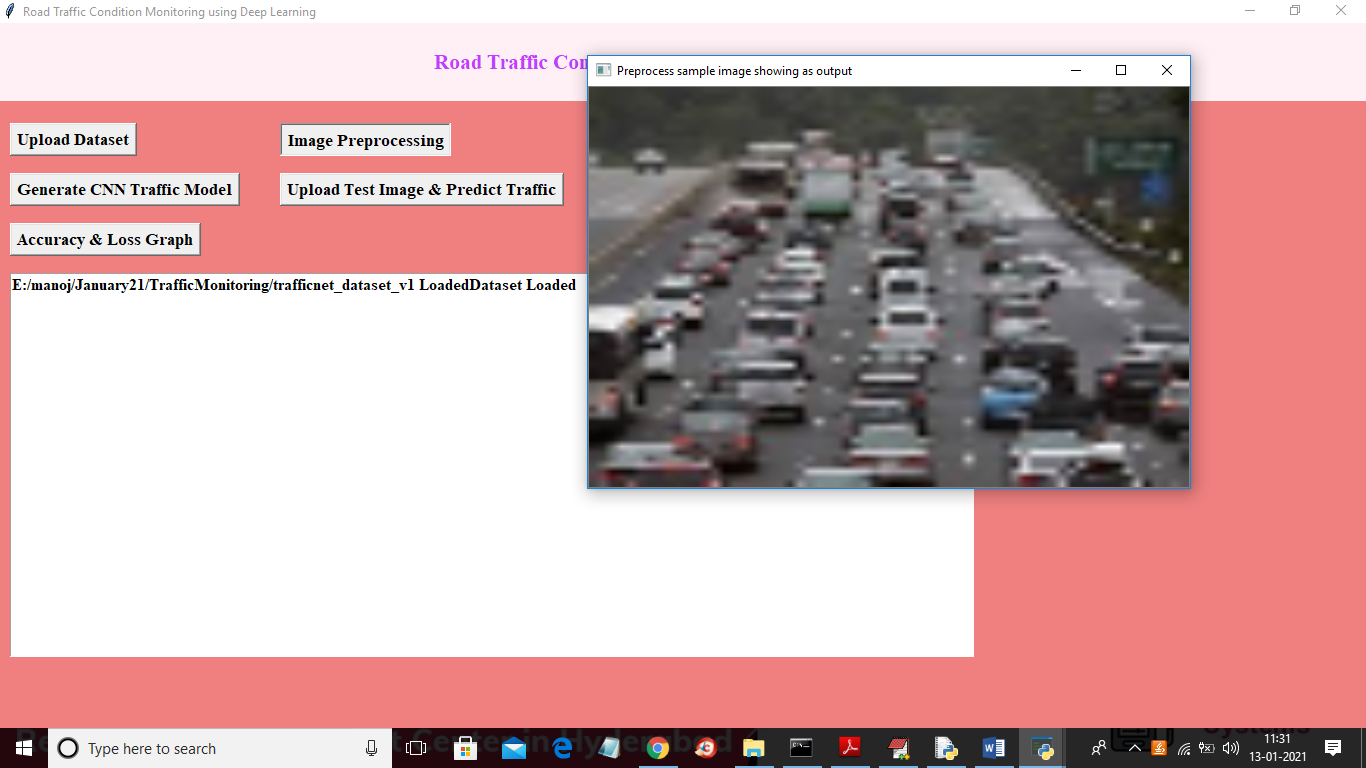
In above screen click on ‘Upload Dataset’ button to load dataset



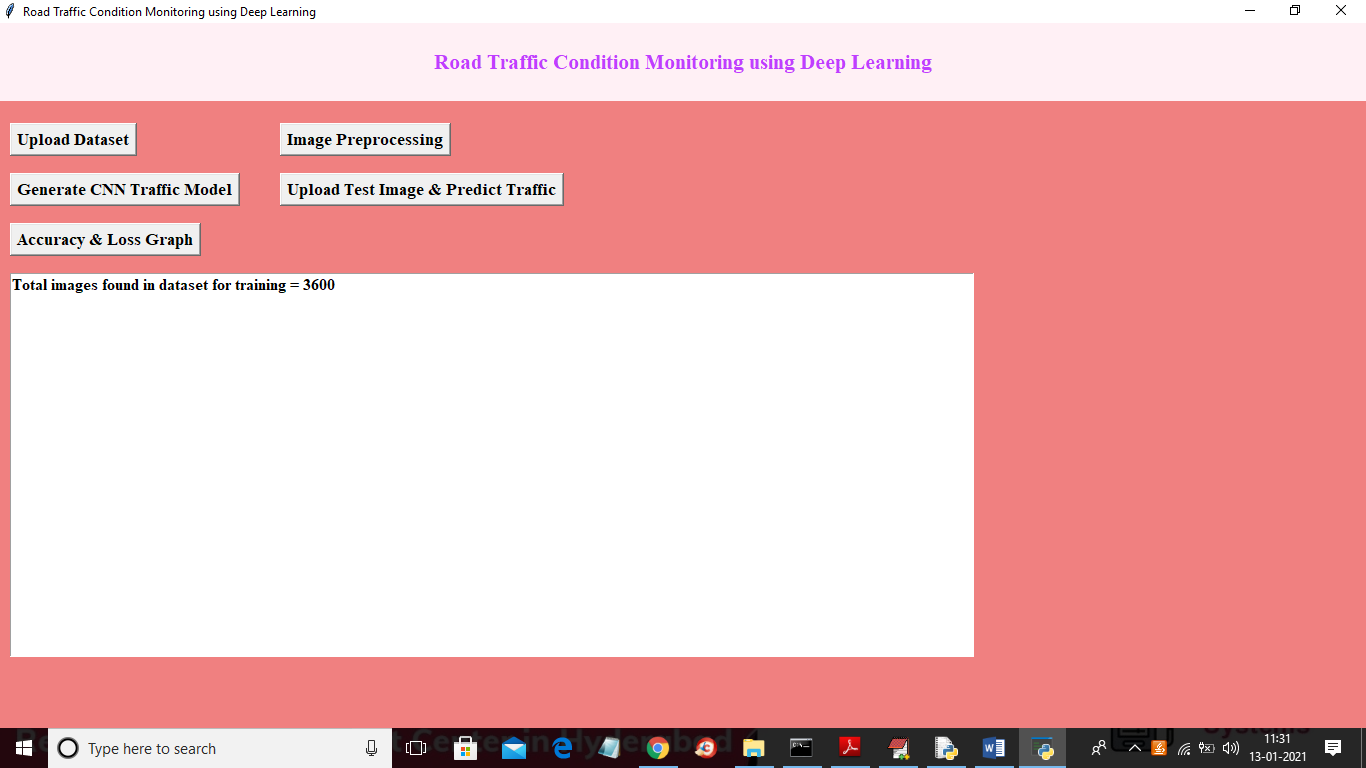
In above screen selecting and uploading ‘trafficnet\_dataset’ folder and then click on ‘Select Folder’ to load dataset and to get below screen



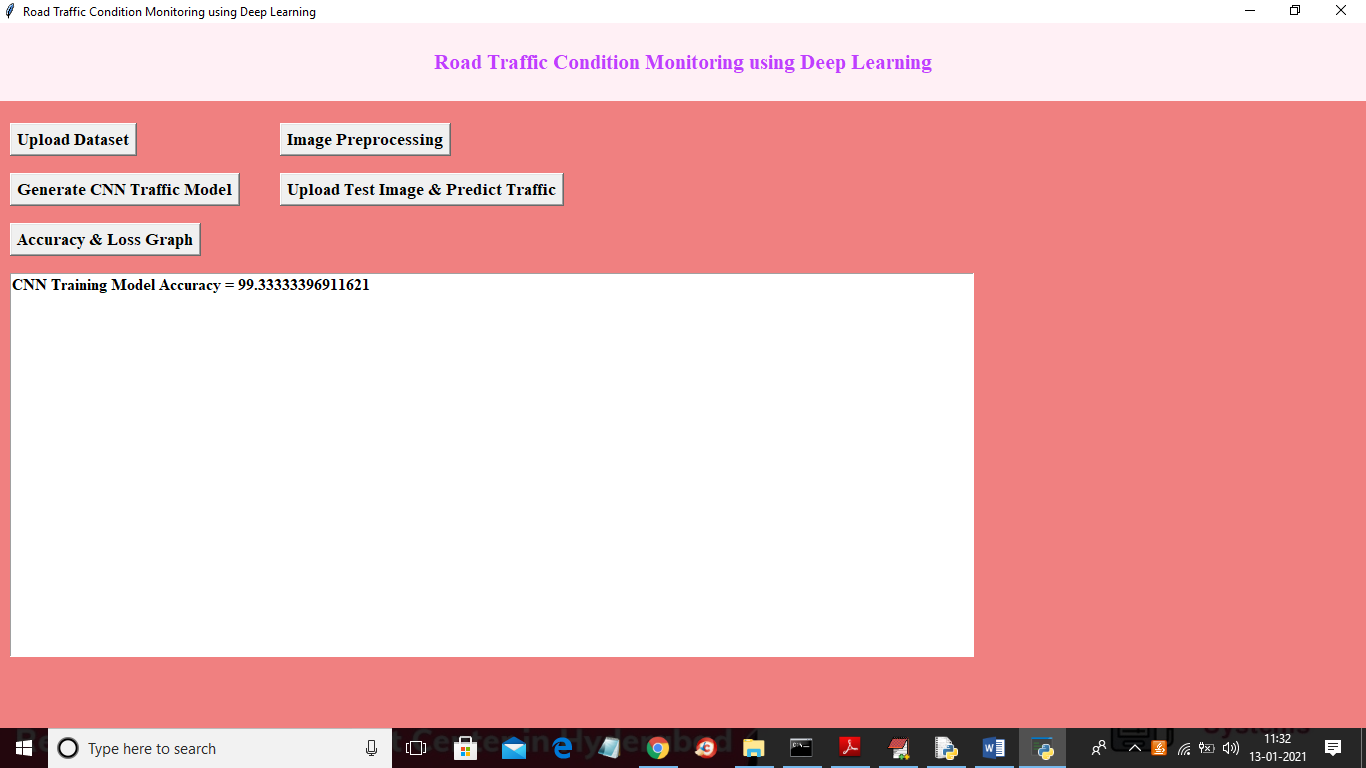
In above screen dataset loaded and then click on ‘Image Preprocessing’ button to read all images and then convert it CNN compatible format



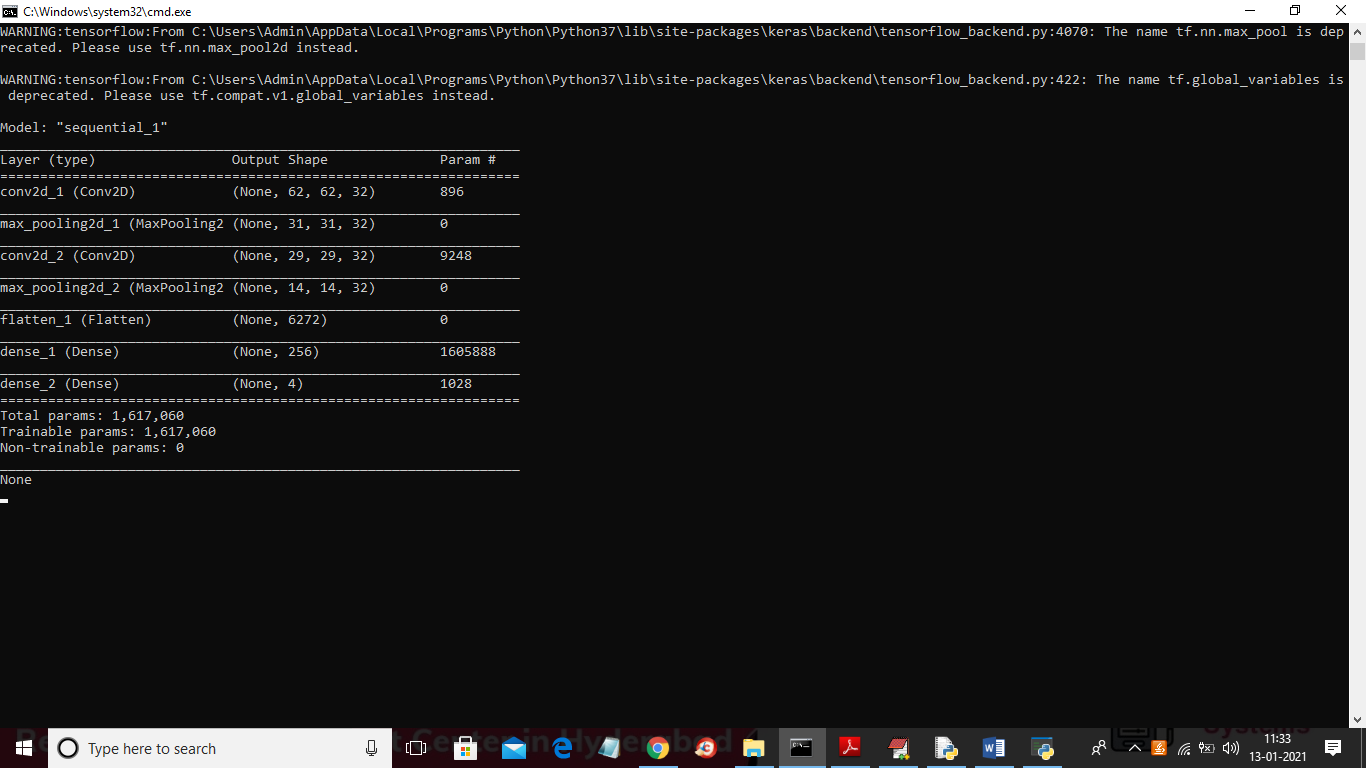
In above screen displaying sample image from processed images and now close above image to get below screen



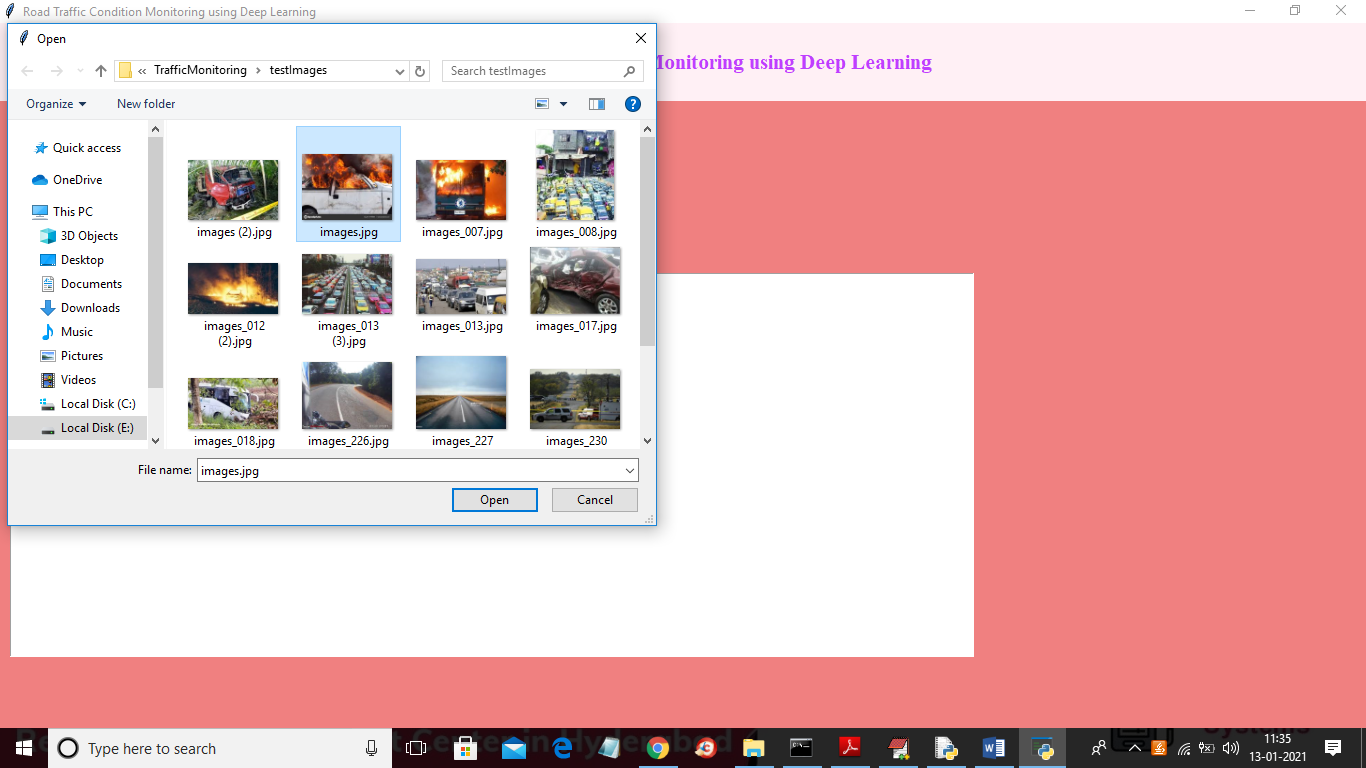
In above screen total processed images showing as 3600 and now click on ‘Generate CNN Traffic Model’ button to generate CNN model and to get below screen



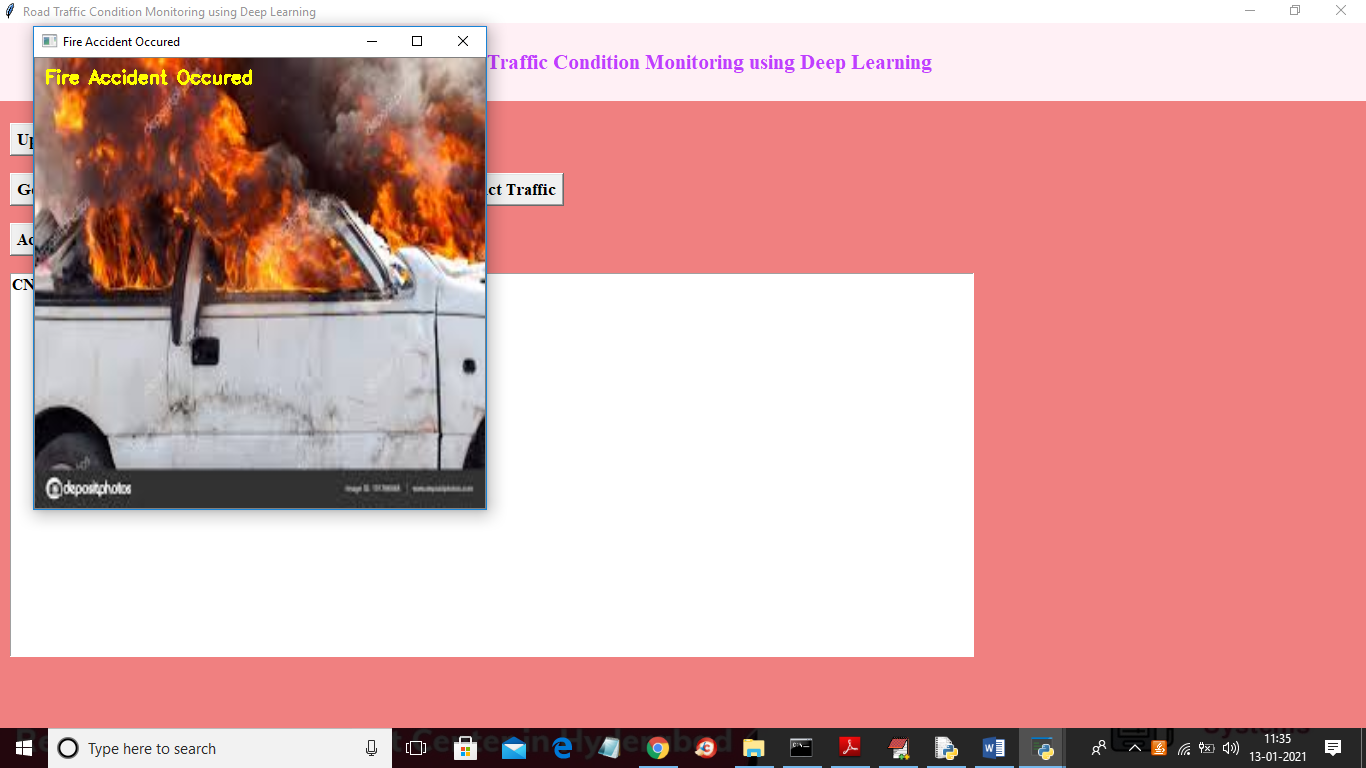
In above screen CNN model generated with prediction accuracy as 99% and in below black console we can see CNN layer details



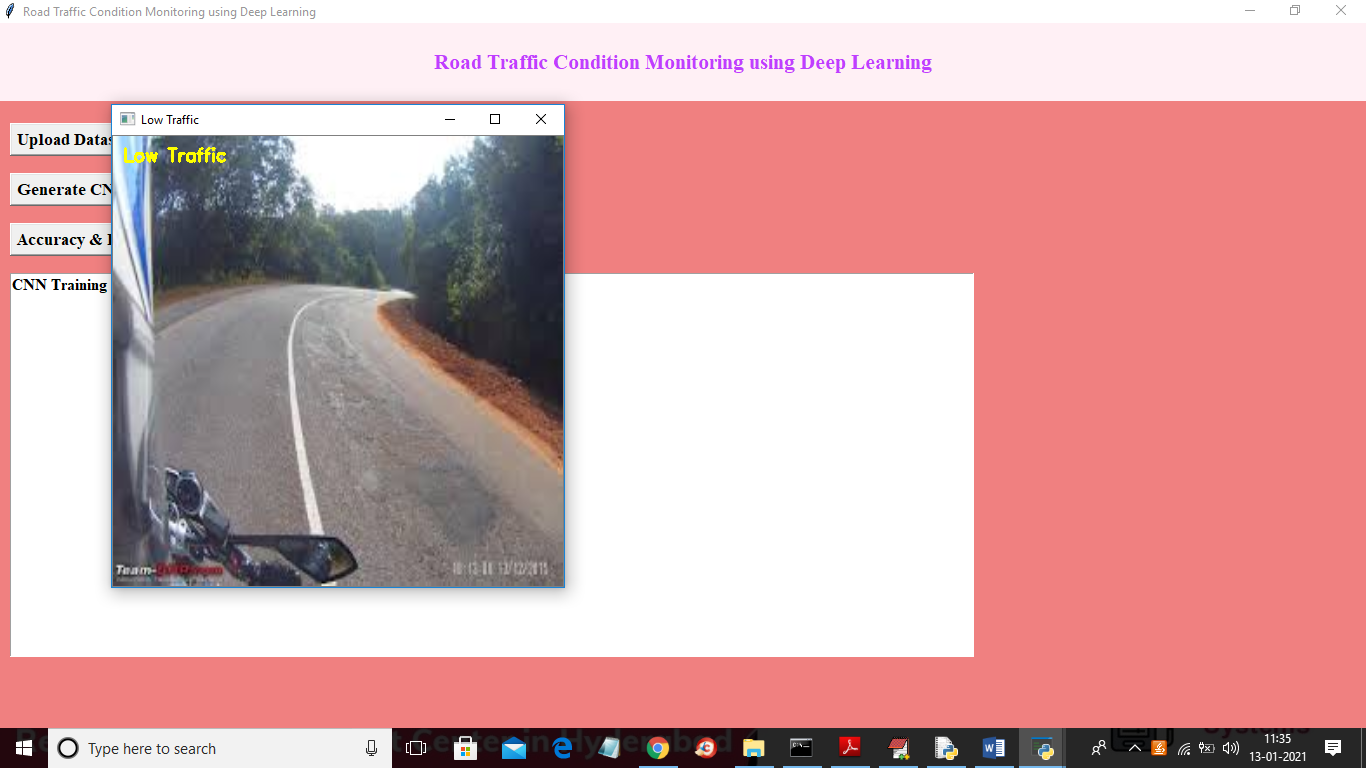
In above screen showing all layers of CNN model and each layer filtering images with different sizes such as 62 X 62, 31 X 31 etc. Now model is ready and now click on ‘Upload Test Image & Predict Traffic’ button and upload image to identify traffic condition

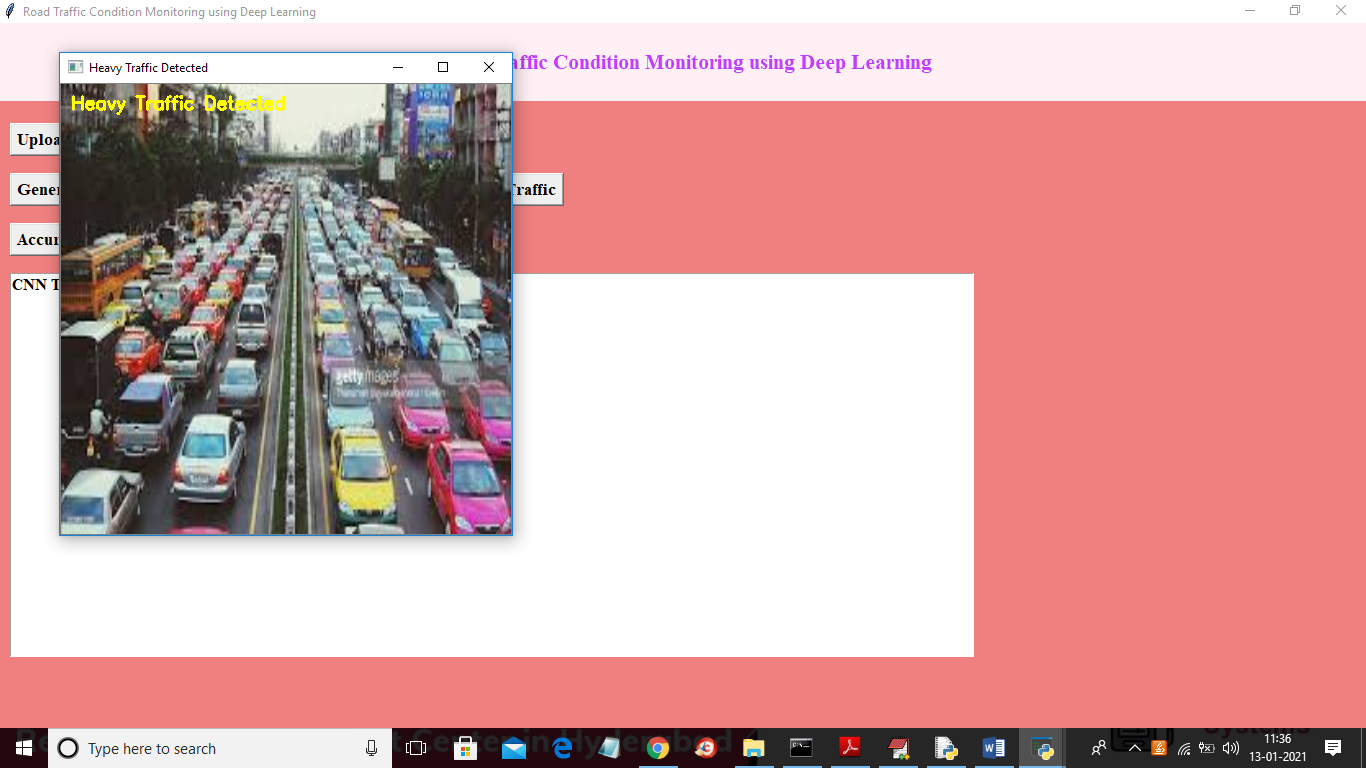


In above screen selecting and uploading ‘images.jpg’ file and then click on ‘Open’ button to get below result



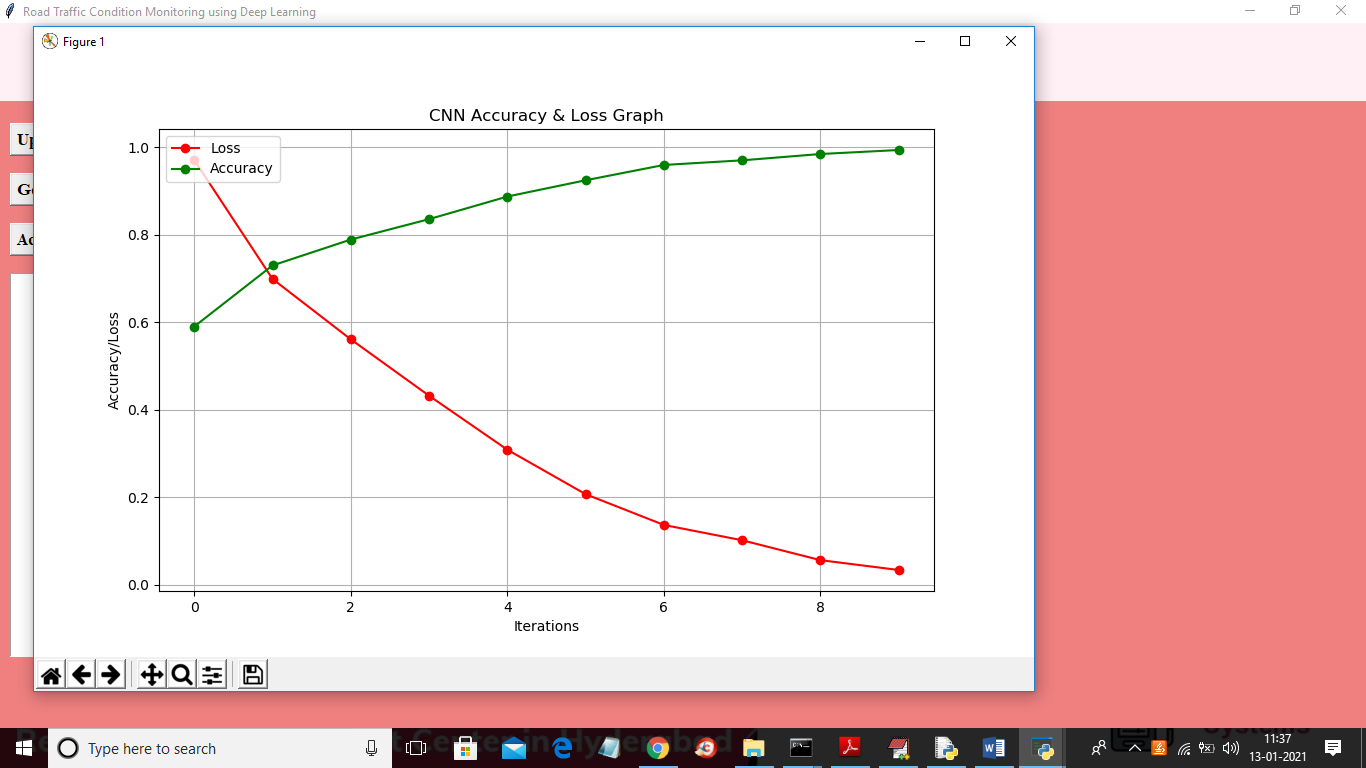
Test another image







Now click on ‘Accuracy & Loss Graph’ button to get below graph



In above graph red line indicate LOSS and green line indicates accuracy and x-axis represents number of iterations or EPOCH and y-axis represents accuracy/loss. In above graph we can see in each iteration accuracy get increase and loss/error rate decrease which means in every iteration CNN model get better