DEEP CNN MODEL FOR CONDITION MONITORING OF ROAD TRAFFIC

A MAJOR PROJECT
Submitted in fulfillment of the award of Degree of Bachelor of Technology in Computer Science and Engineering

Submitted by

SINDHU BUGGANA	V.VARSHITHA	S.VIJAYA
(19RH1A05L2)	(19RH1A05P1)	(19RH1A05K5)

Under the Supervision of

Mr. Vinesh Gone Sir
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
Autonomous Institution, UGC, Govt. of India

Accredited by NBA & NAAC with 'A' Grade
NIRF India Ranking, Accepted by MHRD, Govt. of India
Band A(6-25th) National Ranking by ARIIA, Accepted by MHRD, Govt of India
Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution
Maisammaguda, Dullapally(post), Secunderabad, TELANGANA

December,2022

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
Autonomous Institution, UGC, Govt. Of India
Accredited by NBA & NAAC with 'A' Grade
NIRF India Ranking, Accepted by MHRD, Govt. of India
Band A(6-25th) National Ranking by ARIIA, Accepted by MHRD, Govt of India
Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution
Maisammaguda, Dullapally(post), Secunderabad, TELANGANA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the work embodies in this dissertation entitled "DEEP CNN MODEL FOR CONDITION MONITORING OF ROAD TRAFFIC" being submitted by Sindhu Buggana(19RH1A05L2), Vepachettu Varshitha (19RH1A05P1), S.Vijaya (19RH1A0K5) for the partial fulfillment of the requirement for the award of Bachelor of Technology in **Computer Science and Engineering to Malla Reddy Engineering College For Women, Maisammaguda, Secunderabad** during the academic year 2019–2023 is a record of a bonafide piece of work, undertaken by him/her the supervision of the undersigned.

Approved and Supervised by **Forwarded by**

Signature
Mr.Vinesh Gone **Dr.C.V.P.R.PRASAD**
Asst Prof, Dept of CSE&IT **Professor and HOD**

EXTERNAL EXAMINER

SAK INFORMATICS

CERTIFICATION OF PROJECT COMPLETION

Date: 15-10-2022

This is to certify that the project entitled **"DEEP CNN MODEL FOR CONDITION MONITORING OF ROAD TRAFFIC"** Submitted in fulfilment of the requirements for the degree in **B. Tech (COMPUTER SCIENCE ENGINEERING)** From Malla Reddy Engineering College for Women (UGC -AUTONOMOUS), Maisammaguda, Secunderabad, India is a record of bonafide work carried out by

1. **SINDHU BUGGANA** **(19RH1A05L2)**
2. **VEPACHETTU VARSHITHA** **(19RH1A05P1)**
3. **S. VIJAYA** **(19RH1A05K5)**

Under our supervision and guidance. They displayed analytical capability, had innovation approach to solve problems and produced good result.

For SAK INFORMATICS

Managing Director

Corporate Office: Plot No. 544, F. No. 102, Shiv Sai Murali Ramaneeyam, Pragathi Nagar, Hyderabad, Telangana |
Contact: (0)-9000188676 | E-Mail: mahesh@sakinformatics.com | Web: www.sakinformatics.com

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
Autonomous Institution, UGC, Govt. of India
Accredited by NBA &NAAC with 'A' Grade
NIRF India Ranking, Accepted by MHRD, Govt. of India
Band A(6-25th)National Ranking by ARIIA, Accepted by MHRD, Govt of India
Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution
Maisammaguda, Dullapally(post), Secunderabad, TELANGANA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We **'Sindhu Buggana, Vepachettu Varshitha, S.Vijaya'**, are students of **'Bachelor of Technology in Computer Science and Engineering', session: 2019–2023,** Malla Reddy Engineering College For Women, Maisammaguda, Secunderabad, hereby declare that the work presented in this project work entitled 'Genomic pan cancer classification using Deep Learning' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Sindhu Buggana (19RH1A05L2)

Vepachettu Varshitha (19RH1A05P1)

S.Vijaya (19RH1A05k5)

Date :

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College For Women** and **Department of Computer Science and Engineering** which gave us the opportunity to have expertise in Engineering and profound technical knowledge.

We would like to deeply thank our Honorable Minister of Telangana State **SRI.CH.Malla Reddy Garu**, Founder Chairman MRGI, the largest cluster of institutions in the state of Telangana for providing us with all their sources in the college to make our project success.

We wish to convey gratitude to our principal **Dr. Y. Madhavee Latha**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

We express our sincere gratitude to **Dr.C.V.P.R.Prasad**, Head of the Department of Computer Science and Engineering for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We express our sincere gratitude to **Mrs.A.Radha Rani**, Director of Computer Science and Engineering for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We would like to thank our internal guide **Mr.Vinesh Gone**, and all the faculty members for their valuable guidance and encouragement towards the completion of our project work.

With Regards and Gratitude

Sindhu Buggana(19RH1A05l2)

Vepachettu Varshitha(19RH1A05P1)

s.Vijaya (19RH1A05K5)

ABSTRACT

The traffic surveillance system is accumulated with an enormous amount of data regarding road traffic each and every second. Monitoring these data with the human eye is a tedious process and it also requires manpower for monitoring. Deep learning approach (Convolutional Neural Network) can be utilized for traffic monitoring and control. The traffic surveillance data are pre-processed to construct the training dataset. The Traffic net is constructed by transferring the network to traffic applications and retraining it with self-established data set. This Traffic net can be used for regional detection in large scale applications. Further, it can be implemented across-the-board. The efficiency is admirably verified through speedy discovery in the high accuracy in the case study. The tentative assessment could pull out to its successful application to a traffic surveillance system and has potential enrichment for the intelligent transport system in future.

INDEX

CHAPTER 1

INTRODUCTION

1.1 Overview

As urbanization has accelerated, traffic in urban areas has increased significantly, and a similar phenomenon has appeared in freeways connected to the urban areas. The real-time monitoring of traffic on freeways could provide sophisticated traffic information to drivers, so the drivers could choose alternative routes to avoid heavy traffic. Furthermore, long-term records of traffic monitoring will help develop efficient transportation policies and strategies across urban and suburban areas. Currently, the typical means of monitoring traffic information use closed circuit television (CCTV) or detection equipment. The detection equipment includes a loop detector, image detectors, dedicated short-range communication (DSRC), and radar detectors. In general, CCTVs are installed at fixed locations, and they can monitor the area on the freeway 24 hours a day. CCTV can monitor only limited areas; therefore, multiple CCTV circuits are necessary to monitor a wide range of freeways. However, the installation and maintenance of multiple CCTV circuits are costly. In addition, it is difficult to detect vehicles in CCTV videos automatically due to the overlapping between vehicles because CCTV usually captures freeways in an oblique direction.

Recently, to overcome the limitations of collecting traffic information through CCTV, video collection methods employing unmanned aerial vehicles (UAVs) are being used. Unlike CCTV, a UAV can monitor a wide range of freeways by elevating its altitude or moving its location, and it can travel to a specific location to observe unexpected situations, such as traffic accidents. Furthermore, a UAV views the freeways in a perpendicular direction, so the vehicles in the recorded videos do not overlap. Currently, however, videos from installed CCTV or operated UAVs are monitored by humans. Therefore, as the number of CCTV circuits and UAVs increases, more human resources are required. Moreover, we can not avoid human error; it is highly demanding to analyse real-time videos to effectively monitor traffic information.

Several methods have been developed to automatically analyse traffic conditions on freeways using videos from CCTV or dash cam. In, the vehicle was detected using Mask RCNN from the surveillance video taken with a fixed camera, and the vehicle speed was calculated. In, a vehicle was detected in the image using Ada-Boost, which uses multiple weak classifiers to construct a strong classifier. Recently, as artificial intelligence technology has rapidly advanced, road image analysis methods using deep learning are also

being proposed. A deep learning-based object detection method, Faster R-CNN, was used to detect vehicles in images.

There have also been various attempts to automatically measure traffic from road videos taken by UAVs. Most of these methods consist of object detection techniques for capturing vehicles in images and object tracking techniques for identifying the movements of detected vehicles, and the speed of vehicles are calculated at the end. In, various types of vehicles were detected from UAV video using Yolo v3. In, the vehicle speed was calculated from the results of tracking the vehicle using the moving average of the previous frame and the Kalman filter. A Haar-like feature-based cascade structure is used to detect the location and size of a vehicle in the image with a bounding box, and the convolutional neural network (CNN) method was applied to the detection results to improve the final classification performance. Traffic volume was also calculated by tracking the movement of the vehicle using the KLT-optical.

In contrast to CCTV videos taken at a fixed height, the altitude of UAV varies at every time the video is recorded, and sometimes the altitude of UAV changes during recording. If the image scales are not fixed, we are not able to estimate the vehicle's traveling distance on the actual road by simply measuring moving distance of the vehicle in sequential images. Therefore, to determine the exact speed of a vehicle by tracking the vehicle in sequential images, the image scale of each image should be estimated and the changes in the image scale should be taken into account. For example, the scale of the image was obtained by comparing a pre-defined structure on an actual road with its corresponding object in the first frame of a video. This approach requires a pre-definition of a structure for each location; therefore, images without known structures cannot be utilized. Later, the image scale is calculated by comparing the average sizes of vehicles in the images and pre-measured and averaged actual vehicle size in. Although these methods have somewhat resolved the restrictions associated with a UAV's flight area, the calculated image scale is not accurate because the size of vehicle varies depend on the types of the vehicles. For instance, a detected vehicle can include sedans, vans, buses, of trucks.

MOTIVATION

Deep learning approach (Convolutional Neural Network) can be utilized for traffic monitoring and control. The traffic surveillance data are pre-processed to construct the training dataset. The Traffic net is constructed by transferring the network to traffic applications and retraining it with self-established data set.

1.2 PROBLEM DEFINITION

The Existing method requires a huge amount of Hardware equipments deployed to the Road. Moreover, they are very sensitive to the external noise and environmental conditions. It is more accurate when processing a limited number of vehicles but it does not work well on large scale dataset.

1.3 OBJECTIVE OF PROJECT

The purpose of the proposed work is to make a speedy traffic detection system which reduces the man power and detected Multiclass problems namely fire detection, accident detection, dense and sparse traffic detection. The main aim is to classify the given input image to dense or sparse based on the trained model from the input dataset.

CHAPTER 2

LITERATURE SURVEY

2.1 Survey on Evolving Deep Learning Neural Network Architectures

AUTHORS: Abul Bashar

The deep learning being a subcategory of the machine learning follows the human instincts of learning by example to produce accurate results. The deep learning performs training to the computer frame work to directly classify the tasks from the documents available either in the form of the text, image, or the sound. Most often the deep learning utilizes the neural network to perform the accurate classification and is referred as the deep neural networks; one of the most common deep neural networks used in a broader range of applications is the convolution neural network that provides an automated way of feature extraction by learning the features directly from the images or the text unlike the machine learning that extracts the features manually. This enables the deep learning neural networks to have a state of art accuracy that mostly expels even the human performance. So the paper is to present the survey on the deep learning neural network architectures utilized in various applications for having an accurate classification with an automated feature extraction.

2.2 A Novel Online Dynamic Temporal Context Neural Network Framework for the Prediction of Road Traffic Flow

AUTHORS: Zoe Bartlett et al

Traffic flow exhibits different magnitudes of temporal patterns, such as short-term (daily and weekly) and long-term (monthly and yearly). Existing research into road traffic flow prediction has focused on short-term patterns; little research has been done to determine the effect of different long-term patterns on road traffic flow prediction. Providing more temporal contextual information through the use of different temporal data segments could improve prediction results. In this paper, we have investigated different magnitudes of temporal patterns, such as short-term and long-term, through the use of different temporal data segments to understand how contextual temporal data can improve prediction. Furthermore, to learn temporal patterns dynamically, we have proposed a novel online dynamic temporal context neural network framework. The framework uses different temporal data segments as input features, and during online learning, the updating scheme dynamically determines how useful a temporal data segment (short and long-term temporal patterns) is for prediction, and weights it accordingly for use in the regression model.

Therefore, the framework can include short-term and relevant long-term patterns in the regression model leading to improved prediction results. We have conducted a thorough experimental evaluation with a real dataset containing daily, weekly, monthly and yearly data segments. The experiment results show that both short and long-term temporal patterns improved prediction accuracy. In addition, the proposed online dynamical framework improved predication results by 10.8% when compared with a deep gated recurrent unit model.

Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNN architecture TrafficNet

Detection of traffic congestion is important for route guidance using in intelligent transport system (ITS) to prevent jam escalation. Although the surveillance system has been used in freeway for years, it is hard to automatically identify and report traffic congestion in complicated transportation scene according to various illumination, weather and other disturbances. The detection process based on human eye is time-consuming and tedious as the machine detection accuracy is not high enough to meet the requirements of practical applications. In this paper, a new classifier is proposed using convolutional neural networks (CNN) to generate four TrafficNet based on two championships of ILSVRC including AlexNet and VGGNet. Instead of using fully-connected layers in AlexNet and VGGNet, a support vector machine (SVM) are used after CNN architecture. Congestion and non-congestion images are trained and tested through this new structure. Image database with more than 30000 images are extracted from existing traffic surveillance video and corresponding labels are added manually. With database, those TrafficNet are trained and tested, detection accuracy and training time of those TrafficNet are compared. The experimental results show that the accuracy of proposed method can reach up to 90%, which is much higher than traditional method based on feature extraction without deep learning

2.4 **LSTM network:** A deep learning approach for short-term traffic forecast

AUTHORS: Weihai Chen,Zheng Zhao,Jingmeng Liu and Peter C. Y. Chen

Short-term traffic forecast is one of the essential issues in intelligent transportation system. Accurate forecast result enables commuters make appropriate travel modes, travel routes, and departure time, which is meaningful in traffic management. To promote the forecast accuracy, a feasible way is to develop a more effective approach for traffic data analysis. The availability of abundant traffic data and computation power emerge in recent years, which motivates us to improve the accuracy of short-term traffic forecast via deep learning approaches. A novel traffic forecast model based on long short-term memory (LSTM) network is proposed. Different from conventional forecast models, the proposed LSTM network considers temporal-

spatial correlation in traffic system via a two-dimensional network which is composed of many memory units. A comparison with other representative forecast models validates that the proposed LSTM network can achieve a better performance.

2.5 Deep visual tracking: Review and experimental comparison

AUTHORS: Peixia Li,Dong Wang,Lijun Wang and Huchuan Lu

Recently, deep learning has achieved great success in visual tracking. The goal of this paper is to review the state-of-the-art tracking methods based on deep learning. First, we introduce the background of deep visual tracking, including the fundamental concepts of visual tracking and related deep learning algorithms. Second, we categorize the existing deep-learning-based trackers into three classes according to network structure, network function and network training. For each categorize, we explain its analysis of the network perspective and analyze papers in different categories. Then, we conduct extensive experiments to compare the representative methods on the popular OTB-100, TC-128 and VOT2015 benchmarks. Based on our observations, we conclude that: (1) The usage of the convolutional neural network (CNN) model could significantly improve the tracking performance. (2) The trackers using the convolutional neural network (CNN) model to distinguish the tracked object from its surrounding background could get more accurate results, while using the CNN model for template matching is usually faster. (3) The trackers with deep features perform much better than those with low-level hand-crafted features. (4) Deep features from different convolutional layers have different characteristics and the effective combination of them usually results in a more robust tracker. (5) The deep visual trackers using end-to-end networks usually perform better than the trackers merely using feature extraction networks. (6) For visual tracking, the most suitable network training method is to per-train networks with video information and online fine-tune them with subsequent observations. Finally, we summarize our manuscript and highlight our insights, and point out the further trends for deep visual tracking.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Manual Identification and Classification of the vehicle is a tedious process. So, Two Automatic Identification and Classification Algorithms namely Artificial Neural Network (ANN) and KNearest Neighbour (KNN) are used to detect and classify the vehicles using Acoustic signals which are recorded by microphones attached to the roads. The Manual prediction process has complexity in alone twofold narrow road with assorted traffic. This complexity is rectified by using the automatic detection algorithm (ANN & KNN). ANN & KNN is also used as a classifier and this broadly classifies the vehicles into three categories namely Heavy traffic, Medium traffic and Low traffic. Microphones are used to detect the vehicles by recording the acoustic signals. It records the signal when a vehicle passes over the microphone and shows a peak in energy as a result of recording. The vehicle is automatically detected by locating the energy peak by smoothing the energy contour. The ANN/KNN classifiers are trained using the feature vectors. With the help of the Test data that contains over 180 vehicles, the efficiency of the method is tested with diverse categories.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

1.The CNN is like-minded to precisely categorize the object because it learns the features while the network gathers the images rather than pre-training

2. The dynamic temporal prediction is used to determine the traffic flow in different condition.

3.2 PROPOSED SYSTEM:

An Intelligent Traffic Detection system is proposed to traffic condition (dense or sparse) from the input image and approaches it as a multi-class problem (fire, accident, dense and sparse traffic). The proposed approach uses CNN algorithm which efficiently classifies the input image. The local buffer station stores the initial classification of congestion images. The spatial and temporal values are summed up with the end result.

ADVANTAGES OF PROPOSED SYSTEM:

1.The proposed method overcomes the shortcomings of the traditional method. In this digital world deep learning paves the way for fleet management.

2.Rather than the existing method, the CNN can solve Multiclass problems. This has speedy detection and provides timely information to the public.

3.3 SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows
- **Coding Language**: Python 3.7

3.3 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY

- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4
PROPOSED SYSTEM

The traffic surveillance system is accumulated with an enormous amount of data regarding road traffic each and every second. Monitoring these data with the human eye is a tedious process and it also requires manpower for monitoring. Deep learning approach (Convolutional Neural Network) can be utilized for traffic monitoring and control. The traffic surveillance data are pre-processed to construct the training dataset. The Traffic net is constructed by transferring the network to traffic applications and retraining it with self-established data set. This Traffic net can be used for regional detection in large scale applications. Further, it can be implemented across-the-board. The efficiency is admirably verified through speedy discovery in the high accuracy in the case study. The tentative assessment could pull out to its successful application to a traffic surveillance system and has potential enrichment for the intelligent transport system in future.

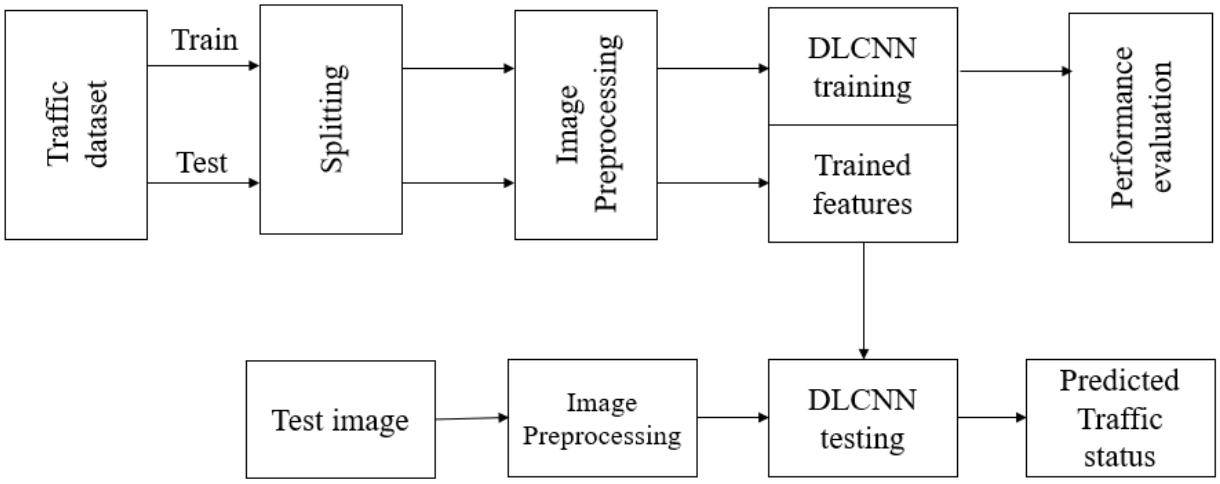


Figure 4.1 Proposed Method

Figure 4.1 shows the block diagram of proposed method. Initially, TrafficNet dataset is spitted into 80% for training and 20% for testing. Then, dataset preprocessing operation is performed to normalize the entire dataset. The image preprocessing operation converts the all the images into uniform size. Further, DLCNN is used for prediction of traffic status i.e., dense traffic, low traffic, accident, and fire occurred from test sample. The performance evaluation is carried out to show supremacy of proposed method.

4.1 Traffic condition dataset

The input dataset of dissimilar classes is gathered from the net. The assessment of output class is set next to the obtained dataset. Four folders namely sparse_traffic, dense_traffic, fire, accident, every folder contains 900 images are generated for training and validation purposes. The folder name represents the class value for classifying output.

4.2 Image pre-processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our AI-Computer Vision models can benefit from this improved data to work on. To train a network and make predictions on new data, our images must match the input size of the network. If we need to adjust the size of images to match the network, then we can rescale or crop data to the required size.

we can effectively increase the amount of training data by applying randomized augmentation to data. Augmentation also enables to train networks to be invariant to distortions in image data. For example, we can add randomized rotations to input images so that a network is invariant to the presence of rotation in input images. An augmented Image Datastore provides a convenient way to apply a limited set of augmentations to 2-D images for classification problems.

we can store image data as a numeric array, an ImageDatastore object, or a table. An ImageDatastore enables to import data in batches from image collections that are too large to fit in memory. we can use an augmented image datastore or a resized 4-D array for training, prediction, and classification. We can use a resized 3-D array for prediction and classification only.

There are two ways to resize image data to match the input size of a network. Rescaling multiplies the height and width of the image by a scaling factor. If the scaling factor is not identical in the vertical and horizontal directions, then rescaling changes the spatial extents of the pixels and the aspect ratio.

Cropping extracts a subregion of the image and preserves the spatial extent of each pixel. We can crop images from the center or from random positions in the image. An image is nothing more than a two-dimensional array of numbers (or pixels) ranging between 0 and 255. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically.

Resize image:

In this step-in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. After that, we then create a function called processing that just receives the images as a parameter.

Need of resize image during the pre-processing phase, some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.

4.4 Proposed ResNet-CNN

Deep neural network is gradually applied to the identification of crop Traffic conditions and insect pests. Deep neural network is designed by imitating the structure of biological neural

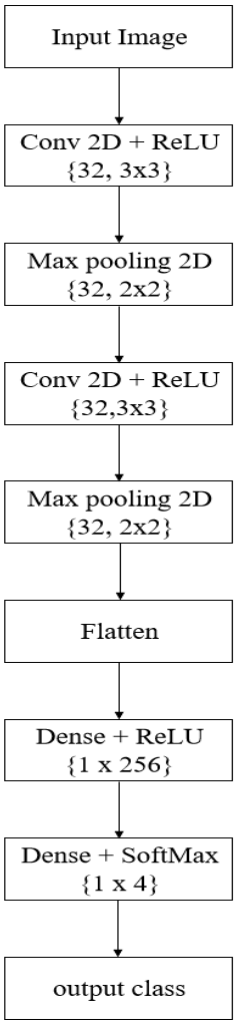


Fig. 2: Proposed ResNet-CNN

Table.1: Layers description.

Layer Names	No. of filters	Kernel size	Feature size
Conv 2D +ReLU	32	3 x 3	62x62x32
Max pooling 2D	-	3 x 3	31x31x32
Conv 2D+ReLU	32	3 x 3	29x29x32
Max pooling 2D	-	3 x 3	14x14x32
Flatten	-	1x6272	1x6272
Dense +ReLU		1 x 256	1 x 256
Dense + SoftMax		1 x 4	1 x 4

network, an artificial neural network to imitate the brain, using learnable parameters to replace the links between neurons. Convolutional neural network is one of the most widely used deep neural network structures, which is a branch of feed forward neural network. The success of AlexNet network model also confirms the importance of convolutional neural network model. Since then, convolutional neural networks have developed vigorously and have been widely used in financial supervision, text and speech recognition, smart home, medical diagnosis, and other fields.

Convolutional neural networks are generally composed of three parts. Convolution layer for feature extraction. The convergence layer, also known as the pooling layer, is mainly used for feature selection. The number of parameters is reduced by reducing the number of features. The full connection layer carries out the summary and output of the characteristics. A convolution layer is consisting of a convolution process and a nonlinear activation function ReLU. A typical architecture of CNN model for crop Traffic condition recognition is shown in Figure 2.

The leftmost image is the input layer, which the computer understands as the input of several matrices. Next is the convolution layer, the activation function of which uses ReLU. The pooling layer has no activation function. The combination of convolution and pooling layers can be constructed many times. The combination of convolution layer and convolution layer or convolution layer and pool layer can be

very flexibly, which is not limited when constructing the model. But the most common CNN is a combination of several convolution layers and pooling layers. Finally, there is a full connection layer, which acts as a classifier and maps the learned feature representation to the sample label space.

Convolutional neural network mainly solves the following two problems.

1) Problem of too many parameters: It is assumed that the size of the input picture is $50 * 50 * 3$. If placed in a fully connected feedforward network, there are 7500 mutually independent links to the hidden layer. And each link also corresponds to its unique weight parameter. With the increase of the number of layers, the size of the parameters also increases significantly. On the one hand, it will easily lead to the occurrence of over-fitting phenomenon. On the other hand, the neural network is too complex, which will seriously affect the training efficiency. In convolutional neural networks, the parameter sharing mechanism makes the same parameters used in multiple functions of a model, and each element of the convolutional kernel will act on a specific position of each local input. The neural network only needs to learn a set of parameters and does not need to optimize learning for each parameter of each position.

2) Image stability: Image stability is the local invariant feature, which means that the natural image will not be affected by the scaling, translation, and rotation of the image size. Because in deep learning, data enhancement is generally needed to improve performance, and fully connected feedforward neural is difficult to ensure the local invariance of the image. This problem can be solved by convolution operation in convolutional neural network.

4.4 ResNet-CNN

According to the facts, training and testing of ResNet-CNN involves in allowing every source image via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1]. Figure 1 discloses the architecture of ResNet-CNN that is utilized in proposed methodology for CBIR system for enhanced feature representation of word image over conventional retrieval systems.

Convolution layer as depicted in Figure 4.3 is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here $d = 3$, since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(k_x, k_y, d)$.

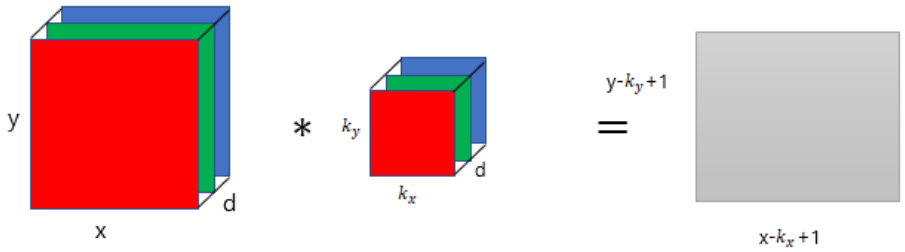


Fig. 3: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\left((x-k_x+1),(y-k_y+1),1\right)$, which is referred as feature map. An example of convolution procedure is demonstrated in Figure 5.2. Let us assume an input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values as given in Figure 3.

1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

(a)

1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

=

3	3	3
2	2	3
3	2	3

Feature map

(b)

Fig. 4: Example of convolution layer process (a) an image with size $\mathbf{5 \times 5}$ is convolving with $\mathbf{3 \times 3}$ kernel (b) Convolved feature map

4.4.1 ReLU layer

Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

4.4.2 Max pooing layer

This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

4.5 Softmax classifier

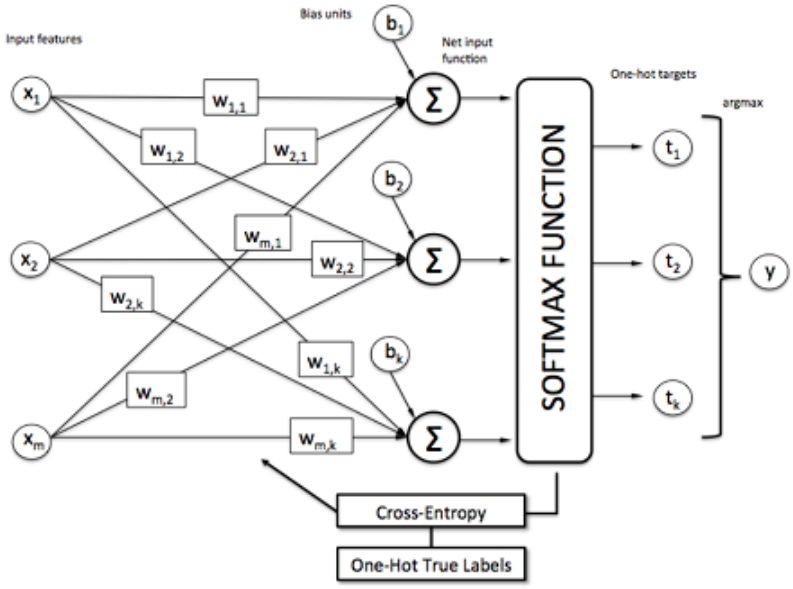


Fig.5: Traffic condition prediction using SoftMax classifier.

Generally, as seen in the above picture softmax function is added at the end of the output since it is the place where the nodes are meet finally and thus, they can be classified. Here, X is the input of all the models and the layers between X and Y are the hidden layers and the data is passed from X to all the layers and Received by Y. Suppose, we have 10 classes, and we predict for which class the given input belongs to. So, for this what we do is allot each class with a particular predicted output. Which means that we have 10 outputs corresponding to 10 different class and predict the class by the highest probability it has.

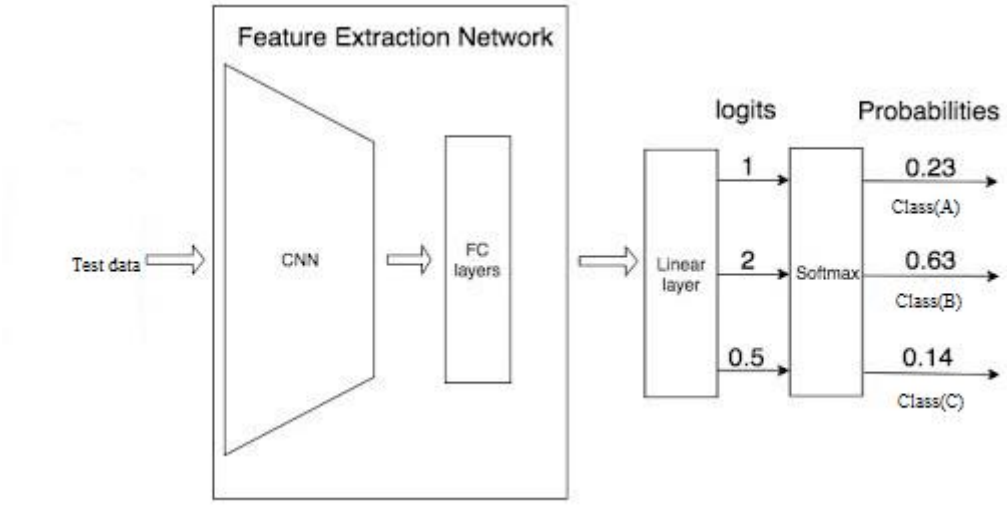


Fig.6: Example of SoftMax classifier.

In Figure 6, and we must predict what is the object that is present in the picture. In the normal case, we predict whether the crop is A. But in this case, we must predict what is the object that is present in the picture. This is the place where softmax comes in handy. As the model is already trained on some data. So, as soon as the picture is given, the model processes the pictures, send it to the hidden layers and then finally send to softmax for classifying the picture. The softmax uses a One-Hot encoding Technique to calculate the cross-entropy loss and get the max. One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if softmax predicts that the object is class A then the One-Hot Encoding for:

Class A will be [1 0 0]

Class B will be [0 1 0]

Class C will be [0 0 1]

From the diagram, we see that the predictions are occurred. But generally, we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function.

So, we choose more similar value by using the below cross-entropy formula.

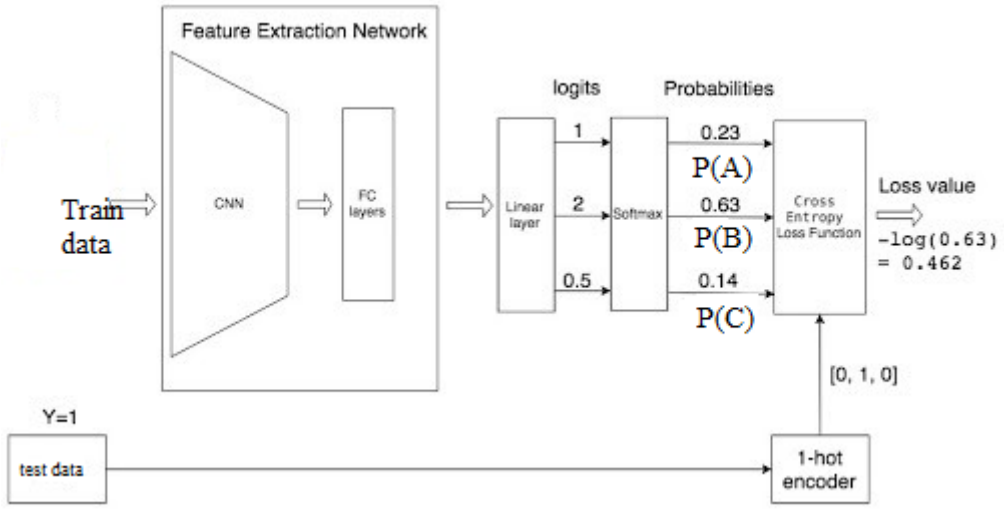


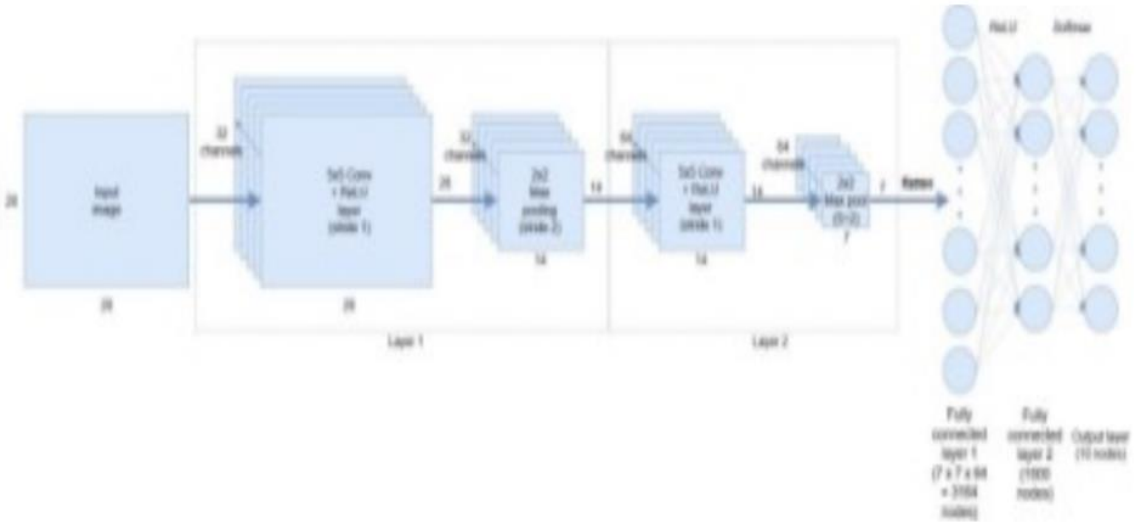
Fig.7: Example of SoftMax classifier with test data.

In the above example we see that 0.462 is the loss of the function for class specific classifier. In the same way, we find loss for remaining classifiers. The lowest the loss function, the better the prediction is. The mathematical representation for loss function can be represented as: -

$$LOSS = np.sum(-Y * np.log(Y\_pred))$$

4.SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:



4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

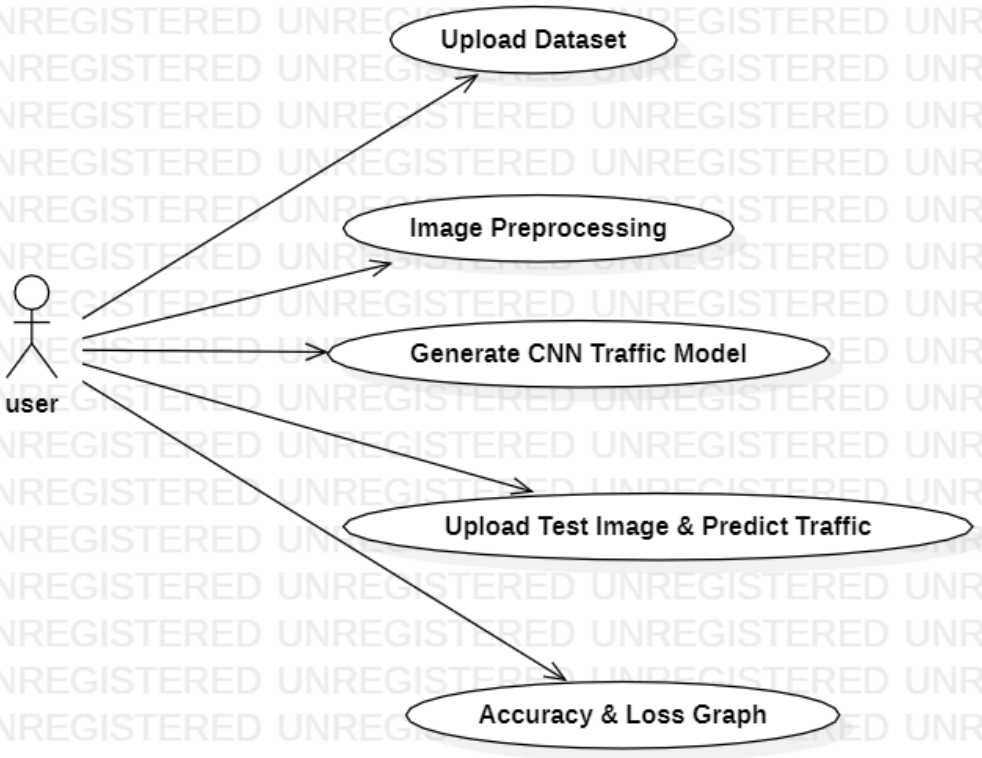
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

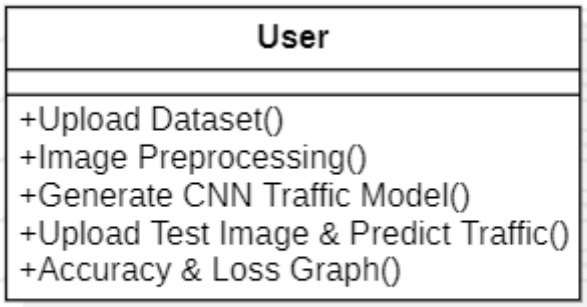
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



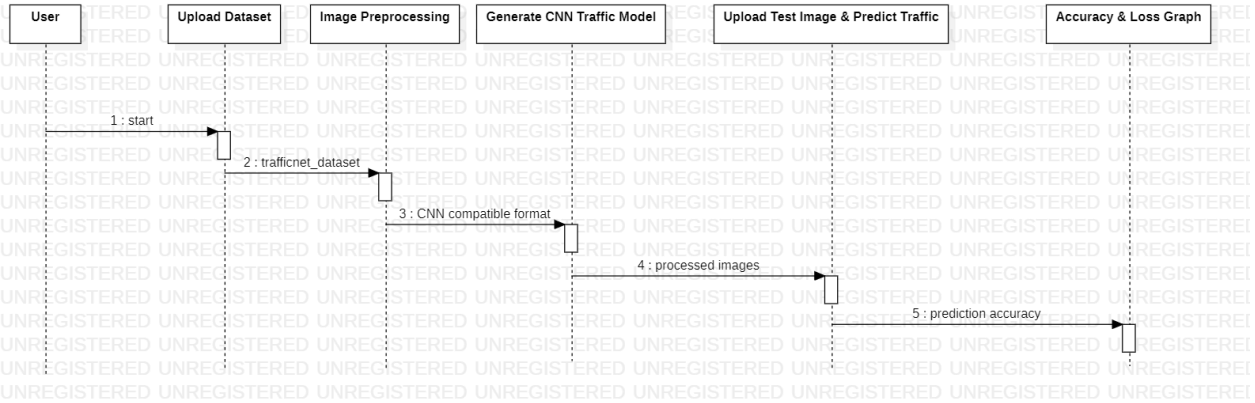
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



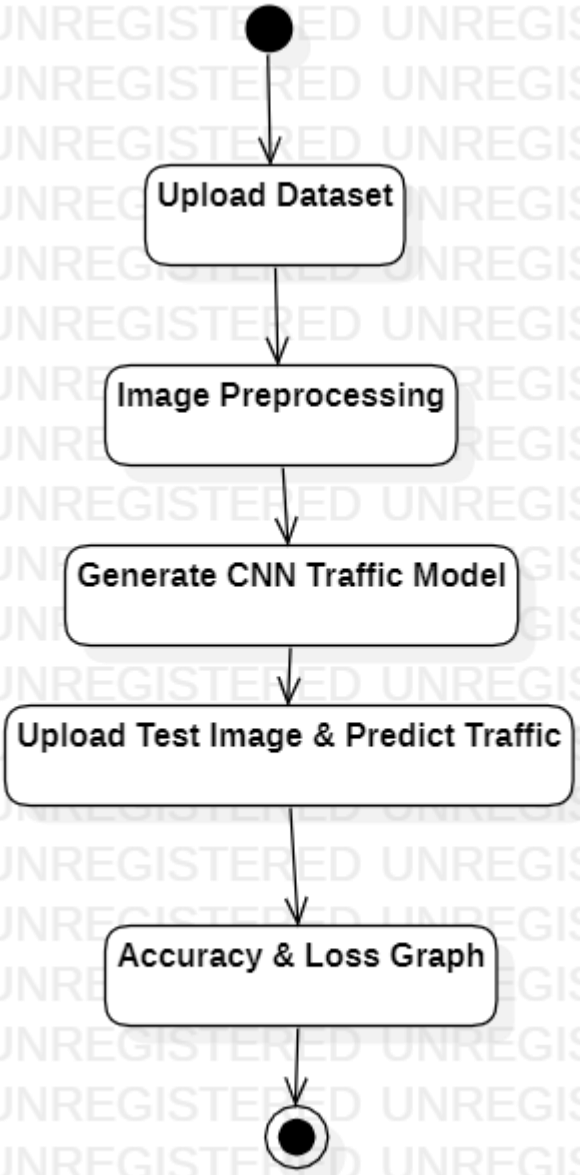
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

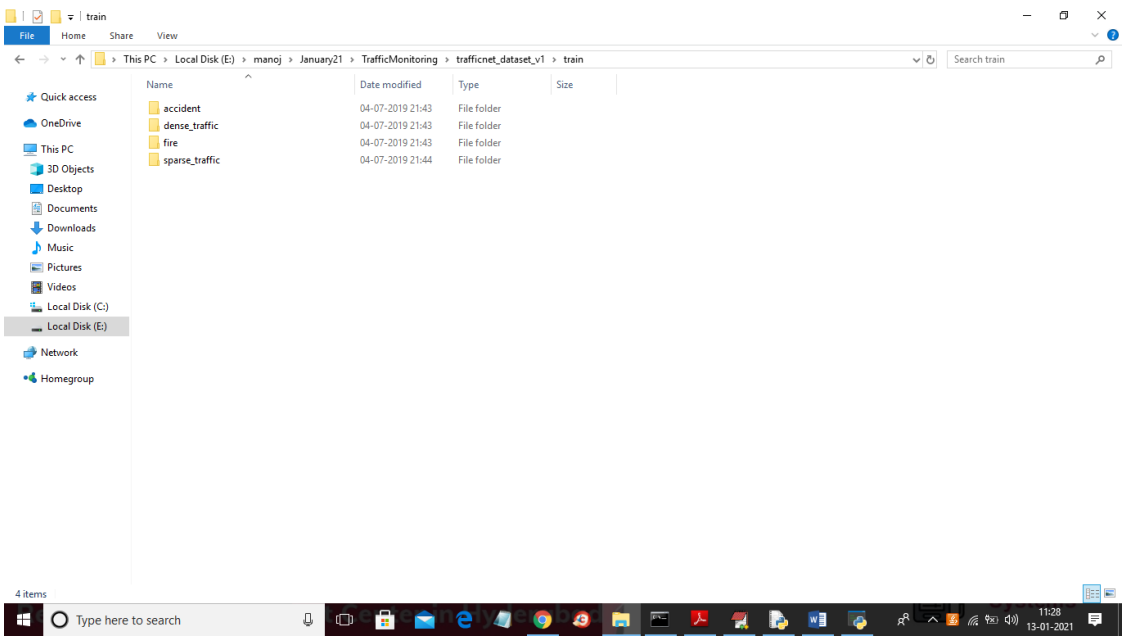
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



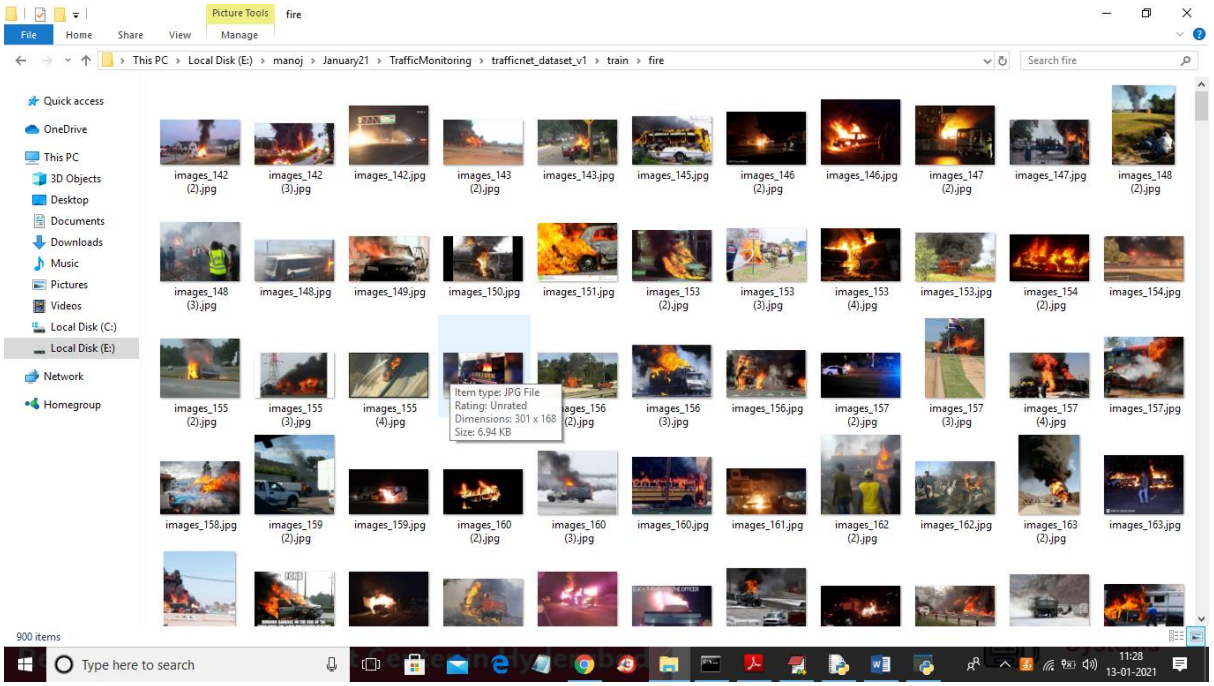
4.3 IMPLEMENTATION:

MODULES DESCRIPTION:

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.

CHAPTER 5

SOFTWARE ENVIRONMENT

Text Vs. Data Instead Of Unicode Vs. 8-bit What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

2. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.
Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So

I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data.*

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly**

programmed". And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year. But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it. So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.
So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning :-

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

3. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.

Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick

to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.
The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. Install Python on Windows 10, 8, and 7 devices.

CHAPTER 6

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

ACCEPTANCE TESTING

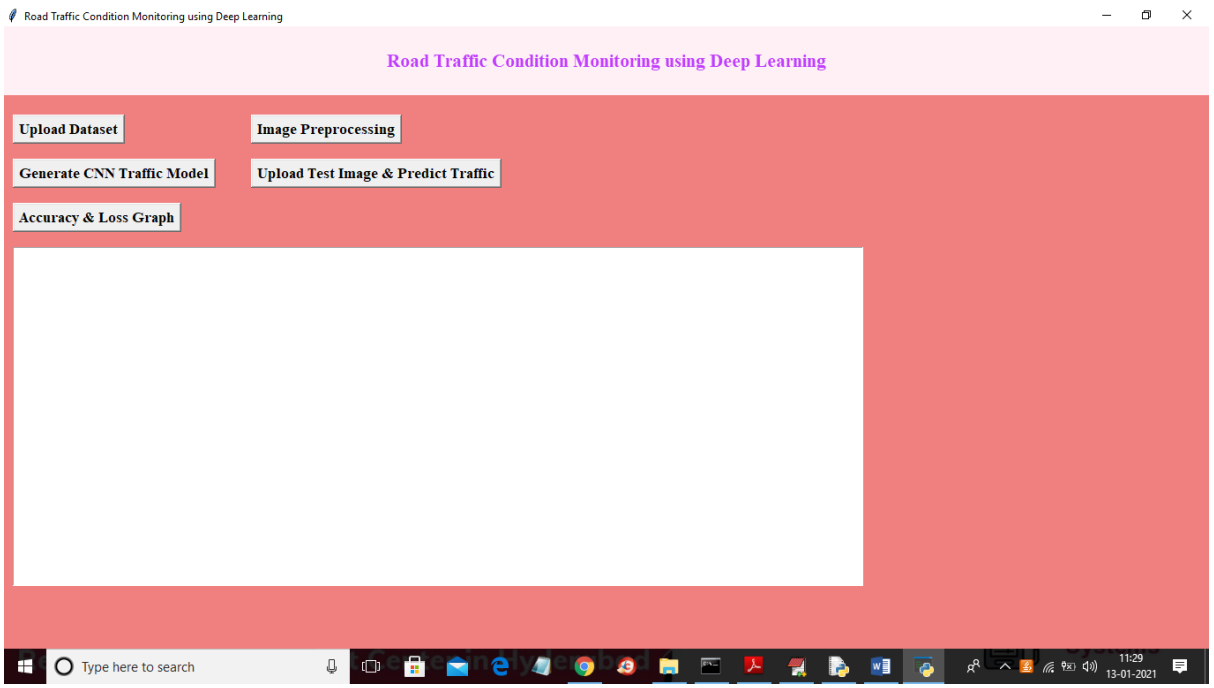
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

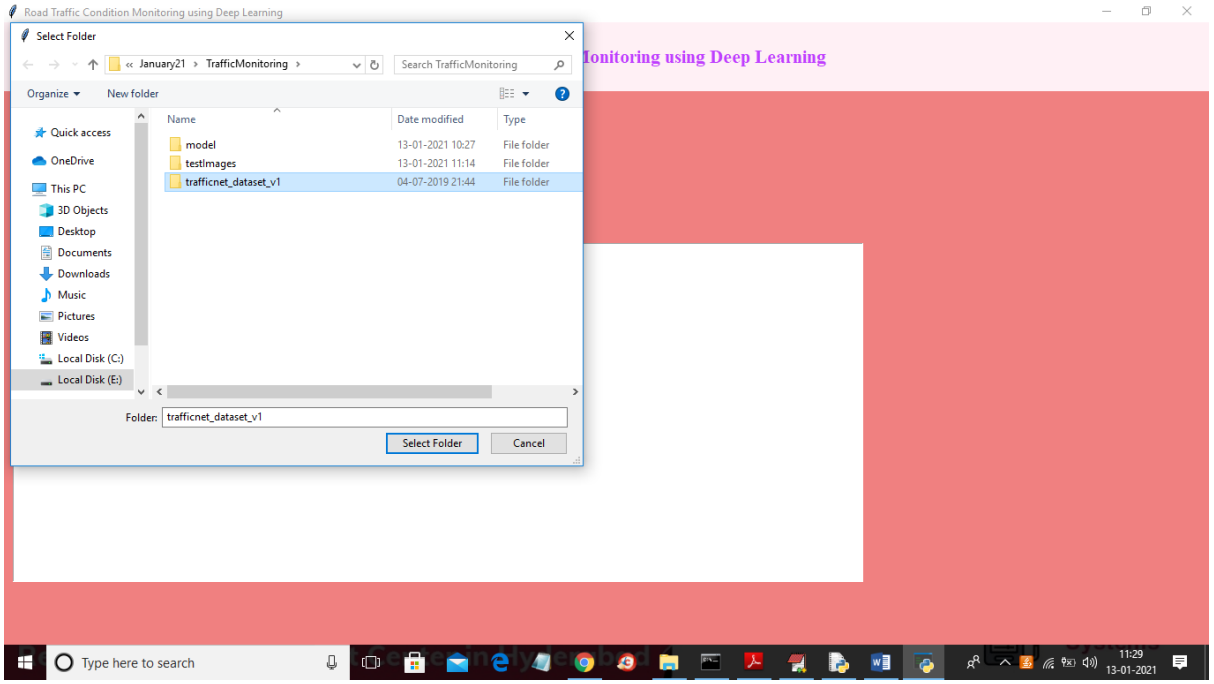
CHAPTER 7

OUTPUT SCREENS

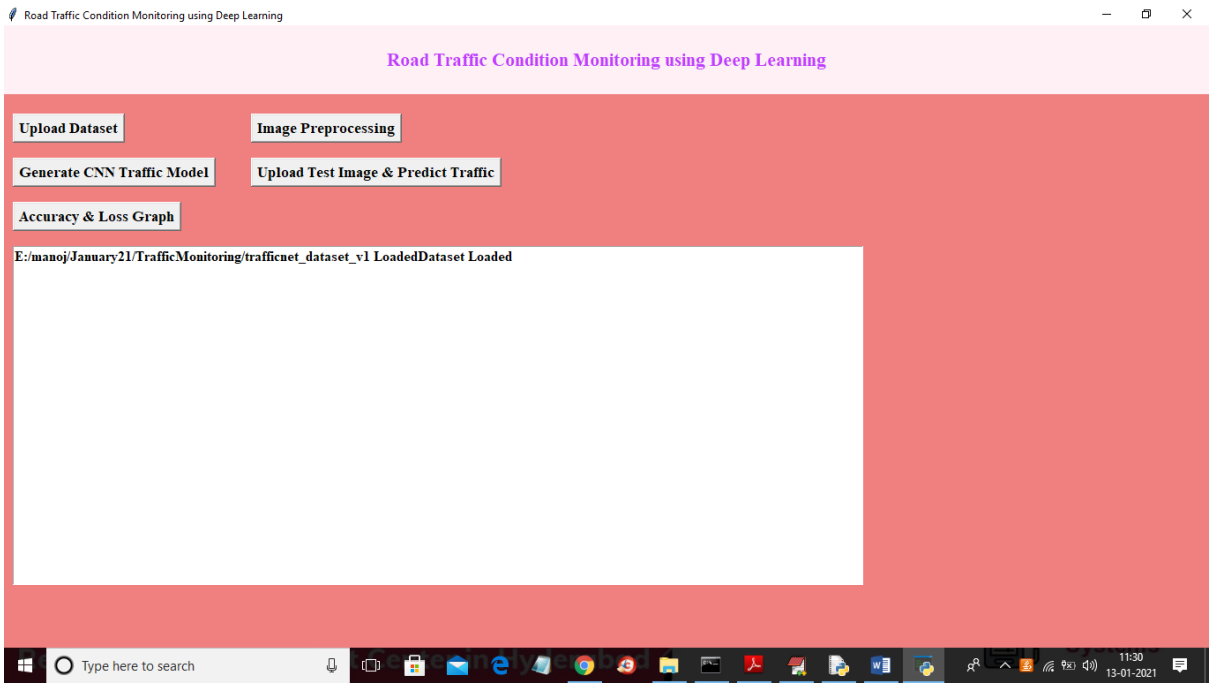
To run project double click on ‘run.bat’ file to get below screen



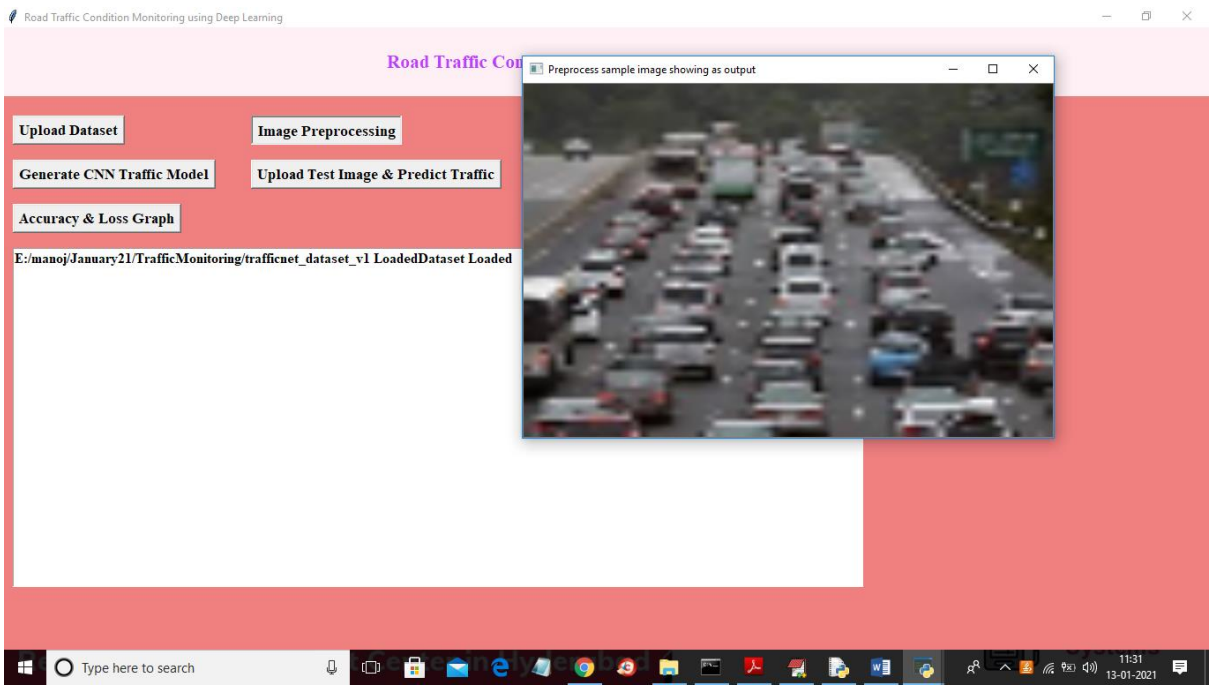
In above screen click on ‘Upload Dataset’ button to load dataset



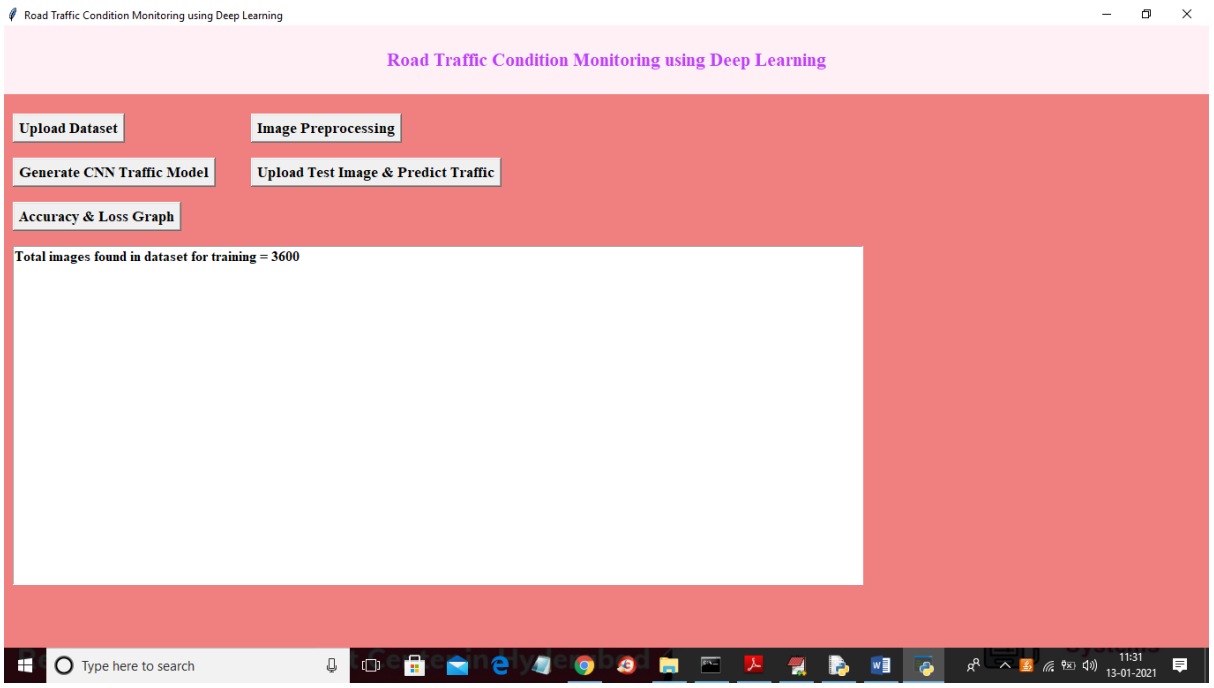
In above screen selecting and uploading 'trafficnet_dataset' folder and then click on 'Select Folder' to load dataset and to get below screen



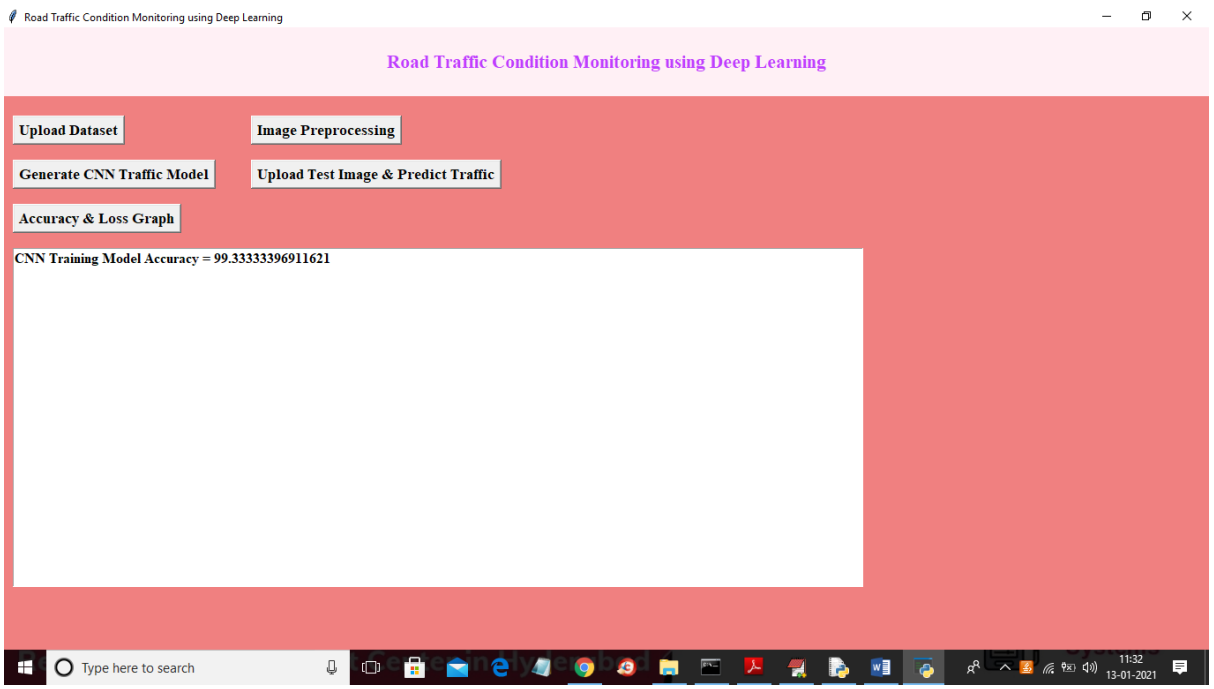
In above screen dataset loaded and then click on 'Image Preprocessing' button to read all images and then convert it CNN compatible format



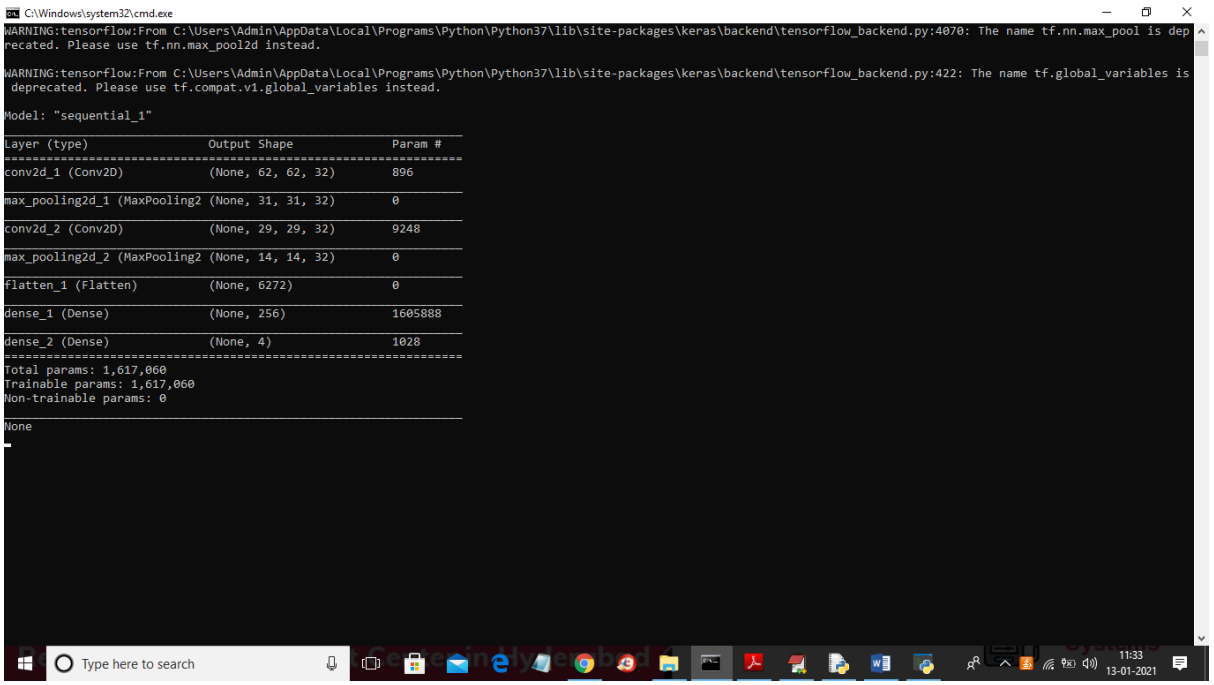
In above screen displaying sample image from processed images and now close above image to get below screen



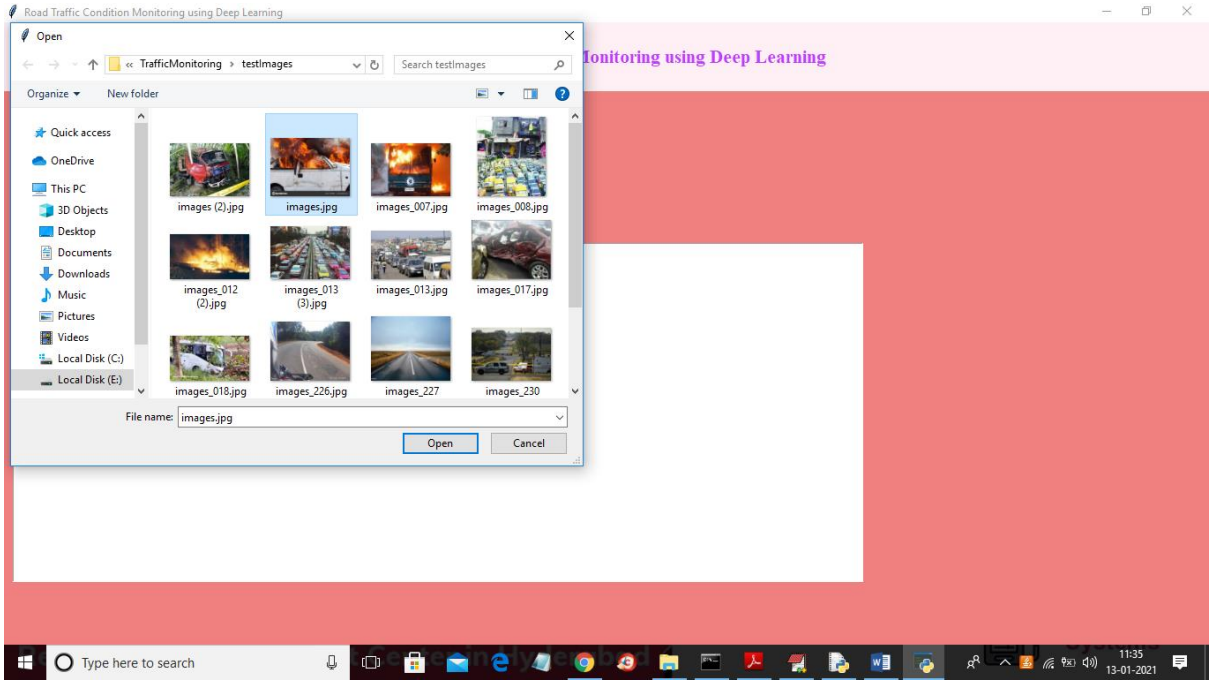
In above screen total processed images showing as 3600 and now click on 'Generate CNN Traffic Model' button to generate CNN model and to get below screen



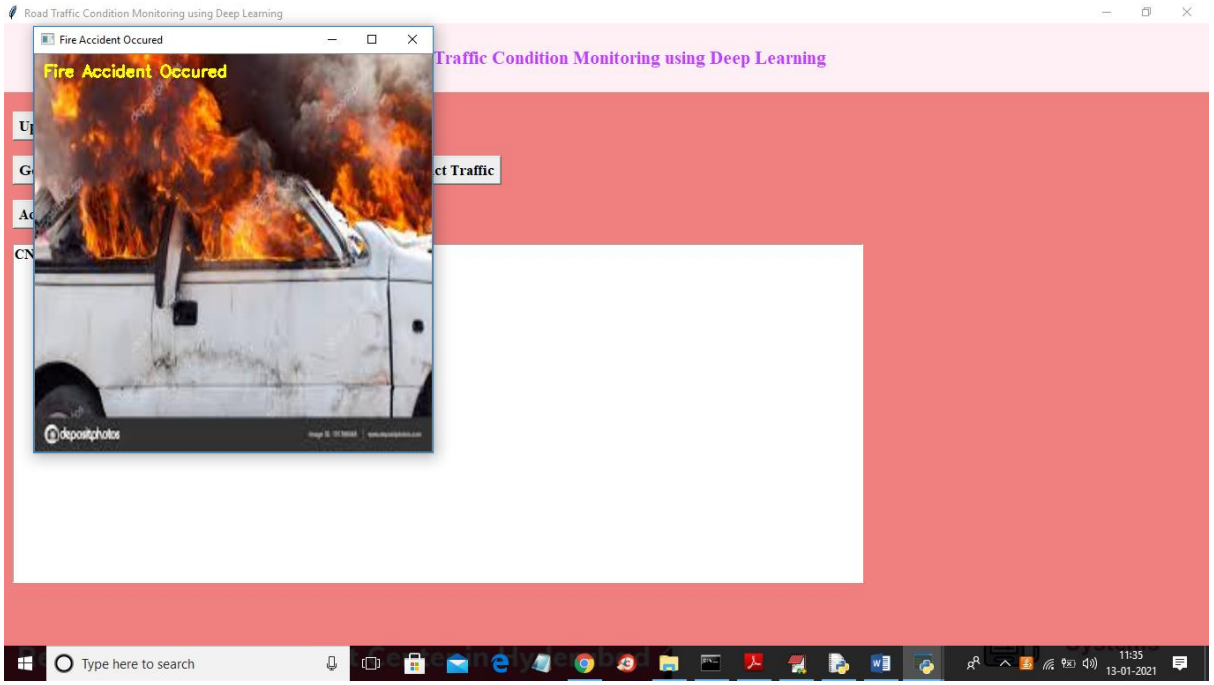
In above screen CNN model generated with prediction accuracy as 99% and in below black console we can see CNN layer details



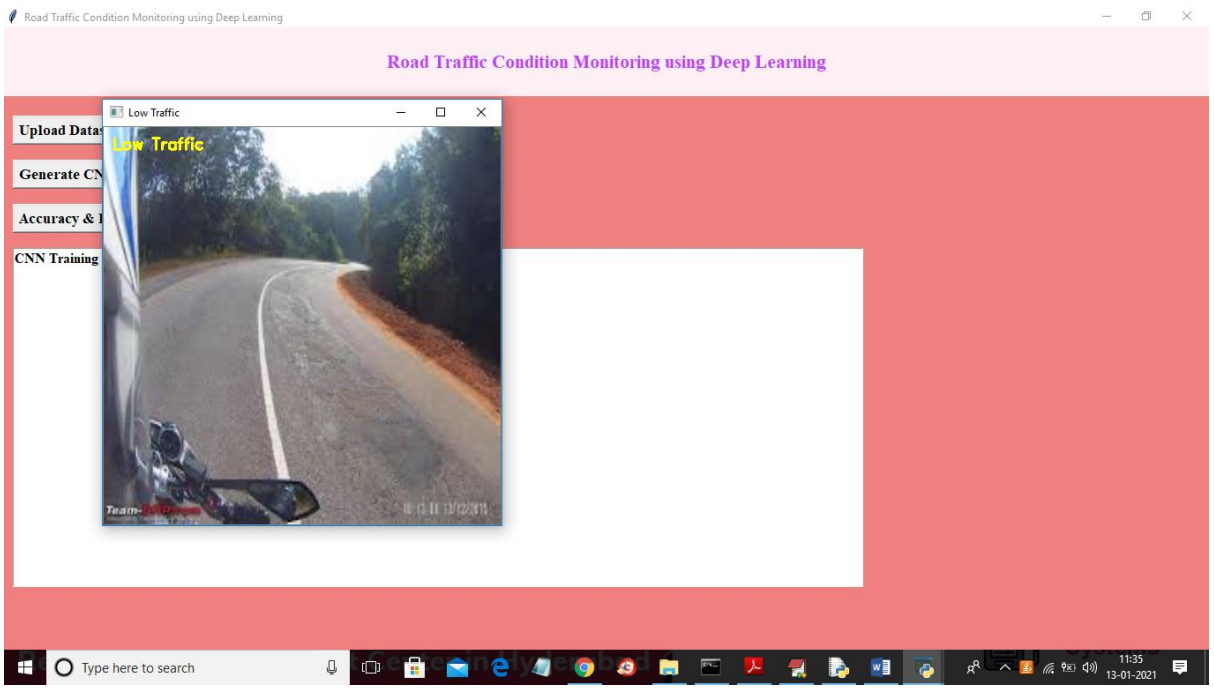
In above screen showing all layers of CNN model and each layer filtering images with different sizes such as 62 X 62, 31 X 31 etc. Now model is ready and now click on 'Upload Test Image & Predict Traffic' button and upload image to identify traffic condition



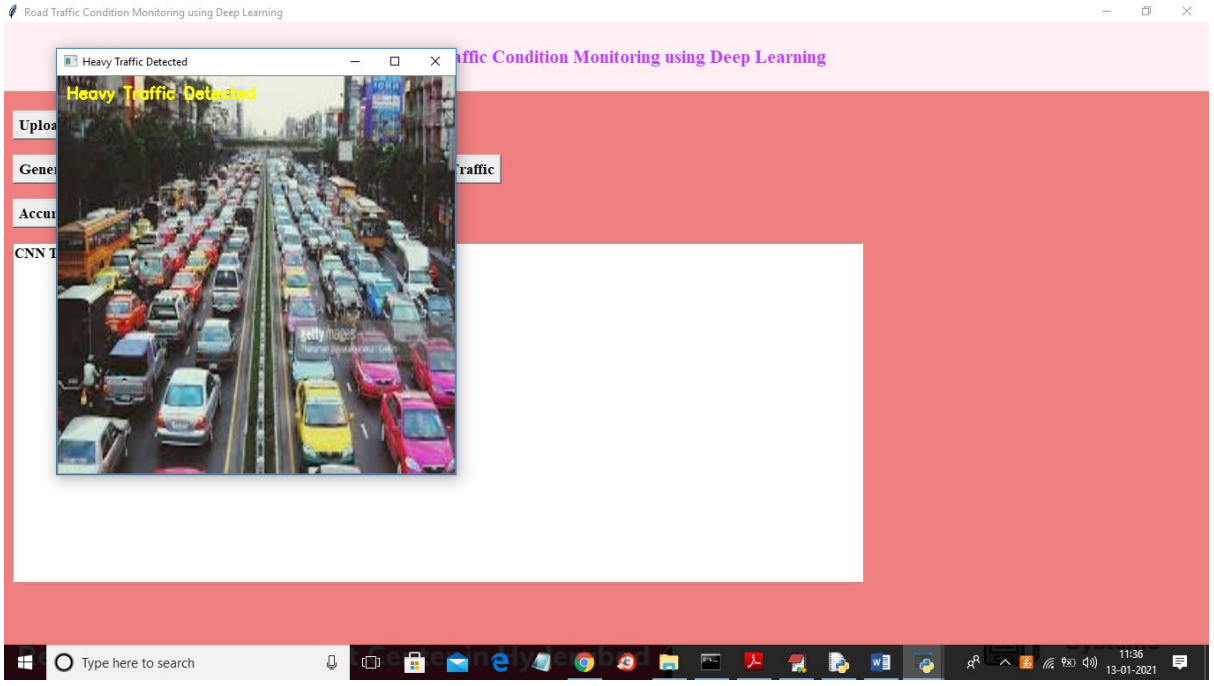
In above screen selecting and uploading 'images.jpg' file and then click on 'Open' button to get below result



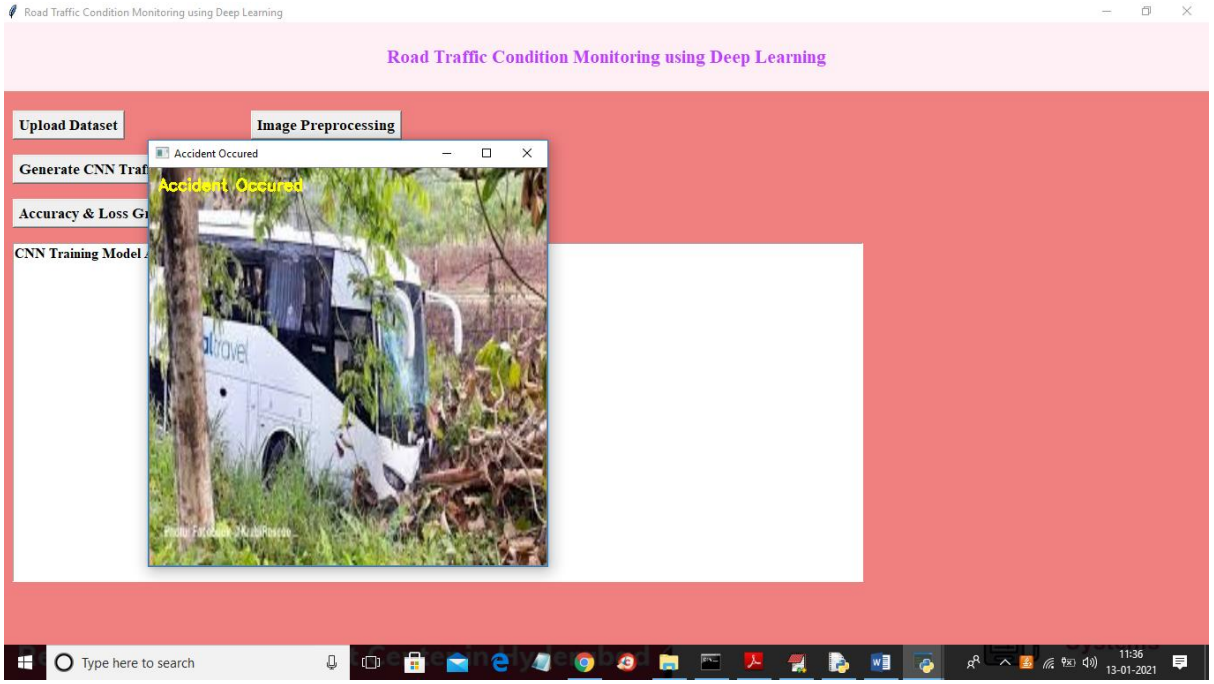
Test another image



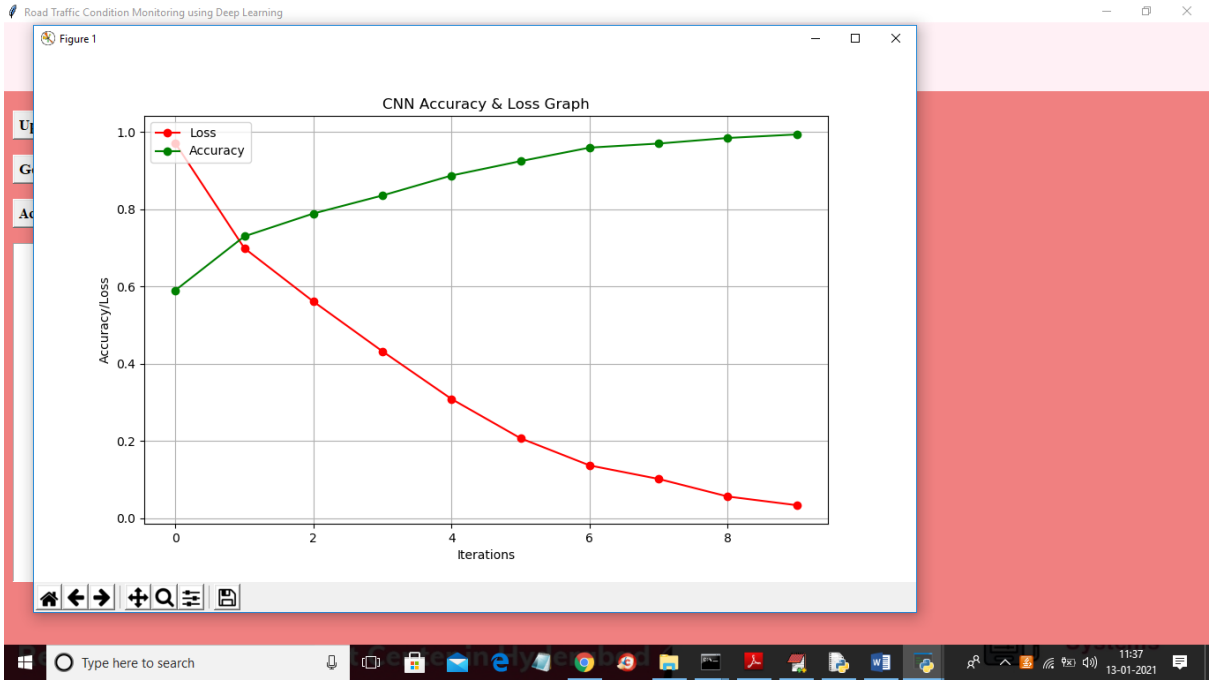
Test another image



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.

8.CONCLUSION

The Convolution neural network is approached to identify the condition of road congestion without human intervention. This is anticipated to deploy deep learning in various realistic functions. The proposed CNN for training and validation is considered a multi-class problem.

Machine learning proved to be an effective and low-cost technique to enable traffic monitoring in real time based on seismic data. In this study, we evaluated three neural network systems for this purpose and demonstrated that CNN provided the best performance in terms of accuracy and speed. CNN also surpassed the others in its ability to detect overlapping signals.

RNN did not perform as well as the others for traffic monitoring because its intrinsic reliance on temporal sequences conflicts with the random nature of traffic data. Although seismic data can be used for traffic monitoring, all neural networks have a shortcoming in terms of counting vehicles because they cannot identify the presence of multiple vehicles of the same class within a waveform frame.

The main limitation of neural networks is the human effort expended in acquiring and compiling a suitable amount of training data. We augmented our dataset by adding random noise. Although the models can be deployed without extra training, we recommend retraining the model as much as possible to guarantee the best performance in the generalization. Neural networks that process seismic data offer compelling advantages over current approaches to traffic monitoring. The seismic record has images compared to videos and other types of monitoring data. Because the system is simple and passive, consisting of a few geophones, it can be implemented for months at a time without supervision. The recorded data can be analyzed at a low computational cost to give clear statistical information for vehicles during the implementation period.

This makes the proposed system suitable for use in hard-to-access roads. Our favored method, based on CNN, is suitable for continuous records of a month or longer; CNN was able to process a month’s worth of data in approximately an hour. The proposed method can be extended by investigating the feasibility of using it to estimate more types of traffic data, dense traffic, low traffic, and accidents(fire, road).

Since we identified the car type via our CNN-based approach, the estimation of the speed of the vehicles could be possible; we presently are investigating accurate speed estimation systems. It may also be possible to extend a similar approach to other types of transportation, such as vessels, bicycles, foot traffic, or airplanes.

Future Enhancement

As a future enhancement, the traffic conditions are detected on the traffic videos in real-time. This can be done by video splitting technique are found and the traffic condition on every frame. Real-time traffic detection on video is quite an important research for developing countries like India where the population is more leads to heavy traffic on a daily basis.

9.REFERENCES

1. Bashar, A. (2019). “Survey onEvolving Deep Learning Neural Network Architectures”. Journal of Artificial Intelligence, 1(02), 73-82.

2. Zoe Bartlett et al, “A Novel Online Dynamic Temporal Context Neural Network Framework for the Prediction of Road Traffic Flow” IEEE Access, vol.7, 2019.

3. H. Lei et al., ``A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning," Pattern Recognit., vol. 79, pp. 290302, Jul. 2018.

4. P. Wang, L. Li, Y. Jin, and G. Wang, ``Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNNarchitecture trafficNet," in Proc. 13th IEEE Conf. Ind. Electron. Appl., May/Jun. 2018, pp. 11341139.

5. X. Zhu, Y.Wang, J. Dai, L. Yuan, and Y.Wei, ``Flowguided feature aggregation for video object detection," in Proc. ICCV, Mar. 2017, pp. 408417.

6.Z. Zhao. Chen, X.Wu, P. C. Chen, and J. Liu, ``LSTM network: A deep learning approach for short-term traffic forecast," IET Intell. Transp. Syst., vol. 11, no. 2, pp. 6875, Mar. 2017.

7.P. Li, D. Wang, L. Wang, and H. Lu, ``Deep visual tracking: Review and experimental comparison," Pattern Recognit., vol. 76, pp. 323338, Apr. 2018.

8. P. Wang and J. Di, ``Deep learning-based object classification through multimode fiber via a CNNarchitecture SpeckleNet," Appl. Opt., vol. 57, no. 28, pp. 82588263, 2018.

9. J. Zhao, Z. Zhang, W. Yu, and T.-K. Truong, ``A cascade coupled convolutional neural network guided visual attention method for ship detection from SAR images," IEEE Access, vol. 6, pp. 5069350708, 2018.

10.T. Pamula, ``Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks," IEEE Intel. Transp. Syst. Mag., vol. 10, no. 3, pp. 1121, Jun. 2018.