

# Dynamic Traffic Lights Using Vehicles Density using open CV Modules

Bussa Bala Naga Pranay, Chintala Sanjay ,  
Durgesh Bhakta  
19R21A0512, 19R21A0516, 19R21A0521

Dr. P. Michael Preetam Raj  
Associate Professor, MLR Institute of technology

## **Abstract:**

**Purpose – The purpose of this article is to understand and solve traffic congestions and properly manage traffic by real time density of traffic on roads in a junction.**

**Design/methodology/approach – Based on the frames captured it will then be processed using algorithms and the density is calculated,**

**Findings – various improper traffic management is leading to lot of chaos and causing severe noise pollution majorly in metropolitan cities.**

**Practical implications – Can create a great impact on various sectors since it reduces lot of commute period and brings in a lot of mental peace among the society.**

**Social Implications- Societal tranquillity can be achieved since traffic is more controlled and more effective.**

## INTRODUCTION:

The issue of traffic congestion is becoming more and more significant in the present-day society, as the number of vehicles on the road is directly proportional to the population growth. Because of the growing number of vehicles and restricted infrastructure development, the traffic situation worsens by the day [2]. In Indian cities, a crucial challenge is the inability to further expand the existing infrastructure, leaving improved traffic management as the only viable solution.

Getting trapped in traffic is a pain for everyone in the vehicle. Every day, several patients die while attempting to reach hospitals on time. One of the oldest methods was for traffic officers to direct traffic manually using hand signals [1]. A traffic management system based on ancient technology is better than a system based on a manual or fixed-time schedule. There is a lot of traffic associated with this concept. There are often green traffic lights for the latter route when there are no lanes and high traffic on other routes. As a result, traffic congestion increases..

The project's purpose is to use Artificial Neural Network (ANN) to create a traffic framework that is responsive to the present traffic environment [1,4]. Convolutional Neural Network (CNN) is used to obtain real-time traffic statistics such as traffic density and wait for length [1]. In addition, a model was trained to forecast which path is best for traffic signal light green. Whenever a lane at a junction has an additional number of vehicles, the traffic light turns green, this would improve traffic flow while simultaneously increasing travellers' comfort and driving safety. The suggested system regulates traffic signals based on traffic density [4]. Additionally, the project intends to give an emergency vehicle signal override through emergency

vehicle detection. This occurs whenever an emergency vehicle, such as an ambulance or fire department, becomes trapped in traffic [5]. A typical signal sequence is enabled under normal operational conditions. In the event of an emergency, an RF signal override is activated [7]. As a result, an interrupt is detected, and regular execution is interrupted for a few seconds. The usual operation is then resumed.

## EXISTING SYSTEM/LITERATURE SURVEY:

Controlling traffic in the current age has become exceedingly challenging due to the rise in automobiles, such as cars and bikes, consequently, signalling systems were delayed for a prolonged period of time [12]. To address this issue, we have developed a density-based traffic signal using Arduino Uno ATmega 328P with a 1000ms delay to regulate traffic based on density at crossings, four-side lanes, or road systems [9].

With traffic congestion being a common issue in numerous metropolitan areas, there is an immediate requirement for an effective traffic surveillance system to regulate traffic flow. Traditional timer-based traffic control systems have been found to be inadequate in solving this problem, and they also fail to prioritize emergency vehicles.

The human population in cities is growing at an exponential rate, as is the number of cars [17]. Traffic control signals have long been used to help cities manage traffic flow. However, traditional traffic control signals fail in terms of time management. It allots equal time slots to each road, regardless of traffic quantity. This causes unnecessary delays for drivers, which is not always possible [15].

A ground breaking real-time traffic control system is introduced in this study that employs image processing techniques to efficiently manage traffic [17]. The system involves placing a camera at each traffic signal point to capture images of the roads where traffic is expected to be congested. The Mat lab image processing tools are utilized to determine the number of vehicles in these images, and varying timings are assigned based on the count, with a green signal enabling vehicles to move [7,8]. The proposed prototype employs LEDs to display the green and red signals, and a seven-segment display to indicate the diminishing timer for the green signal [10].

LITERATURE SURVEY:

BASIC CONCEPTS:

S. No	Abstract or objectives	Techniques used	Limitations
1	A Real-Time Density-Based Traffic Signal Control System	1. Decision support System (DSS) 2. Sensor network 3. Cloud IoT	Does not detect any crash detection, mob detection, and emergency services such as fire engines, or ambulances.
2	IoT based Smart Traffic density Control using Image Processing	1. Internet of Things (IoT) 2. Sensor Node	Proposed using raspberry pie controller.
3	An Investigation Approach used for Pattern Classification and Recognition of an Emergency Vehicle	1. Sensor System 2. Deep Neural Network 3. Internet of Things (IoT)	Proposed only to for Classification and Recognition of an Emergency Vehicle
4	Implementation of Efficient Automatic Traffic Surveillance using Digital Image Processing.	1. Sensor System 2.c++ programming language 3.arduino programming	Very cost effective and very hard for maintaining. Does not provide accurate result as expected.
5	A Hybrid Framework for Expediting Emergency Vehicle Movement on Indian Road	3. Image Processing using SSD Mobile net.	Works using pre-defined libraries such as RCNN which is not so accurate.
6	System Integrated with Acoustic Based Emergency Vehicle Detection	1. Raspberry Pi 2. Internet of Things (IoT). 3. Machine Learning 4. Computer Vision (OpenCV)	This model utilizes many sensors to make it 100% automated, so maintenance and expenses of each sensor can become a burden.
7	Real-time Area Based Traffic Density Estimation by Image Processing	1.Artificial Neural Networks (ANN) 2.Deep learning (DI)	Developed only for a particular location. (Bangladesh)
8	Smart Traffic System with Real Time Data Analysis Using IoT.	1. Internet of Things (IoT) 2. Convolutional Neural Network (CNN)	Density of traffic is not calculated which lead to many limitations.
9	TRAFFIC CONTROL WITH EMERGENCY OVERRIDE	1. OpenCV 2. Machine Learning 3. Image Processing	Chances of executing false signal override protocol by faking the sensors.
10	Density Based Traffic Signal System Using Arduino Uno.	1. Internet of Things (IoT) 2. Arduino 3. Sensor Node	Very cost effective and very hard for maintaining. Does not provide accurate result as expected.

Table. 1 Literature survey

**Open CV Framework:**

Open CV is also known as Open Computer Vision Framework. This cross-platform package allows us to develop real-time computer vision applications [16]. Image processing, video recording, and analysis are its primary functions, as well as features such as face recognition and object recognition. [17].

**CNN Basics:**

CNN is comprised of three layers, they are:

- 1.Convolutional layer
- 2.Pooling layer
- 3.Connected layer (FC layer)

Additionally, we have two more layers to these above three layers, which include:

- 1.Dropout layer
- 2.Activation function

All these together form a complete Convolutional Neural Network (CNN)

The features of an image are analysed and extracted by CNN algorithms. Convolutional Neural Networks are advanced machine learning algorithms that can process vast datasets comprising millions of parameters, primarily designed for analysing 2D images, and linking the resulting representations to their respective outputs [15]. A CNN is a multi-layer, supervised network that can learn new features dynamically from datasets. Recently, CNN have demonstrated state-of-the-art results in almost all significant classification challenges [16]. Additionally, within the same framework, they are capable of systematically isolating elements and classifying them.

**CONVOLUTIONAL LAYER:**

The task of convolutional layer is to extract the features of given input image, this is done by applying convolution filters to the given input image. There is a mathematical process involved in this layer, when a filter is applied to the image then the dot product of the filter and the parts of input feed image covered by filter is performed. After performing the dot product function maps are generated as an output [13].

Later, this very same feature map is used as an input to other layers. The Conv2D function's arguments are as follows:

- 1.Filters:** applied to input feed to generate feature maps
- 2.Kernel size:** gives the size (n x n) of the convolution filter matrix.
- 3.Activation:** all layers, excluding the output layer, are activated using the Rectifier Linear Unit function. Using ReLU, we also incorporated nonlinearity to our network. To locate any linear relationships in the feature map, this is required.
- 4. Input Layer:** it contains the number of channels and the geometry of the input pictures (3 for colour)

**Pooling layer:**

The pooling layer is the next layer in our convolutional neural network [14]. The pooling layer's primary purpose is to reduce the spatial dimension of data travelling through the network. Pooling can be accomplished in two ways in convolutional neural networks [13]. There are two types of pooling: maximum pooling and average pooling. The more popular of the two, Max Pooling, scans the highest value for each section of the image. Average pooling computes the average of picture elements in a predefined size zone. The pooling layer connects the convolutional layer and the fully connected layer [11].

**Fully connected layer:**

The Fully Connected (FC) layers consist of weights, biases, and neurons and are utilized to connect neurons between layers [2,10]. In this layer, the output of the preceding convolutional layer is flattened, and each node of the current layer is connected to every node in the subsequent layer. This layer receives input from the preceding layer, which may be a convolutional layer, a ReLU layer, or a pooling layer [15,1]. The classification process starts from this point.

**.Dropout:**

The training dataset may be overfilled if all the features are linked to the FC layer. If a model can perform well on training datasets but displays poor performance when used with fresh datasets, it is said to be over fit [14]. A dropout layer is used to address this issue, shrinking the size of the neural network model by removing a number of neurons during training [16]. 20% of the nodes in the neural network are randomly eliminated when a dropout of 0.2 is exceeded [29,30].

**Activation:**

We need an activation function which can work efficiently for multi-class classification and among the activation functions such as sigmoid, tanH, Softmax and ReLU, the best suitable for this situation are Softmax and ReLU [6]. Each activation function has its own specific application [26,28].

In neural network process these activations functions play major role as they specify what decision to be taken or what node towards the end of process should be activated.[6]

**Model:**

The Sequential Model API is utilized to construct deep learning models, in which sequential classes and model layers are generated and added. In this particular model, the first step involves creating a 2D convolution layer comprising 32 filters, with 3 x 3 kernels and a ReLU activation and pooling layer. Next, we merge the output from these layers, allowing data to flow into the fully connected layers [16].

**PROPOSED METHODOLOGY:**

A CNN is a multi-layer, supervised network that can learn new features dynamically from datasets. Recently, CNN have demonstrated state-of-the-art results in almost all significant classification challenges [6]. Convolutional

neural networks are advanced machine learning algorithms that can process vast datasets comprising millions of parameters.

The proposed method for traffic density prediction supports live monitoring of multiple vehicles from different perspectives, in this live data of different vehicles are collected and sent to services where we segregate the vehicle types based on previously indicated vehicle, Then from the video stream features will be extracted and sent to cloud where we apply our machine learning algorithm to calculate traffic density & classifying the vehicles type such as heavy, small, emergency services etc. [11].

We also keep a database of different cars for detection of different types of vehicles and from different perspectives, and after detection of vehicle density, traffic lights are modified by analysing the priority of the roads [17].

**Dataset Description:**

This dataset consists of 10,517 images of vehicles, crowd, Emergency servicing vehicles are classified accordingly to classes which help in training a deep convolutional neural network [6].

**Data Pre-processing:**

In Pre-processing of data, we preform data cleaning using manual and automated Keras function utilization for removal of useless data as it can impact our results [24].

On preforming data cleaning, data is scaled to [0, 1] from its original RGB [0, 2] state for fitting to training size. Then data is partitioned into three stages they are, train data, test data and validation data [21].

We take 80% of data i.e., 8,413 images as train data, and reaming 20% i.e., 2,103 images for validation and testing purpose. The training data set is used to train model, but the test data set is hidden so you can test the model's accuracy [10, 20].

**Data Augmentation:**

Efficiency of any training-based algorithm can be increased when we have more data to train on, we can achieve this by augmenting of available image database [13]. In augmentation by Keras we have two techniques one is in-place data augmentation or non-fly augmentation, these enables scaling, resizing, rotating, shifting, zooming, flipping to horizontal and flipping to vertical of available image data [9].

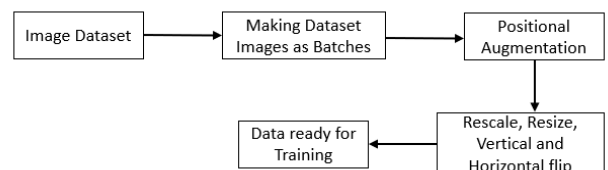


Fig. 1 Process of Data Augmentation

Figure 1 explains the working process of in-place data augmentation

1. The Image-Data-Generator is available data and is given as input[21].
2. Then on the Image-Data-Generator various types of augmentation techniques are applied such as scaling, resizing, rotating and etc., by making them into batches for easy processing[18,19].
3. These batches will now get higher data and will be used to train by CNN [27].

**SYSTEM ARCHITECHURE:**

The Machine Learning based user-friendly prediction of vehicles and calculation of traffic is designed in such a way that it receives the input as an image, based on the image the model recognizes whether there is a defect or not and produces an output [19].

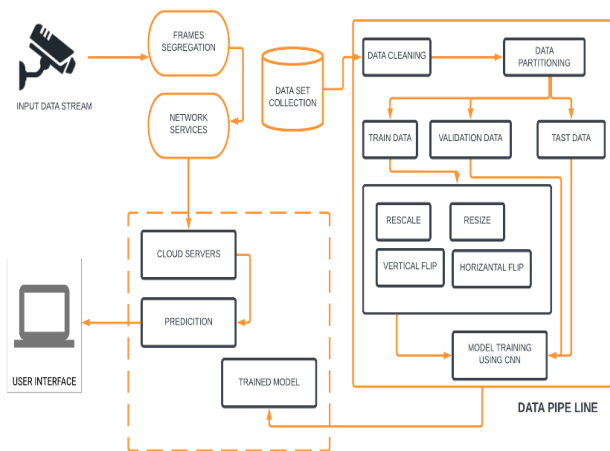


Fig.2 System Architecture

**Input Feed:**

1. Camera collects data feed from the roads.
2. Then the collected data feed is sent to the program.
3. Which then passes through internet connectivity modem.
4. The cloud server contains trained deployment model which takes data feed for analyzation.
5. Data feed analyzation is done through prediction.
6. If there is any fault in the data feed (image) it notifies the user through python user interface.

**Training Model Information:**

**1. Data Collection**

- For training the model the data is collected in various ways like from third party applications (Kaggle), collecting images manually and web scraping scripts by data scientists [29].

**2. Data Pipeline**

- Data input pipeline is used to convert images into 3D spatial data structures, each dimension has numbers ranging from 0 – 255 based on RGB scale [25].

**3. Data Cleaning**

- Data cleaning is the process of fixing blurry, incorrect, unformatted, duplicate and incorrect data within the collected data set [27].

**4. Data Partitioning**

- After data cleaning the data is divided into three non-overlapping sets, they are [22].
  - I. Training Set
  - II. Validation Set
  - III. Test Set

**5. Data Augmentation**

- Data Augmentation generates more training samples from existing samples for accuracy.
- It includes resize, rescale, horizontal flip and vertical flip [23,24].

**6. Train Data using CNN**

- It involves convolution, rectified linear activation and pooling.
- Trained model – it is the model artifact that is created by the above training process [31, 32].

**UML DIAGRAMS:**

The following UML diagrams are used to represent the proposed system.

- Use Case diagram
- Sequence diagram
- Activity diagram

**Use Case Diagram:**

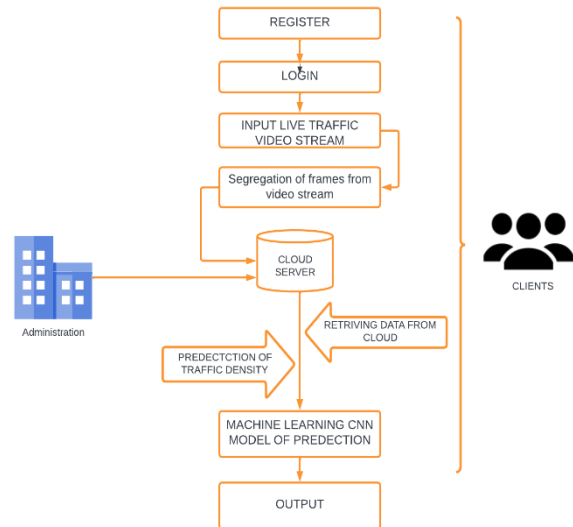


Fig. 3 Use-Case Diagram

The above Figure 3 describes the function and scope of the system and also the system requirements.

There are two major actors in our use case diagram, they are:

1. Admin
2. End User

- In total the application contains eight use cases that represent system functionality.
- Each actor interacts with a particular use case.
- The admin interacts with the error logs cloud server & database, ML CNN model and the output predicted for checking the condition of the services

and errors, updating the database, and verifying the output predicted by the ML CNN model.

- The user interacts with register, login, input feed camera and the output predicted.

**Note-**In case of the user, the interaction with the output means that the user can only view the displayed output but cannot modify the output.

**Activity Diagram:**

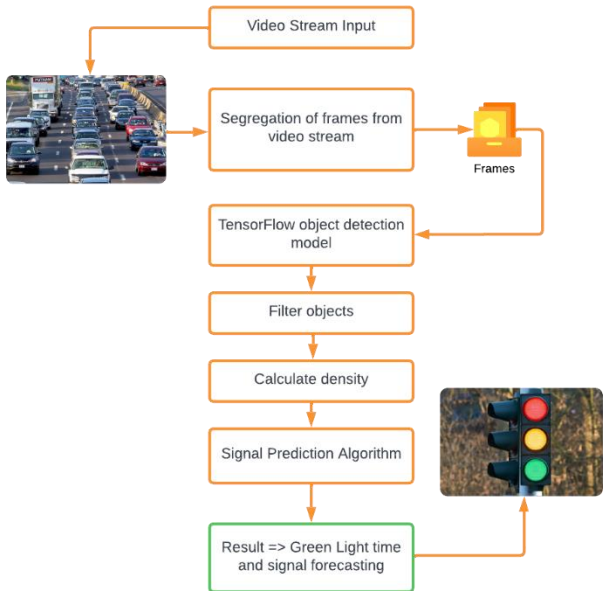


Fig. 4 Activity Diagram

The above Figure 4 provides an overview of the application by the sequence of the actions in a process

- Initially the user has to register the road and signal through application, after registration the user has to login into the application with respective login credentials.
- Once done with the login process, images stream is to be provided through the live URLs. Live-monitored images are sent as input feed via camera.
- Now the extracted data is sent to the cloud server, the database in the cloud contains a large number of datasets which include huge data sets related to the vehicles and traffic.
- The ML CNN model is trained in such a way that based on the existing data set present in the database it predicts the presence of the vehicles and traffic [33].
- If any vehicle is predicted then the application notifies the user regarding the presence of the vehicles or crowd the shapes with different colour code are augmented on the video stream as shown in below figure.



Fig.5 Augmenting shape on vehicles.

**Sequence Diagram:**

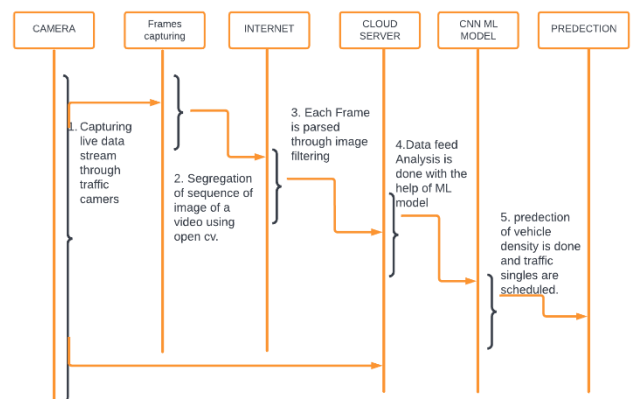


Fig.6 Sequence Diagram

From the figure 6 the sequence is as follows:

1. The camera collects data feed from the traffic surveillance video stream.
2. Then the collected data feed is sent to the program.
3. Where the images are segregation into the frames of images and each image is sent to service for processing.
4. The collected data is then passed through an internet connectivity modem.
5. The cloud server contains a trained deployment model which takes data feed for analysis.
6. Data feed analysis is done through Machine learning prediction which involves: -
  - Data Collection
  - Data Pipelining
  - Data Cleaning
  - Data Partitioning
  - Data Augmentation
7. Prediction of traffic density is done and traffic signals are scheduled according to the priority [6].

**CONCLUSION:**

The primary focus of this research is the development of a density-based traffic signal control system that is adaptive and real-time. It lays the groundwork for constructing an intersectional traffic regulation structure. Since the solution is decentralized, as explained in the thesis, a four-way crossing was selected for testing traffic scenarios. Traffic

movements trigger the junctions based on density, This reduces overall wait times and enables more efficient traffic flow. The system works automatically, depending on the collection of the density images sent from the website to the server. [11].

**Future work:** The effect of weather conditions on image quality, such as heavy rain or fog, has not been taken into account. The NN approach could be expanded to a multi-agent network and tested on a multi-intersection model [8].

#### REFERENCES:

- [1]. Dr. Brenner, "Consultant for Intelligent Transportation System Studies," Muscat.  
[Accessed: 25-Nov-2015].
- [2]. Divya Vani P., Aruna. K. and Ragendra Rao K., "Internet of Things-A Practical Approach to Certain Cloud Services using CC3200," Internet of Things-A Practical Approach to Certain Cloud Services using CC3200- Volume 117 No. 10.  
[Accessed: 25-Nov-2017].
- [3]. Po-Yi Liu, and Hsu-Yung Cheng (2010) "Vehicle Tracking in Daytime and Nighttime Traffic Surveillance Videos", International Conference on Education Technology and Computer (ICETC), 978-1-4244-6370- IEEE  
<https://ieeexplore.ieee.org/document/5529800>  
[Accessed: 25-Nov-2022].
- [4]. Marek Wojcikowski. "An intelligent image processing sensor The algorithm and the hardware implementation", in 1st International Conference on Information Technology, Gdansk.  
[www.elib.cs.sfu.ca](http://www.elib.cs.sfu.ca)
- [5]. A. Cao, B. Fu and Z. He, "ETCS: An Efficient Traffic Congestion Scheduling Scheme Combined with Edge Computing," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019.
- [6]. Marek Wójcikowski. "FPGA-Based Real-Time Implementation of Detection Algorithm for Automatic Traffic Surveillance Sensor Network", Journal of Signal Processing Systems, vol. 68, pp. 1-18, July 2012.
- [7]. Ashish Jain, Manisha Mittal, Harish Verma, Amrita rai, "Traffic Density Measurement based On-road Traffic Control using Ultrasonic Sensors and GSM Technology" Proc. of Int. Conf. on Emerging Trends in Engineering and Technology.
- [8]. A. Cao, B. Fu and Z. He, "ETCS: An Efficient Traffic Congestion Scheduling Scheme Combined with Edge Computing," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019, pp. 2694-2699.
- [9]. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks", Communications of the ACM, Vol.60, Issue 6(2017), , PP 84-90.
- [10]. Uthara E. Prakash, Athira Thankappan, Vishnupriya K. T., Arun A. Balakrishnan, "Density Based Traffic Control System Using Image Processing", Proceedings of 2018 International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR).
- [11]. P. N. Spoorthi, S. D. Yashwanth. "Intelligent Traffic Management System", 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 2021
- [12]. W. A. C. J. K. Chandrasekara, R. M. K. Rathnayaka, L. L. G. Chaturanga. "A Real-Time Density-Based Traffic Signal Control System" 2020 5th International Conference on Information Technology Research (ICITR), 2020.
- [13]. R. Bhargavi Devi, D. Kavya Reddy, E. Sravani, Gaddam Srujan, Shiv Shankar, Shubhro Chakrabarty. "Density based traffic signal system using Arduino Uno", 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017
- [14]. Uthara E. Prakash, K. T. Vishnupriya, Athira Thankappan, Arun A. Balakrishnan. "Density Based Traffic Control System Using Image Processing", 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 2018
- [15]. "Smart Trends in Information Technology and Computer Communications", Springer Science and Business Media LLC, 2016
- [16]. Anilloy Frank, Yasser Salim Khamis Al Aamri, Amer Zayegh. "IoT based Smart Traffic density Control using Image Processing", 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), 2019
- [17]. Aneesa Saleh, Steve A. Adeshina, Ahmad Galadima, Okechukwu Ugweje. "An intelligent traffic control system", 2017 13th International Conference on Electronics, Computer and Computation (ICECCO), 2017
- [18]. Koenderink, J and Van Doorn, A. The structure of locally orderless images. *IJCV*, 31(2/3):159-168, 1999.
- [19]. Boureau, Y, Bach, F, LeCun, Y, and Ponce, J. Learning mid-level features for recognition. In *CVPR*, 2010.
- [20]. A review of convolutional Neural network applied to Fruit image processing.
- [21]. The difference between R-CNN and Fast R-CNN in image analysis by Karan Aggarwal Issue 5 , October 2022.
- [22]. Data Cleaning: Problems and Current Approaches. by E. Rahm and H. Hai Do , volume 23 December 2000 IEEE paper
- [23]. A survey on image data augmentation for deep learning -journal of big data <https://doi.org/10.1186/s40537-01-0197-0> 2019
- [24]. Olaf R, Philipp F, Thomas B. U-Net: convolutional networks for biomedical image segmentation. In: *MICCAI*. Springer; 2015, p. 234-41.
- [25]. Seyed-Mohsen MD, Alhussein F, Pascal F. DeepFool: a simple and accurate method to fool deep neural networks. *arXiv preprint*. 2016.
- [26]. Alexander B, Alex P, Eugene K, Vladimir II, Alexandr AK. Albuementations: fast and flexible image augmentations. *ArXiv preprints*. 2018.
- [27]. Fast R CNN by Ross Girshick Microsoft Research published in IEEE volume 23 2012.
- [28]. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014.
- [29]. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *ICLR*, 2014.
- [30]. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman.
- [31]. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.
- [32]. J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012.
- [33]. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.