

# Emergency and Non-Emergency Vehicle Classification using Machine Learning

ASWATHY T G

20MCAR0105

PROF. DR. GANESH D

DEPARTMENT OF COMPUTER APPLICATION

JAIN KNOWLEDGE CAMPUS, BANGALORE

JAIN UNIVERSITY BANGALORE, INDIA.

## ABSTRACT

*A highly populated country like India faces too much traffic jam. Sometimes emergency vehicles like ambulances, fire engine get stuck in the traffic causing threat to life in many cases. It is important to give priority to this vehicle and help to clear its path. But it is difficult or sometimes impossible for traffic police to handle this. So, differentiating a vehicle into an emergency and non-emergency category can be an important component intraffic monitoring as well as self-drive car systems can easy understand the emergency vehicle as reaching on time to their destination is critical for these services. In case of automatic vehicles, they need an alert input message like this to clear the path to emergency vehicles. on other hand, this is useful for all types of vehicle to get sudden alert to clear the path to emergency vehicles. For this reason, we need an automated system that will be able to detect an emergency vehicle in heavy traffic road, let the controller know or automatically navigate other vehicles to clear its path. In this work, we have proposed an automated system to detect emergency vehicle using the Machine Learning.*

**KEYWORDS:** *Intelligent Transport System (ITS), Support Vector Machine (SVM), Convolutional*

*Neural Network (CNN), Decision Tree, Recurrent Neural Network (RNN), space invariant artificial neural networks (SIANN)*

## I. INTRODUCTION

According to a report published by Times of India about 146,133 people were killed in road accidents in India in the year 2016. Unfortunately, about 30% of deaths are caused due to delayed ambulances. Another Indian government data shows. More than 50% of heart attack cases reach hospital late, which can constitute unavailability of ambulances too but majority of it is due to patients stuck in traffic.

Since the population and transport system increase day by day, the demand for managing them increase at the same time. The world is getting populated so fast. Therefore, the number of machines from any types including vehicles increased at the same time. That being said, new topics like traffic, accidents and many more issues are needed to be managed. It is hard to manage them with the old methods, new trends and technologies have been found and invented to handle each and every milestone that human kind is trying achieve. One of these challenges is traffic in highways and cities. Many options like traffic light, sign, etc. deployed in order to deal with this phenomenon. It

seems that these options are not enough or not so efficient alone. New technologies like object detection and tracking are invented in order to utilize automated camera surveillance to produce data that can give meanings for a decision-making process. This phenomenon has been used for different kind of issues. The new trend Intelligent Transport System (ITS) has many elements which object detection and tracking are one of them. This system is used to detect vehicles, lanes, traffic sign, or vehicle make detection. The vehicle detection and classify ability gives us the possibility to improve the traffic flows and roads, prevent accidents, and registering traffic crimes and violations.

Emergency vehicles include ambulances, fire department vehicles, police vehicles, and privately-owned vehicles for firefighter or life support agencies. In traffic jams, many people do not bother to give pass-way for the emergency vehicle and also traffic police can't see which lane they should clear for the ambulance. Therefore, many patients lose their lives before reaching hospitals. Traffic jam situation is also a very big challenge for the firefighter team. Every second is very valuable for the firefighter team. Many peoples' lives and many properties are lost for delaying the emergency firefighter services on the emergency situation. We need to a build a system to detect vehicle and classify it as an emergency or regular vehicle using Machine Learning and Convolution Neural Network. Humans can easily recognize vehicles in videos or images or to identify different types of cars. In computer algorithms and programs, it is highly depending on the types of data. Some challenges like the weather or light are also plays important role on making the process easy or much hard. At the same time, we have different types and shapes of vehicles. More than that the new challenge could be to identify moving objects in a video in real time where they are different in size and shape. There are different techniques and methods for vehicle detection and classification. The variety of these techniques are in types of algorithms like Support Vector Machine (SVM), Convolutional Neural Network (CNN), Decision Tree, Recurrent Neural Network

(RNN) etc. The field is constantly evolving since the industry is focused on this system or Computer visionary. In this thesis we investigate two algorithms SVM and Decision Tree to identify how they can apply in the field and which one works better than the other.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicit programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

## II. EXISTING SYSTEM

Existing system is a Manual traffic control system where the police officers convey sign board, sign light and Whistle to control the vehicles on the road. This system requires the presence of a traffic policemen. This method requires human labor and it is time consuming.

## III. PROPOSED SYSTEM

The proposed system detect vehicle and classify it as an emergency or regular vehicle using deep learning and convolutional neural network.

## IV. METHODOLOGY

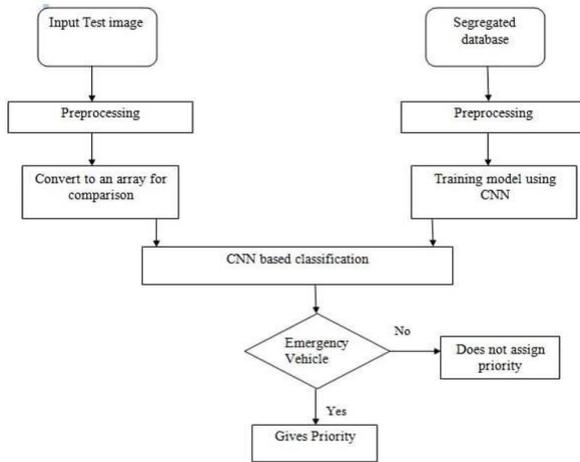


Fig no 1.2

### 4.1 DATA COLLECTION:

The Dataset of Emergency Vehicles was taken from UCI Repository dataset.

Three datasets are used in our system

Test - contains vehicle images for testing.

Train - contains vehicle images for training.

Sample submission – contains image names and shows whether they are emergency vehicles or not.

There are a number of existing datasets which have images of emergency vehicles

such as ambulance, fire engine, and police vehicles.

These datasets were generally collected for very specific uses with neural networks that were designed to classify vehicles based on certain characteristics. In our project we aim to classify 2 classes of vehicles i.e.; Emergency vehicle, Non-Emergency vehicle.

Machine Learning requires the ability to learn features automatically from the data, which is generally only possible when lots of training data is available especially for problems where the input samples are very high-dimensional, like images. The dataset as a whole contains 2000 images in which 1500 used for training and 500 used for testing.

### 4.2 PREPROCESSING OF DATA

Data preparation is required when working with neural network and Machine Learning models. In our project we use `get_transforms` and `image data bunch` functions for standardizing images. The images are stored in the jpeg format.

`get_transforms`:

To get a set of transforms with default values that work pretty well in a wide range of tasks, it's often easiest to use `get transforms`. Depending on the nature of the images in your data, you may want to adjust a few arguments, the most important being:

`do_flip`: if True the image is randomly flipped (default behavior)

`flip_vert`: limit the flips to horizontal flips (when False) or to horizontal and vertical flips as well as 90-degree rotations (when True)

`max_rotate`: if not None, a random rotation between `-max_rotate` and `max_rotate` degrees are applied with probability `p_affine`

`max_lighting`: if not None, a random lighting and contrast change controlled by `max_lighting` is applied with probability `p_lighting`

`max_warp`: if not None, a random symmetric warp of magnitude between `-max_warp` and `maw_warp` is applied with probability `p_affine`

`get_transforms` return a tuple of two lists of transforms: one for the training set and one for the validation set (we don't want to modify the pictures in the validation set, so the second list of transforms is limited to resizing the pictures). This can be passed directly to define a `DataBunch` object which is then associated with a model to begin training.

The defaults for `get_transforms` are generally pretty good for regular photos

- although here we'll add a bit of extra rotation so it's easier to see the differences. `ImageDataBunch`:

Before any work can be done a dataset needs to be converted into a `DataBunch` object, and in the case of the computer vision data - specifically into an `ImageDataBunch` subclass. This is done with the help of data block API and the `ImageList` class and its

subclasses. However, there is also a group of shortcut methods provided by ImageDataBunch which reduce the multiple stages of the datablock API, into a single wrapper method. These shortcuts methods work really well for: Imagenet-style of datasets (ImageDataBunch.from\_folder)

A pandas DataFrame with a column of filenames and a column of labels which can be strings for classification, strings separated by a label\_delim for multi-classification or floats for a regression problem (ImageDataBunch.from\_df)

A list of filenames and a function to get the target from the filename (ImageDataBunch.from\_name\_func)

A list of filenames and a regex pattern to get the target from the filename (ImageDataBunch.from\_name\_re)

bs (int): how many samples per batch to load (if batch\_size is provided then batch\_size will override bs). If bs=None, then it is assumed that dataset.getitem returns a batch.

num\_workers (int): how many subprocesses to use for data loading. 0 means that the data will be loaded in the main process.

valid\_pct to indicate the percentage of the total images to use as the validation.

set.ds\_tfms is a tuple of two lists of transforms to be applied to the training and the validation (plus test optionally) set.

fn\_col is the index (or the name) of the the column containing the filenames.

label\_col is the index (indices) (or the name(s)) of the column(s) containing the labels.

tfms are the transforms to apply to the DataLoader.

The size and the kwargs are passed to the transforms for data augmentation.

### 4.3 BUILDING A MODEL

Next step is to build our model using neural networks. Since we have our data ready so it's time to feed it into a model. We can do this by building a convolution neural network from scratch but doing this would be practically inefficient. So, we take the weights of a pre trained CNN model that has learnt to recognize features (certain kind of things e.g gradient, edges circles

e.t.c). Here we would be using a pre-trained ResNet152 Convolution Neural Net model, and use transfer learning to learn weights of only the last layer of the network. We use the `thecnn_learner` function for this.

### 4.5 IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools;
- Analyzing and manipulating the image;
- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hardcopies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

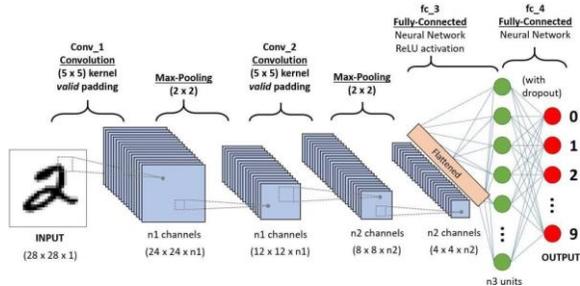
### 4.6 MACHINE LEARNING

Machine Learning can be considered as a subset of machine learning. It is a field that is based on learning and improving on its own by examining computer algorithms. While machine learning uses simpler concepts, Machine Learning works with artificial neural networks, which are designed to imitate how humans think and learn. Until recently, neural networks were limited by computing power and thus were limited in complexity. However, advancements in big data

analytics have permitted larger, sophisticated neural networks, allowing computers to observe, learn, and react to complex situation faster than humans. Machine Learning has aided image classification, language translation, speech recognition. It can be used to solve any pattern recognition problem and without human intervention.

#### 4.7 Convolutional neural network CNN

In Machine Learning, a Convolutional Neural Network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter-intuitively, most Convolutional Neural Networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series.



CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to over fitting data. Typical ways of regularization, or preventing over fitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using

smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

#### 4.8 LIBRARIES

##### Pandas

In computer programming, pandas are a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and timeseries. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

##### Fastai

fastai is a Machine Learning library which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard Machine Learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches. It aims to do both things without substantial compromises in ease of use, flexibility, or performance. This is possible thanks to a carefully layered architecture, which expresses common underlying patterns of many Machine Learning and data processing techniques in terms of decoupled abstractions. These abstractions can

be expressed concisely and clearly by leveraging the dynamism of the underlying Python language and the flexibility of the PyTorch library. fastai includes:

A new type dispatch system for Python along with a semantic type hierarchy for tensors

A GPU-optimized computer vision library which can be extended in pure Python

An optimizer which refactors out the common functionality of modern optimizers into two basic pieces, allowing optimization algorithms to be implemented in 45 lines of code

A novel 2-way callback system that can access any part of the data, model, or optimizer and change it at any point during training

A new data block API

### NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the *ndarray* object.

### Torchvision

Torchvision is a library for Computer Vision that goes hand in hand with PyTorch. It has utilities for efficient Image and Video transformations, some commonly used pre-trained models, and some datasets (torchvision does not come bundled with PyTorch, you will have to install it separately)

## V. SYSTEM ANALYSIS

Dataset for Image Detection For learning the features of the emergency vehicle and differentiating it from other vehicles the dataset is divided into 3 sections, namely: 1.) Training Dataset: It contains 439 images of emergency vehicle and 372 images of non- ambulance from different angles and different views so that the model can learn the features of

the emergency vehicle and differentiate it from all other vehicles. 2.) Validation Dataset: It contains 20 percent of the training dataset which is used to validate the accuracy of the model in predicting ambulance.

3.) Test Dataset: It contains 53 images which are used to calculate the accuracy and the effectiveness of the model. Ambulance detection using images is used to predict if an ambulance is coming or approaching the car from behind. This is being applied in order to move the vehicle aside and give the ambulance a proper and delay free commute system. The detection of emergency vehicle coming from behind is done using object detection in images in this module.

A camera is mounted on the rear end at top of the vehicle to record in the live happenings in the surroundings of the car. The live feed is then used to provide input to the module which then works upon the input to detect if there is any ambulance in the surrounding or not. The frames are taken from the live feed one by one and fed into the deep learning convolutional neural network created to process images and detect the ambulance in the images. The frame is resized into 32X32 dimensions which is the size on what the model is trained upon. The pixel values of the image are scaled in the range between 0-1 to provide ease to the model to do the predictions instead of the range of 0-255 and then the image data is converted into the required array format that the model accepts. The array is then fed into the deep learning model and the model performs its computations in order to learn the features required to differentiate the ambulance from other vehicles.

After the module performs its computation on the image fed into the network, the detection of ambulance in the image with the confidence level is given out as the result and if the ambulance is detected, then the car is either stopped or move aside to let the ambulance pass through, giving ambulance the priority of commute.

## VI. SYSTEM DESIGN & ARCHITECTURE

The embedded systems in a traffic signal can be programmed to accept an input from the detection unit whenever an emergency vehicle is detected and subsequently switch the signal to green from red. A

reliable and robust system that can accurately detect an emergency vehicle and fast track its flow through heavy city traffic is an asset to any Intelligent Transportation System or Smart City venture. Autonomous vehicles can also have built-in emergency vehicle detection capabilities to allow priority movement of ambulances, fire engines etc. In both use cases, it is essential to ensure that there are sufficient computational resources for the execution of the computer vision models. Both object detection as well as image segmentation differ from conventional image classification in the sense that they identify the location/coordinates of the object under detection. On the other hand, an image classifier would simply assign a particular label to image when the object it is trained to detect is found in the image. For the intelligent traffic signal application, a conventional image classifier would be ineffective as it is necessary to identify the lane in which the emergency vehicle is present, so as to switch the signal for that particular lane. An object detection model would be an ideal fit for this application. In the case of an autonomous vehicle, greater precision is required as the vehicle will have to maneuver itself based on the spatial extent of the emergency vehicle. In this case, an instance segmentation model, which traces the emergency vehicle by performing pixel wise classification, would be the best fit. Although the object detection model will generate a bounding box, it will be unable to provide the exact coordinates of the emergency vehicle.

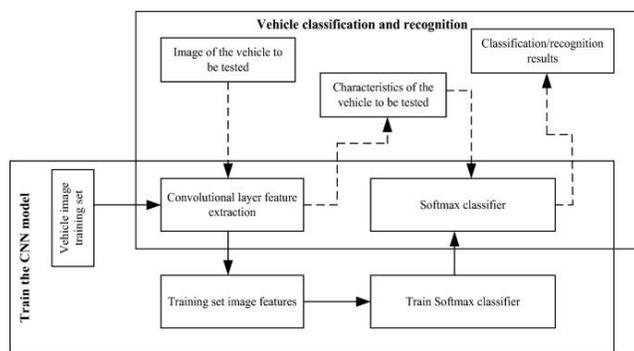


Fig. 1.4 System Architecture

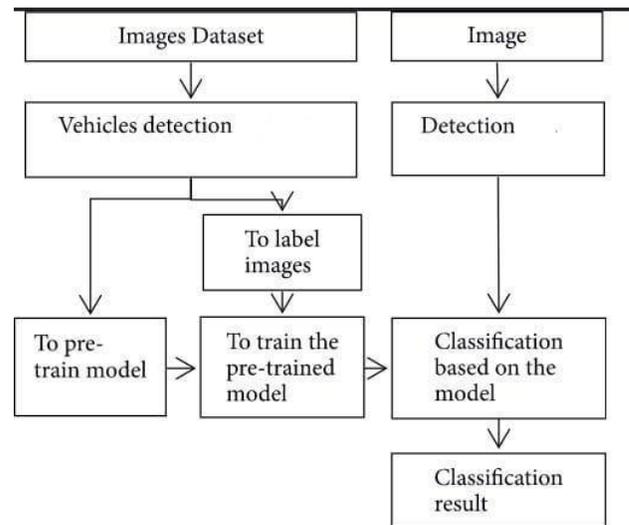


Fig 1.5

## VII. CONCLUSION

In this work, we have proposed a model that can detect emergency vehicles such as ambulance, fire truck, police vehicle on a heavy traffic road. Unlike western countries, Indian cities cannot think of having separate lanes for emergency purpose due to lack of road planning and infrastructure. With the lives of the patients depending on the speedy arrival of the ambulances to hospital, an alternative solution to the above problem is the need of the hour. Improvement in the quality of life along with substandard public transportation has resulted in spiraling growth of private automobiles. Moreover, the ambulances often get stuck at the traffic signals where all other vehicles try to squeeze in to all the available space so as to move ahead as soon as the signal turns green.

Our model will solve this problem. This project focuses on using image processing techniques as an efficient way of detecting emergency vehicles since this method reduces the chances of failures and helps in ensuring that the emergency vehicles have a hassle-free movement while trying to perform their duties. With this automated process, no human effort will be required to manually help such scenario. Our model has achieved impressive results in detecting and identifying emergency vehicles of all kinds.

## VIII. SCOPE FOR FUTURE DEVELOPMENT

Furthermore, research can be done to capture an image from a live video and detect the ambulance, as videos given as an input by the user. Our model can be embedded with CCTV to track emergency can and give priority in that road to pass the emergency can. Developing emergency response capabilities during the night and off-nominal weather conditions are challenging and as essential as response during normal weather conditions. Hence research into this is essential for a complete EV identification system. Since the shape of the vehicle is hardly visible in the night, the predominant features that can be used for vision-based identification are the flashing lights on the emergency vehicle. Thus, an extension of this work can also aim at developing algorithms to identify EVs during nighttime by capturing these flashing light features. This work has mainly focused on developing emergency vehicle identification using vision-based techniques. Emergency vehicles also output sound features that can act as a rich source of information. In crowded traffic scenarios, sound features are captured much before when the emergency vehicles are in the line of sight. Thus sound-based features when fused with vision-based techniques can also lead to more robust emergency vehicle detections.

## IX. BIBLIOGRAPHY

- [1] "Traffic Management for Emergency Vehicle Priority Based on Visual Sensing" by Kapileswar Nellore, Gerhard P. Hancke [2016]
- [2] "Front collision warning based on vehicle detection using CNN" by Junghwan Pyo, Jiwon Bang, Yongjin Jeong [2016]
- [3] "Implementing Smart Control of Emergency Vehicle Using Image Processing" by A.ArockiaSelvaraj, A.Richard, K.Saminathan, J.Vigneshwaran, R.Vasanthan [2017]
- [4] "Real-Time Vehicle Detection using Machine Learning Scheme on Embedded System" by Ju-Seok Shin, Ung-Tae Kim, Deok-Kwon Lee, Sang-Jun Park, Se-Jin Oh, Tae-Jin Yun [2017]
- [5] "Vehicle Detection and Localization on Bird's Eye View Elevation Images Using Convolution Neural Network" by Shang-Lin Yu, Thomas Westfechtel, Ryunosuke Hamada, Kazunori Ohno, Satoshi Tadokoro [2017]
- [6] "Vehicle Detection Using Alex Net and Faster R-CNN Machine Learning Models: A Comparative Study" by Jorge E. Espinosa, Sergio A. Velastin, and John W. Branch [2017]
- [7] "Vehicle Type Classification Using Convolutional Neural Network" by Kshitij Kaushik, Kunal Mali, Shrikant Dangi, Tushar Dalal, Pallavi Suradkar [2018]
- [8] "Smart Detection of Emergency Vehicles in Traffic" by Deepa, Navya Kumari, Manisha K, Manu Manjunatha, Kshama shetty [2018]
- [9] <https://www.kaggle.com/abhisheksinghblr/emergency-vehicles-identification>
- [10] "Development of Vision-Based Response of autonomous Vehicles towards Emergency Vehicles Using Infrastructures Enabled Autonomy" by Abhishek Nayak [2019]
- [11] "DeepCrash: A Machine Learning-based Internet of Vehicles System for Head-on and Single-vehicle Accident Detection with Emergency Notification" by Wan-Jung Chang, Liang-Bi Chen, Ke-Yu Su [2019]
- [12] "Emergency Vehicle Detection on Heavy Traffic Road from CCTV Footage Using Deep Convolutional Neural Network" by Shuvendu Roy, Md. Sakif Rahman [2019]
- [13] "A Hybrid Framework for Expediting Emergency Vehicle Movement on Indian Roads" by Dr. Rajeswara Rao, K.V.S Dr Minal Moharir, Abhishek Raman Kaushik S [2020]
- [14] "Leveraging Computer Vision for Emergency vehicle detection Implementation and Analysis" by Dr. Rajeswara Rao K.V.S, Kaushik S and Abhishek Raman [2020]

- [15]“Lightweight PVIDNet: A Priority Vehicles Detection Network ModelBased on Machine Learning for Intelligent Traffic Lights” by Rodrigo Carvalho Barbosa, Muhammad Shoaib Ayub, Renata Lopes Rosa, Demostenes Zegarra Rodriguez Lunchakorn Wuttisittikulkij [2020]
- [16]“Acoustic-Based Emergency Vehicle Detection Using Convolution NeuralNetworks” by Van-Thuan Tran and Wei-Ho Tsai [2021]
- [17]“Intelligent Traffic Control System” by Srikanth S, Srivatsa K, Venkata Prabakaran S, Revathy P [2020]