# ARTIFICIAL INTELLIGENCE BASED REAL-TIME ADAPTIVE TRAFFIC LIGHT MANAGEMENT SYSTEM

Thesis Submitted For the Award of the Degree of

DOCTOR OF PHILOSOPHY

**in**

**Computer Science and Engineering**

**By**


**Usha Mittal**

**Registration Number: 41600328**


**Supervised By**

**Dr. Priyanka Chawla**

Transforming Education Transforming India


**LOVELY PROFESSIONAL UNIVERSITY**
**PUNJAB**
**2023**

# DECLARATION

I, hereby declared that the presented work in the thesis entitled "Artificial Intelligence Based Real-Time Adaptive Traffic Light Management System" in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision Dr Priyanka Chawla. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

Name of the scholar: Usha Mittal

Registration No.: 41600328

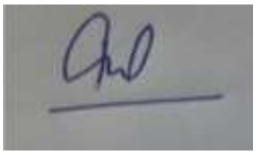
Department/school: School of Computer Science and Engineering

Lovely Professional University,

Punjab, India

# CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled "**Artificial Intelligence Based Real-Time Adaptive Traffic Light Management System**" submitted in fulfillment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Science and Engineering, is a research work carried out by Usha Mittal , 41600328, is bonafide record of her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



**(Signature of Supervisor)**

Name of supervisor: Dr Priyanka Chawla

# Abstract

Traffic management is becoming a critical problem for society as vehicle traffic volume rises. Due to static traffic management regulations on roadways, traffic flow may become congested as it has been growing on roads. An innovative and intelligent traffic control system is required to manage the traffic flow on roads, especially in developing countries. The intelligent system reduces the shortcomings of a fixed timer control system. Machine Learning and soft computing techniques can be utilized to optimize signal timing depending on traffic information on different lanes.

In this thesis, a literature review on various aspects like vehicle detection and classification, emergency vehicle detection techniques, and green signal optimization methods has been done. Based on published works, the methods and techniques utilized for designing intelligent traffic controllers have been evaluated. This study will aid in methodically disseminating results. Therefore, it helps researchers working in similar areas choose the most effective datasets and techniques for vehicle detection, emergency vehicle detection, and green signal optimization.

Various datasets have been extracted from different open-source libraries to perform the experimental work. Various models have been implemented on the chosen datasets for vehicle detection and classification and their performance has been analyzed. It has been observed that the proposed ensemble of the Faster R-CNN and SSD model outperformed the other existing models. Also, the results of the proposed model have been analyzed for traffic density estimation. For emergency vehicle detection, two techniques that are RFID and siren-based models have been used. An ensemble of fully connected layers, CNN, and RNN models have been implemented. According to the experimental findings, the proposed ensemble produced reliable results for the detection of emergency vehicles.

Finally, to optimize the green signal of the traffic controller, vehicle density information and the emergency vehicle's presence or absence are considered input parameters. An adaptive neuro-fuzzy inference system has been trained based on traffic density and flow rate of intersection. Experimental results proved that the proposed ANFIS model optimizes the green signal better than fixed-timer-based and fuzzy-based systems.

The suggested technique can be utilized to lessen junction wait times for passengers along with road congestion. Further, it also helps reduce fuel consumption and $CO_2$ emissions.

# Acknowledgement

This thesis would not have been possible without the assistance of many people. I want to thank them for their invaluable help sincerely. I want to express my gratitude to the almighty God for giving me the courage and strength to conduct this study.

I sincerely thank my supervisor, Dr. Priyanka Chawla, Professor of Computer Science and Engineering, Lovely Professional University, Punjab. Her extensive knowledge, profound insight, and wide experience have impacted me significantly. Her thoughtful advice and practical direction helped me accomplish my goal. She has consistently pushed me to conduct high-quality research. Her subject-matter expertise inspires me to put in more effort and provide more significant results. I could not have imagined a better supervisor and guide for my research work.

I want to thank the Doctoral Committee for keeping track of my research progress and for their insightful suggestions to make the work better and more effective. I am grateful and proud of Computer Science and Engineering, Lovely Professional University, for giving me all the necessary tools for successful research work. Additionally, I want to express my gratitude to my friends and colleagues for their unwavering assistance and support throughout my research.

I want to take this opportunity to express my gratitude to my family members, who are always there to provide me with moral and emotional support. By God's grace, this work allows me to make my family proud.

# Contents

## List of Figures

## List of Tables

# Chapter 1
# Introduction

A vital component of any country's economic growth relies on the conditions of road transportation. It impacts the rate, structure, and pattern of growth [1]. India is known for being the second-largest road network in the world, having a length of 5.89 million kilometers [2]. In India, 64.5% of commodities are moved through a road network, and about 90% of passengers use roads for traveling [2]. Road traffic has progressively increased with the connectivity improvement between towns, cities, and villages across the country. Automobile sales and freight transportation by road are rapidly increasing in India.

In today's society, efficient logistics and people transportation are critical to the social economy's success. However, the existing infrastructure that relies on conventional traffic management systems, such as the loop detector-based SCOOP system, breaks down due to the rise in societal needs for transportation. Traffic congestion increases the burden on people's daily lives [3]. According to a government report [4] published by the United States Department of Transportation, due to traffic congestion, there were more than 5.7 billion gallons of fuel wastage during 2000 in 75 big US cities, as well as a negative socio-economic influence of 3.6 billion hours of traffic interruption. By 2011, the costs had escalated to 5.5 billion hours of interruption, $121 billion in fuel wastage, and more than 25 million tonnes of automobile pollution [5]. As a result, a popular study area in recent years has been how to efficiently use current transportation infrastructure to reduce the conflict between transportation resources and demand for products and people mobility.

US Department of Transportation reports revealed that traffic congestion has three significant reasons. The first factor causing traffic congestion are incidents like construction zones, accidents, and bad weather. The second factor is traffic demand, which includes typical and unusual traffic patterns. The third cause is traffic management systems and physical bottlenecks under transportation infrastructure. Moreover, 40% of traffic congestion occurs due to bottlenecks, 25% from traffic incidents like accidents, 15% from bad weather conditions, 10% accounts for work zones, and the rest is affected by signal timings at intersections and uncertain events [6].

Congestion can occur due to an excess of the road's service capacity, a surge in cars, or a reduction in the road's throughput because of road accidents. Whatever the reason, when the traffic flow hits saturation, congestion immediately arises. Furthermore, a driver's rapid braking on a slick road can cause a breaking wave in the vehicle behind him, resulting in a significant delay. As a result of the protracted delay, traffic will get stagnant.

Consider the cities of Mumbai and Bangalore. Given in a traffic survey [7] of 416 cities across 57 nations, Banglore has been known for its worst traffic management, whereas Mumbai is following closely behind in the fourth position. A journey in Bangalore during a heavy traffic jam takes 71% longer. Similarly, it is 65% longer in Mumbai [8]. Improved traffic flow can reduce the number of accidents and passengers' travel time.

The state and central governments have adopted several ways to deal with this issue. Traffic light controllers have been deployed in the accidentally-prone areas, and laws have been enforced against all traffic violators.

## 1.1 Traffic Lights

The function of a traffic light is to serve as an indicator that uses a universal color code to indicate whether it is secure to drive, ride a bicycle, or walk at a road intersection, a zebra crossing, or any other location. Traffic lights for vehicles in India typically have three significant lights: a red light that signifies a stop, a green light that denotes a go, and a yellow light that denotes the vehicle is about to stop. In India, no particular signal is used for pedestrians crossing. The traffic signals have been advantageous to all commuters. It enhanced traffic flow and may have saved people time in addition to reducing the number of incidents. A simple traffic light is shown in figure 1.1.

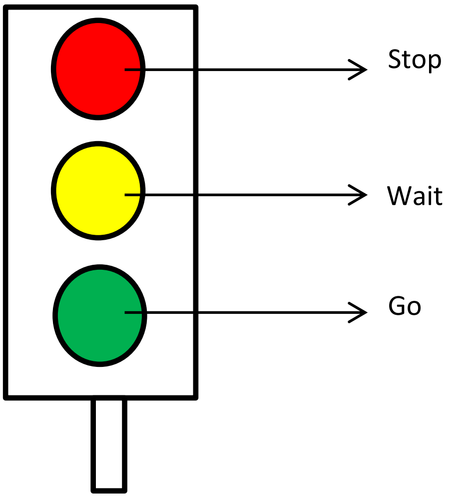



Figure 1.1: Basic Structure of Indian Traffic Lights

There are three standard systems for traffic control in use:

a) *Manual traffic Controlling*: As the name implies, manual traffic control requires human interaction. The traffic police officer is responsible for traffic control in a particular region. The traffic officers use a signboard, a sign light, and a whistle to regulate traffic.

b) *Standard traffic lights with fixed timers*: Traffic controllers with predetermined timers regulate traffic at intersections. The timer comes pre-programmed with a specific pre-defined value. Depending on the timer's value, the lights change from red to green automatically.

c) *Electronic Sensors*: Installing proximity sensors or loop detectors on the road is another leading technique. These sensors measure the amount of traffic on the route. Traffic signals switch to other states based on data collected from sensors.

Traditional procedures have several disadvantages. Manual traffic control requires considerable human resources. It is not practicable to have traffic police manually manage traffic in all regions of a city or town due to a lack of resources. As a result, there is a demand for a better, smart, and more intelligent traffic control system. Static traffic control employs a traffic signal with a countdown for each phase set in stone and does not respond to real-time traffic on the road. Accuracy and coverage are generally at odds since high-quality data is typically obtained utilizing novel and expensive technologies, which means that a limited budget will limit the options if using electronic sensors. Furthermore, as most sensors have a limited effective range, there is a need for many sensors to provide complete coverage.

## 1.2 Need for Traffic Management

Today's traffic controllers are not programmed and are susceptible to human mistakes. Because of overcrowding, increased daily travelers, globalization, and reduced or thin highways, traffic monitoring is a top priority in many countries. Traffic influx is being hampered by improper signaling system architecture and traffic rules violations, causing congestion. Pollution and global warming affect the inside city environment due to traffic congestion at various intersectional crossings. Commuters also pay a hefty price for long periods stuck in traffic because of excessive fuel usage. Every government is concentrating on traffic monitoring and network management challenges. The economy's productivity of a particular nation is directly impacted by better road networks allowing free traffic movement.

Fixed-time traffic signals lack the characteristics needed to accommodate traffic variations; instead, they have varying timings based on the time of day. The signal cycle is repeatable, and the timing between stages is adjusted for every cycle. Managing a specific intersection based on predetermined signals entails identifying the time scheduling of green, yellow, and red lights for every traffic stream, regardless of the vehicle's density approaching the intersection. The features and average traffic density for a certain period determine the signal duration.

Nevertheless, the addition of more traffic signals has a lot of unfavorable effects and challenges:

a) *Heavy traffic delays are caused by traffic lights*

   The number of automobiles on the road has increased, which has resulted in severe traffic congestion. The mornings, before work, and the evenings, after work, were when this occurred most frequently at significant crossroads. This problem primarily causes people to waste more time on the road.

b) *The road user must still wait even though there is no traffic*

   At particular intersections, there may occasionally be no traffic. Drivers must wait till the traffic signal turns green because it is still red. They will be held accountable if they run a red light.

c) *An emergency can be stuck in a traffic jam*

   Ambulances, fire trucks, and police cars regularly get stuck during traffic congestion, especially at junctions with traffic lights. This results from the waiting traffic for the signal to turn green. It is crucial since it can stop an emergency from developing into a complicated and potentially fatal situation.

In the last few years, technology has developed in every way imaginable. More quickly than ever before, science and technology are evolving. Artificial intelligence (AI) is changing our daily lives, starting with the obvious AI capabilities designed as assistants, like face unlocks criteria [9]. Furthermore, AI has made inroads into fields such as mathematics, cybernetics, medical science, neurology, engineering, philosophy, economics, education, psychology, and transportation logistics, to name a few. It can also be defined as the study and composition of intelligent agents aware of their surroundings and making decisions based on that knowledge. AI's fundamental goal is to create and build an automated system that can accomplish activities that people can do based on experience and observations. AI is the branch of computer science concerned with discovering and developing innovative and intelligent computers that respond like live beings.

The creation of "The Logic Theorist" marked the start of the modern AI revolution. It directs us to "Machine Learning." Machine learning's primary goal is to complete tasks assigned to it to address a specific issue by utilizing its existing knowledge and statistical data. The data must be reliable and consistent for learning to occur as quickly and effectively as feasible. Significant AI developments such as machine learning (ML), deep learning (DL), and neural networks (NN) enhance a machine's capacity to learn from data based on prior knowledge. [10–12]. Numerous machine learning applications are available today, including weather prediction, music and hotel recommendation systems, market analysis, facial recognition, speech, etc. When it comes to machine learning and artificial intelligence, mathematical logic and statistics are essential. Embedded systems are another important use of AI in which software is embedded in computer hardware [13].

Different governmental organizations request high-tech technology to oversee the country as technology advances. Transportation networking plays a significant part in a smart city, mainly in metropolitan centers, because it is divided into numerous diverse sectors. Most traffic congestion occurs in metropolitan areas, leading to increased air pollution and the danger of accidents. A smart city must have stronger administration that plans strategically to handle traffic issues. Manual traffic control has proven impractical in India due to rising urbanization.

Additionally, as the volume of data acquired from multiple traffic cameras grows, central monitoring systems are experiencing scaling challenges. According to a replication based on this, the local unblocking strategy improves road competence and minimizes crowded breakdowns in localized situations. A substantial increase in average ambient temperature and noise pollution was observed and reported amid heavy traffic. Improvements to geometry and traffic are crucial for the speedy and efficient flow of traffic needed in metropolitan areas. Climate change is also a primary factor for transportation management in developing countries like India, South Africa, Chile, and China.

## 1.3 Strategies for Traffic Management

Intelligent Transportation Systems (ITS) is the use of computing, information, and communications technology for the real-time management of vehicles and networks, including the movement of people, products, and services. It is mandatory to grasp the causes of mobility and how it is done to comprehend the transportation system and the necessity to model traffic flow. People and things must migrate between different locations due to people's daily activities. The transportation system provides the infrastructure and means to ensure that people and goods are in the right place at the right time to undertake the activities that result in products and services when the market needs them [16]. As the demand for road safety and connectivity between road networks has increased in recent years, ITS has attracted much interest [17].

To summarise, traffic monitoring hardware with wireless communication abilities can collect real-time data in existing traffic infrastructure. Then collected information will be sent to appropriate control agencies in an accurate and timely manner, helping them to make intelligent decisions to develop smart traffic controllers to reduce overall traffic congestion [18]. The present methods are broadly classified into two groups based on the numerous techniques of using the data acquired in ITS.

a) *The reactive approaches*

   This strategy primarily employs the ITS's real-time traffic data to build traffic management solutions for the present traffic scenario [19]. Traffic flow management, Adaptive traffic signal control, congestion detection, and other related approaches are examples of related methods.

b) *The proactive approaches*

   As reactive methods work and provide decisions based on real-time information, the reactive techniques emphases on analyzing the massive traffic information collected by ITS as well as extracting the data features of relevant control objects like movement characteristics of the participants of a specific traffic system, the characteristics and patterns of the traffic flow about time and so on.

Proactive strategies, as opposed to reactive approaches, can help ITS improve the overall efficiency of the transportation system (in terms of road safety, congestion prevention, and so on) as well as the travel experience of system users (such as passengers, pedestrians, etc.). This remark is justified since, as previously stated, the reactive methods are intended to react to an identified traffic-related issue to reduce the system's negative effect. But the reactive strategies will always be late in responding to observed traffic circumstances. Thus, traffic jams that occur due to uncertain circumstances and need to be timely handled will cause the failure of the traffic management system and cause high congestion. On the contrary, reactive approaches can only run the system in a steady state, even in traffic variations, due to a lack of sufficient traffic infrastructure. The proactive techniques can also predict upcoming traffic for a specific time.

The idea of automatic traffic signals was the only way to ensure traffic flowed smoothly. Preliminary studies indicate that if general-purpose machines were operated online and in real-time:

a) Using traffic information collected from several vehicle-detection sensors improves the timing of signals at each node, which is critical for overall system efficiency.
b) It also determines the proper temporal connection between nodes while accounting for the current traffic flow direction and speed.
c) It controls discrete signals to create archetypal circumstances.
d) To make sure everything is in functioning order, it inspects the traffic flow and signal operation.

Modern communication systems that rely on sensor tags to collect data and offer information on the current condition of the roads employ smart traffic signal controls to manage traffic signals. They make judgments based on priorities and are dynamic, so they work in real-time. The automated system uses this information to make judgments, such as when to activate each traffic light based on traffic volume on the roadways. Modern technologies like DL with Image Processing, OpenCV, intelligent controls, and AI are used by these automatic systems to make traffic-directing decisions, which traffic officers like police officers or traffic marshals frequently carry out. Other application areas include the management of freeways, intersection traffic signal control, and traffic and incident management.

## 1.4 Different Techniques for Traffic Lights

a) *Non-Electric gas-lit traffic lights*

   J. P. Knight suggested the idea created by Saxby & Farmer's railway signal engineers [20]. Three semaphore arms connected to red, green, and blue gas lamps make up the pattern. A traffic cop ran it. The police officer might face the wrong way as he had to flip the gas bulb manually. In this controller, the green light meant "Caution," while the red light meant "Stop." Even though the traffic control model was in operation, it had been out of commission for some time due to an explosion caused by a gas light leak.

*b) Electric traffic lights*

In 1912 [21], Lester Wire designed electric traffic lights with red-green color and a buzzer. The buzzer gave a warning to the drivers for color change. In 1920 [22], for the four-way intersection, three-color traffic lights were proposed by William Potts. The police constable had to operate the lights, which soon became obsolete.

*c) Automatic timers based traffic lights*

In 1922 [17], automatic timers were added by the company "Crouse Hinds ."Countdown timers were introduced in 1990 in traffic lights. It is the oldest and most widely used method to control traffic. This controller replaced the human job as each lane gets an equal period for green signal in a periodic manner. This model saves money by reducing the number of traffic officers required at the intersection. It also helps the pedestrian plan whether sufficient time is available to pass the intersection. This model has limitations, too, like it needs to be fixed on real-time data. Also, it doesn't provide priority to emergency vehicles. A timer-based traffic light is shown in figure 1.2.

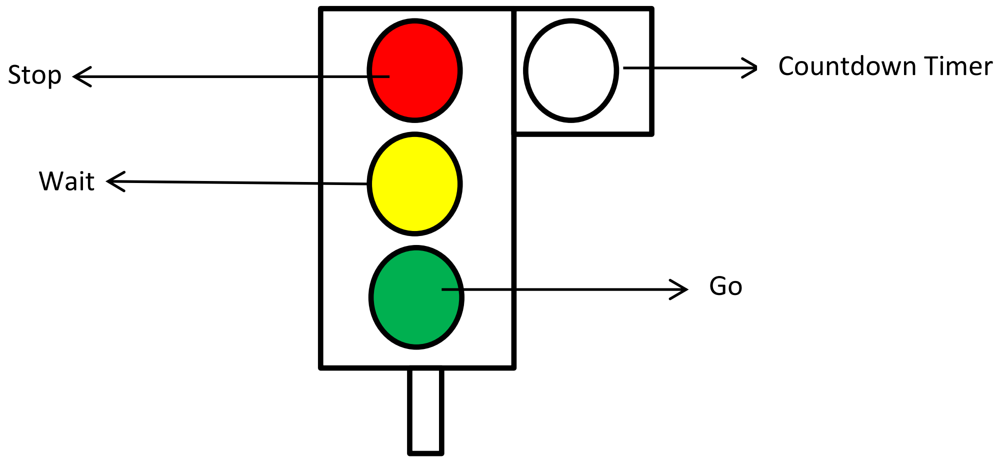



Figure 1.2: Fixed Timer Based Traffic Light with Countdown Timer

*d) Computerized traffic lights*

In 1950, computers made traffic lights a big turn where pressure plates were placed at the intersection to detect cars at the red light [22]. In 1952, pressure-sensitive detectors were used to measure inbound and outbound traffic. Using vehicle detection algorithms, one can estimate how long the green light will last based on vehicle density. In the 21st century, various computerized techniques like image processing, Fuzzy Logic, Artificial Neural networks (ANN), the Internet of Things(IoT), Wireless Sensor Networks (WSN), and hybrid systems [14] are used for the designing of intelligent and adaptive traffic lights.

e) *Image processing Based Technique*

Image processing is a technique for manipulating photos and videos to improve their quality or extract relevant information [15]. It includes three steps:

- Image acquisition
- Analyzing and manipulating the image
- Extracted feature or altered image

The traffic lights model using image processing was proposed in 1973 by the University of Tokyo engineers. Images from different roadsides are taken to analyze the current traffic scenario, like vehicle density, length of the queue, etc. Vehicles having high priority, like ambulances, fire brigade, and police vans, are detected and can be prioritized. Although this model works on real-time data due to environmental issues and quality concerns, a high-resolution camera may increase the overall implementation cost.

Some other challenges in this approach are:

- Which type of cameras should be used to take images so that images will be of high resolution and objects will be identified efficiently? Whether the camera can take images with a full resolution during nighttime or if the weather could be clearer.
- What should be the cameras' height so they can detect the vehicles from a long distance?

f) *Fuzzy expert system (FES) based traffic controllers*

In the Fuzzy expert system, fuzzy logic is used instead of Boolean logic. It consists of rules and membership functions to design the system and reason the data. Two categories of fuzzy inference systems exist the Mamdani-based Fuzzy Inference System (FIS) and the Sugeno-based Fuzzy inference system (FIS). Fuzzy logic is an integral part of artificial intelligence which works efficiently with incomplete data. It uses linguistic variables that are easy to understand by a human. It was initially used in 1977 for designing traffic controllers [23]. In this, fuzzy rules are designed based on the congestion on a lane, the number of objects approaching the green light, the presence of emergency vehicles, etc., and the output will be the extension in green signal timing to a particular roadside. From the literature survey, It is determined that the system performs better than the fixed timer system. But the major drawbacks are a lack of self-adaptability and self-learning capabilities. It makes decisions based on the pre-defined rules fed into the system. The architecture of the fuzzy inference system is shown in figure 1.3.

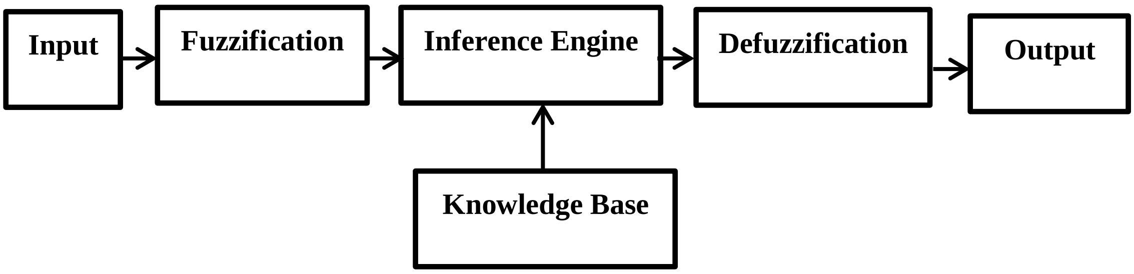



Figure 1.3: Architecture of Fuzzy Expert System

g) *Artificial Neural Network (ANN) based traffic controllers*

ANNs are computing systems inspired by biological neurons. These systems learn things from the examples rather than programmed by task-specific rules. It is having self-learning and self-adaptive ability [19]. It works well with real-time data. Multi-layer NN provides the acceptable solution, but as it is a mathematics-based model thus, it makes the system difficult to understand and analyze the computations that will be performed on hidden layers. The volume of the training dataset also dramatically impacts the efficiency of the ANN system. The system's efficiency will increase as the training dataset grows larger. The application of neural networks for traffic management was used in 1989 [24]. A basic architecture of the neural network is given in figure 1.4

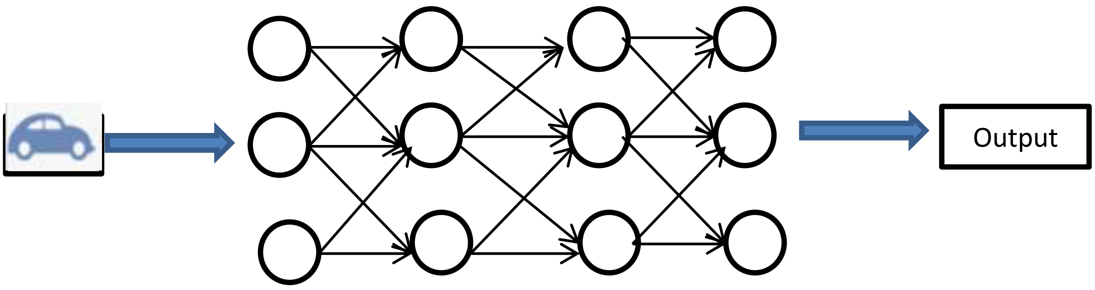



Figure 1.4: Architecture of Artificial Neural Network

h) *Wireless Sensor Network (WSN) based traffic controllers*

In wireless sensor networks, sensors are used to collect data about vehicles in each lane, the presence of ambulances, fire brigades, etc., and send the information to controllers [25]. Controllers have specific algorithms implemented in them. Based on the collected real-time data, decisions are taken, and the signal is allocated to a particular roadside. Delays in the elicitation, processing and sending of information to the controllers are the significant challenges of WSN, which affect the system's efficiency.

i) *Hybrid Techniques for traffic controllers*

Two or more technologies combined to overcome the limitations of each other are known as a hybrid approach. Shortcomings of the fuzzy system are the need for self-adaptation capability [26], the ANN's lack of inference, the WSN's delay problems, and the quality factor in image processing techniques. Fuzzy logic and NN can be combined to improve flow rate and vehicle waiting time at an intersection. A hybrid image processing and NN system can provide high accuracy in vehicle recognition and classification, density estimation, and emergency vehicle detection. Similarly, the hybrid system of the Internet of Things (IoT) and Wireless sensors help with vehicle detection, velocity estimation, flow rate determination, and emergency vehicle detection. Signal optimization results can be improved by hybridizing genetic algorithms and swarm intelligence [27]. Related literature review shows that hybrid systems provide high accuracy and performance.

In recent years, traffic management has extensively used video monitoring and surveillance systems for security, ramp metering, and providing travelers with real-time information and updates. Video surveillance systems can also be used to estimate traffic density and vehicle categorization, which can subsequently be utilized to manage traffic signal timers to improve traffic flow and reduce congestion.

## 1.5 Benefits of Smart Traffic Management System

Many traffic-related problems arise due to fixed-signal timer traffic control systems at crossings. They don't modify the phase sequence or length. Increased demand for road capacity necessitates innovative traffic control methods, which can be found in the ITS discipline.

The smart city paradigm combines information and communication technology to increase the efficiency of various city services. One of the primary industries that make citizens' lives easier is transportation. Many drivers, by nature, want to get to their destination as fast as possible. Arriving sooner reduces the travel time of automobiles on the road, reducing $CO_2$ emissions and preserving the city's green environment. Several researchers have devised a clever and efficient method for scheduling competing traffic flows at signalized road intersections. These tactics focus on reducing the length of traffic queues at intersections or minimizing the average delay per car.

Processes or procedures that do not require human interaction are called "automation." Sensors provide critical traffic data, and a programmed controller determines the system output. A programmable controller controls the actuator that turns on and off the lights.

An intelligent traffic light is an automated system that creates different traffic signals based on real-time road conditions and data from sensors located at various locations along the road and nearby intersections in the context of traffic management.

An intelligent traffic system includes auxiliary peripherals such as sensors, detectors, communicators, and other equipment to regulate traffic flow. The term "intelligent" refers to traffic signals' ability to adapt and adjust to current road circumstances using attached peripherals and respond appropriately in each situation.

The following are the components of an intelligent traffic control system:

a) *A central control system*

   The central control system serves as the framework for the traffic control system. The system includes traffic lights, signals, cameras, and queue detectors. The AI-based system can analyze real-time data by collecting information from 3D AI cameras and queue detectors with computer vision capabilities. The AI system assists in transmitting the best data to regulate the operation of traffic lights and signals for the smooth flow of traffic.

b) *Smart signal light*

   Intelligent traffic lights and signals can reduce congestion and travel time at intersections. The intelligent controller can manage the congestion and allow the traffic to pass without a predetermined scheduling plan.

One of the most obvious benefits of deploying advanced technology in traffic management is the ability to control and reduce traffic congestion and accidents in metropolitan areas. Data can be collected regarding traffic flow, climate, weather, and other factors using the sensor system deployed on the road surface. The data will be analyzed and processed using a computer system, and the results will be available to the public through some application. Drivers can choose safe traffic routes by knowing the current traffic situation on the road.

As a result of the technological revolution, intelligent automated systems are displacing old operating approaches. In major cities worldwide, an intelligent traffic management system gives you an advantage by offering safe public transportation, stiff penalties for disobeying traffic laws, and intelligent traffic congestion solutions. IoT, AI, computer vision, supervised machine learning, and big data are just a few of the advanced technology solutions that are helping to solve traffic management problems in real time. Smart roads, smart highways, smart street lighting, and computerized traffic signaling are all part of an intelligent traffic management system.

## 1.6 Gap Analysis

Based on the study of various traffic management systems, some of the research gaps that have been identified are as follows:

a) *Video/Image capturing of traffic scenes at odd times and in bad weather conditions*

Most research uses standard visible cameras at intersections to take traffic information. But these cameras cannot provide accurate and high-quality videos/images at odd times and in bad weather conditions. Thus, more information is needed to underestimate or overestimate the actual vehicle density, which leads to the undesirable green signal timing. In the proposed model, thermal images are utilized rather than using standard visible cameras.

b) *Estimation of vehicle density on traffic roads*

Previous works used sensors, loop detectors, and image processing techniques to estimate vehicle density. Image processing methods must provide satisfactory results due to the poor quality of the captured images/videos, and visible cameras must provide accurate and sufficient information at odd times and in bad weather conditions. In most studies, traffic density was computed in terms of the number of vehicles present or the area covered by vehicles available on the road. In the proposed approach, density estimation is done in terms of the total number of units available on the roads. Vehicles are classified into six categories, and each type of vehicle is assigned a unit value. Unit value depends upon the size and shape of the vehicle type.

c) *Prioritizing the emergency vehicles like (ambulances, fire trucks, and police vans) at the intersection*

Most earlier studies considered only ambulances for priority and optimizing green signal timing. However, fire trucks and police vans are a high priority and lifesaver vehicles. Thus, while providing the green signal to a particular road, the presence of emergency vehicles is also checked in the proposed model. If more than one type of emergency vehicle is present, priority will be given to the highest priority vehicles, and lesser priority vehicles will be considered after that.

d) *Coordinated control intersections*

From the literature, it has been analyzed that optimizing green signal timing is very beneficial. But it will be more helpful if adjacent intersections will also be coordinated. It will reduce the overall waiting time of passengers and reduces fuel wastage. In this study, green signal optimization is performed on stand-alone and coordinated intersections.

## I.7 Research Objectives

This study's primary goal is to design an intelligent and artificial intelligence-based traffic light management system. To accomplish this task following objectives were proposed.

a) Detection of vehicles on lanes and classifying them into different categories like 2-wheeler, 4-wheeler lightweight, 4-wheeler heavy vehicles, and 6-wheeler vehicles using convolutional neural network.
b) Detection of high-priority or emergency vehicles such as fire brigade, ambulances, police vans, etc., and prioritizing them in crossing the intersection using RFID.
c) Vehicular density calculation on the current lane as well as the adjacent lane from the data collected in objective 1.
d) Determining the green signal timing of each phase depends upon density using a hybrid neural network system and fuzzy logic.
e) Sending traffic information from the present junction to the adjacent junction for managing traffic signals using a wireless sensor network.

## 1.8 Major Contributions of the Thesis

Significant contributions made by the thesis are summarized as follows:

a) A vehicle detection and classification model based on ensemble learning has been proposed. For detection and classification, thermal and visible images are used as input. Thus, the proposed model can detect at night and in bad weather conditions like fog, rain, etc.
b) Emergency vehicles like ambulances, fire trucks, and police vans have been identified using two different methods. The Proposed model prioritizes an ambulance more than fire trucks and police vans. Thus, if, at the same time, more than one emergency vehicles are present, these are prioritized according to the given preference. Thus, the proposed model can reduce the waiting time of high-priority vehicles at intersections.
c) A density estimation method is introduced based on the number of vehicles on the road. The algorithm utilizes the unit values assigned to each vehicle type to compute the traffic density.

d) An adaptive fuzzy inference system is utilized to optimize the green signal timing at an intersection. The suggested methodology can reduce delays, reduce overall vehicle wait times, and increase intersection throughput.
e) For signal optimization at the coordinated intersections, techniques to transfer traffic information at an adjacent intersection are proposed. The methods are based on the distance between the adjacent intersections.

## 1.9 Thesis Outline

The thesis has been organized into eight chapters. A brief overview of all the chapters is as follows:

**Chapter 1** covers the introduction of traffic light controllers, highlighting the requirement of traffic light controllers and different types of traffic light controllers. Different technologies have been discussed in building intelligent and adaptive traffic light controllers. Strategies for traffic management are presented in detail. The benefits of using intelligent traffic light systems are also given in this chapter. Finally, the chapter concludes with the gap analysis, research objectives, contribution of the thesis, and thesis organization.

**Chapter 2** includes a detailed literature review on three main aspects: vehicle detection and classification, emergency vehicle detection, and green signal optimization methods. The review reported in this chapter is conducted by locating relevant research studies from well-known electronic resources and the most important conferences in the field. The literature review of vehicle detection and classification is based upon various parameters like extraction from images or videos, type of cameras used, models and techniques utilized, datasets considered for training and testing the models, metrics considered for evaluation of models, and performance achieved. Similarly, a survey of different methods of emergency vehicle detection has been done. Finally, a detailed study is done for optimizing green signals using various techniques based on fuzzy logic and its hybrid systems. The percentage of the status of the research work for vehicle detection and classification, emergency vehicle detection, and green signal optimization has been presented in the form of pie charts which benefits the researchers to know about the state-of-the-work carried out in the stated areas.

**Chapter 3** explains the basic concepts of classification, localization, and detection. It also covers the basic concepts of CNN models and their different types of layers. A comparison of visible images and thermal images is also given in detail. The concept of transfer learning is presented and explained in how it improves the performance of the models. This chapter also contains the different datasets taken for the implementation of vehicle detection. An introduction to the deep learning architectures utilized for implementation is also discussed. Further, the chapter describes the proposed vehicle detection models' overall design and implementation. The method of traffic density estimation is also presented.

The proposed vehicle detection model is an ensemble of Faster R-CNN and SSD models. On the four separate datasets, the suggested model is trained and tested. Images from the datasets are annotated, and vehicles are classified into six categories. An algorithm is proposed based on bounding box coordinators and confidence scores to obtain the final detection. Traffic density estimation is performed regarding the number of units on the roads. Different value of a unit has been decided for each type of vehicle.

**Chapter 4** describes the introduction to RFID technology, the components of RFID, and the different types of RFID. Various types of RFID types based on wavelength are also presented. Three types of emergency vehicles are considered for detection: ambulances, fire trucks, and police vans. Priority to the type of emergency vehicles is defined. A Python simulation-based system has been implemented to detect and prioritize emergency vehicles.

**Chapter 5** presents the concepts of sound-based detection and different features extracted from sound signals for object detection. An introduction to the recurrent neural network and its associated parts is also explained. The dataset of the siren sounds has been taken from the Google Audioset library. The sound files extracted are processed to extract meaningful features. Based on the extracted features, three deep-learning models are trained.

The first model is designed using fully connected layers. The second model consists of convolutional layers, max-pooling, and drop-out layers. LSTM layers are utilized in the third architecture. Methods of selecting an optimal number of layers and parameters in each model are also presented in detail. Finally, an ensemble-based model based on optimal configurations of the three models has been implemented.

**Chapter 6** explains the concepts of optimization and different techniques for optimization. An introduction to fuzzy logic and an adaptive neuro-fuzzy inference system is presented. A model based on ANFIS is implemented to optimize the green signal timings and prioritize emergency vehicles. The traffic density of the current road and adjacent road, as well as the traffic flow of the intersection, is considered input parameters of the proposed model. The wireless sensor networks and AWS S3 service send traffic information from one intersection to another.

**Chapter 7** covers the results from the proposed models implemented in this research to design an intelligent and smart traffic light controller. The performance of the vehicle detection and classification model is analyzed using precision vs. recall curve, accuracy, and mAP metric. Precision vs. recall results of the proposed model analyzed on four datasets are explained using line graphs. mAP analysis is given in the tabular form, and comparative analysis is shown using line graphs. From the experimental results, it has been concluded that the proposed model outperforms compared to its base estimators. Also, the proposed model performs better on thermal than visible images. A comparison between the proposed model and the previous studies is also provided. The traffic density computation results further demonstrate the proposed model's superiority.

The emergency vehicle detection results based on RFID have been shown by considering different cases of the arrival of emergency vehicles towards the intersection. When multiple emergency vehicles of the same type and different types are approaching the intersection, the proposed system can efficiently detect and prioritize them.

The performance of the sound-based emergency vehicle detection results is done by computing the accuracy and inference speed of the model. From experimental results, it has been observed that the proposed ensemble and RNN-based model outperform the other two models. A comparative analysis of the proposed model is also given based on machine learning models and other existing studies. Although the time taken by the proposed ensemble model is higher than other models, it provides acceptable results.

Finally, the green signal provided by the proposed model is compared with the fuzzy-based system and fixed timer-based controller. The proposed system also considers the presence or absence of emergency vehicle on the road and prioritize the roadside where the emergency vehicle is present. Results are discussed by considering the different cases to find the order in which the proposed model serves roadsides around the intersection. According to an analysis of the experimental data, the proposed model performs better than fuzzy-based and fixed-timer-based systems.

**Chapter 7** concludes the research given in the thesis and discusses the consequences for the future. This chapter concludes that the system's results are promising and can be used to benefit society. The system's performance can be improved in the future by considering the road's width and the vehicles' speed. We can also optimize the green signals using a hybrid model of genetic algorithm and fuzzy logic.

## Chapter Summary

In this chapter, the introduction to traffic congestion causes, the need for traffic management, and strategies to manage traffic have been documented. Since the proposed model is based on intelligent and AI-based techniques, an introduction to different AI-based traffic controllers has been described. This chapter explains the components of an intelligent traffic controller and the benefits of a smart and adaptive traffic management system. This chapter also highlights the gap analysis and the objectives that were framed for this thesis. In the end, the thesis outline has been documented in this chapter.

# Chapter 2
# Literature Review

Traffic control signals are signaling devices deployed at intersections to manage traffic flow. The first traffic signal in the world was only operational for a short time. In 1868, J.P. Knight [20] proposed gas-lit traffic lights, and railway signal engineers of Saxby and Farmer designed it. It was composed of red and green gas lamps connected with three semaphore arms operated by a police officer. It successfully controlled the traffic, but it burst on 2 January 1896 due to a gas leakage of the lantern and injured the police constable operating it. In 1908, kerosene lamps based traffic lights were used in Toledo, Ohio [21]. The red light indicated "Stop," and the green for "Go." Before altering a traffic light, a traffic officer blew his whistle to notify commuters of the change. This model was used for the first two decades of the 20th century in the United States.

A police officer in Paris built a traffic control system in 1912 controlled by a rotating, four-sided metal box with the words "Stop" and "Go" printed on it. Lester Wire proposed the first electric traffic lights in 1912, in which red-green signals were used [21]. In 1914, electric traffic lights with buzzers were used to warn of color changes [22]. In 1920, William Potts, in Detroit, Michigan, designed the first traffic lights for four-way intersections and three-color lights [22]. In 1922, automatic timers were used in traffic lights by the company Crouse Hinds [17]. Cities replaced traffic police with this model, saving money. Semaphores and towers had been abandoned by 1930 because they needed to be bigger, the semaphores too small, and they were impossible for travelers to see at night [17].

In 1950, the rise of computers gave a big turn to traffic lights in America. By analyzing the waiting vehicles at the red light using pressure plates, the duration of the red light was determined at the intersection [21][22]. With the advancement in technology, vehicle detection algorithms made traffic lights easier. Countdown timers were developed for traffic lights in 1990 [21]. Timers help drivers and pedestrians by allowing them to analyze the time they need to wait at intersections and for pedestrians to cross the street. By this time, researchers had introduced image processing, fuzzy logic, sensors, NN, etc., in designing traffic lights. After the 1990s, with the advancement in AI technologies, numerous studies have been conducted to improve traffic light management systems.

In this work, mainly three tasks are performed to achieve the defined objectives that are as follows:

[1.] Detection and classification of vehicles and traffic density estimation.

[2.] Detection of emergency vehicles detections, that is, ambulances, police vans, and fire trucks.

[3.] Optimization of green signal timing and sending traffic information from one intersection to another.

## 2.1 Research Methodology

The research methodology consists of a philosophical analysis of all the assumptions associated with the particular field of study. The demand for researchers to thoroughly and objectively describe all available knowledge about a phenomenon raises the need for reviews. It could be done to derive more general conclusions about a phenomenon or as a stepping stone to more research projects. Generally, it includes the concepts of different phases, models, and qualitative and quantitative techniques. The review is conducted in this study by planning, conducting, and reporting the review, as shown in figure 2.1. The stages of this literature survey are to create a framework for the review process, execute the survey, investigate review results, record the review results and explore various research challenges.

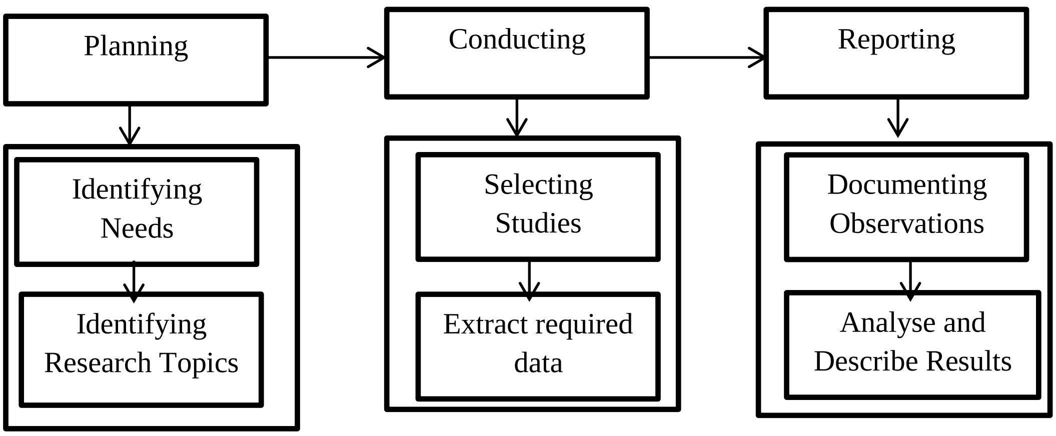



Figure 2.1: Overview of the research methodology

### *2.1.1 Planning*

The planning process begins with determining the requirements for conducting the literature review. The top conferences and electronic databases related to intelligent and adaptive traffic light management are considered to conduct a literature survey. Then, research sub-topics are further identified, and valuable research articles are extracted based on that.

The electronic databases of journals, conferences, and magazines like Google Scholar, ACM Digital Library, IEEE Explore, Springer, and Science Direct are explored to start with the search process. The primary aim of this survey is to find and categorize the existing literature emphasizing vehicle detection and classification, emergency vehicle detection, and green signal optimization. Major sub-topics of the literature survey are shown in figure 2.2.

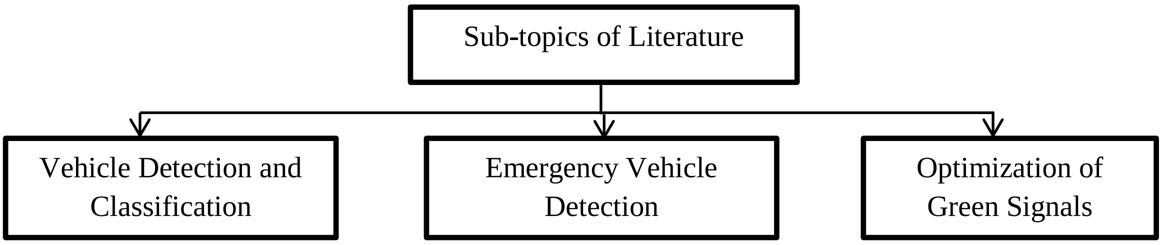



Figure 2.2: Topics of Literature Survey

### *2.1.2 Conducting*

This stage involves selecting the studies and extracting the required data. Selecting studies aims to choose the relevant articles considering recent technologies and methods. It covers research papers from symposiums, conferences, journals, workshops, and magazines.

### *2.1.3 Reporting*

This stage involves documenting observations, analyzing the extracted studies' findings, and reporting the final results. The survey observations have been given in the form of pie charts and tables by choosing appropriate factors and parameters of findings and methods utilized in the previous studies.

## 2.2 Literature Review Based on Vehicle Detection and Classification

The first step for designing a traffic light controller is to estimate the traffic volume at a particular intersection. It can be done by recognizing and detecting the number and types of vehicles on the road at a particular time. Nowadays, Methods based on ML and DL are frequently employed for the same.

Ozkurt et al. (2009) [28] utilized NN to develop a vehicle classification and traffic volume estimation model. The author collected data from the Istanbul Traffic Management Company (ISBAK) to conduct the study. In this study, work was carried out by designing three components that were moving object detector (MOD) for the detection of moving vehicles, a vehicle identifier (VI) to identify the type of vehicle, and a traffic density calculator (TDC) to compute the density of vehicles. The suggested model provided an accuracy of 94%.

Suryanto et al. (2011) [29] used the generalized Hough transform technique for tracking objects. In this, the spatial color histogram model was used for object representation. It was concluded from the results that the proposed work successfully tracked the objects even when they were in the same background with different sizes.

Kembhavi et al. (2011) [30] used color probability maps, Histograms of Oriented Gradients (HoG) as well as Pairs of Pixels to capture color-related information about the vehicles and their surroundings. Integration of these methods produced a high-dimensional feature set. The Partial Least Squares method was used to reduce the feature set's dimensionality and improve the proposed model's overall efficiency. The suggested model was compared with previous methods on the Google Earth San Francisco data set and Overhead Imagery Research Dataset (OIRDS) and provided comparable results.

Arrospide et al. (2012) [31] evaluated the performance of symmetry features with Bayesian classification for verifying the vehicles. The symmetry results via the Kolmogorov-Smirnov test achieved an accuracy of 80%. The author also evaluated the HOG descriptor and proved that it had high between-class separability compared to symmetry. HOG-based configuration achieved an accuracy of 92.48%, and it was highest, 96.94%, considered with front close/middle features.

Arrospide et al. (2013) [32] proposed and evaluated the performance of log-Gabor functions which was based on the previous Gabor filter. The newly defined log-Gabor filter was better than the Gabor function. Experiments were performed with both filters, and results showed that log-Gabor filters had an accuracy of 95.84% at a scale of 4 while only 95.14% with a Gabor filter.

Zhang et al. (2013) [33] used the Gabor wavelet transform and Pyramid Histogram of Oriented Gradients (PHOG) to extract features from vehicle images. The author proposed a cascade ensemble-based method with a rejection scheme to consider the scenarios in which no decision was taken if ambiguity occurred. The first ensemble consisted of k-nearest neighbors (kNNs), support vector machine (SVM), multi-layer perceptrons (MLP), and random forest. Similarly, a collection of base MLPs was integrated into the second ensemble with a rotation forest scheme. A rejection scheme was implemented in both ensembles by considering the consensus degree from majority voting to a confidence score and by restricting the classification of ambiguous samples if the value of the consensus degree is less than a threshold. The final category of the vehicle was decided by dual majority voting from both ensembles. A dataset from the local police department was collected to carry out the results. The proposed model provided an accuracy of 98.65% with a rejection rate of 2.5%.

Arrospide et al. (2014) [34] explored the combination capabilities of multiple techniques like SVM, HOG, and Gabor filters and designed an ensemble of classifiers. The proposed approach was evaluated on the GTI vehicle image dataset, and it provided an overall accuracy of 98%. The suggested method demonstrated that the fusion of classifiers, as opposed to the best single feature-based classifier, was highly advantageous for vehicle recognition and produced an additional gain of about 3%.

Qi et al. (2014) [35] used sparse representation-based classification and a post-processing method (boundary box shrinking) for pedestrian detection. The sparse coefficients were computed with two types of dictionaries. The detection was accomplished by assigning the given image to the class, which reduced the residual between the given input and the corresponding approximation. Experimental results were performed on the LSI dataset using three different feature extraction methods where the HOG feature demonstrated the best performance. The boundary box shrinking approach decreased the log-average miss rate.

Dollar et al. (2014) [36] suggested three different visual recognition systems and demonstrated pedestrian and general object detection results using rapid feature pyramids. The approximation was valid for images with broad spectra and failed for narrow band-pass spectra images.

Karpathy et al. (2014) [37] investigated CNN's effectiveness at classifying large-scale videos. The author claimed that CNN architectures outperform approaches based on features in learning strong features from sparsely labeled data. On the UCF-101 Action Recognition dataset, the author investigated the model's generalization performance while keeping the top layers and saw appreciable performance gains.

Simonyan et al. (2014) [38] presented a deep convolutional neural network with up to 19 layers for large-scale classification. The author suggested that the increased depth of CNN had improved the overall classification accuracy, and the proposed model achieved state-of-the-art performance on the ImageNet challenge dataset. The proposed model was evaluated using top-1 and top-5 errors and scored top-5 test error as 6.8%, which was the lowest among other configurations.

John et al. (2015) [39] presented a two-phase pedestrian detection method. The detection process first extracted candidate pedestrians from the infrared images. The images were segmented using adaptive fuzzy C-means clustering. Then, potential pedestrians were trimmed based on traits of human posture. The binary classification was carried out, and pertinent characteristics were learned simultaneously using a convolutional neural network (CNN). The proposed method yielded better detection accuracy with reduced computational accuracy.

Zangenehpour et al. (2015) [40] presented an approach in which the HOG method was used for feature extraction and SVM to classify moving objects from crowded traffic scenes. The dataset was collected from different intersections to conduct the study. In the proposed method three steps were used. Firstly moving objects were detected and tracked from videos. Then objects were classified based on their appearance, and finally, the computation of the probability of belonging to each class depended upon appearance and speed. The developed technique gave an overall accuracy of greater than 88%. The author also claimed that classification accuracy was good for vehicles and less for cyclists and pedestrians.

He et al. (2015) [41] evaluated the performance of SPP-net for object detection on ImageNet 2012, Pascal VOC 2007, and Caltech101 datasets. First, feature maps of the whole image were computed only once and then collected from random positions to create fixed-length feature sets. The author claimed that the SPP model performed faster than the R-CNN model and yielded better accuracy on the PASCAL VOC 2007 dataset.

Angelova et al. (2015) [42] presented a new real-time approach to object detection that exploited the efficiency of cascade classifiers and fast features with the accuracy of deep neural networks. The proposed algorithm worked at 15 frames per second in real time. The given method had an average miss rate of 26.2% on the public Caltech Pedestrian dataset.

Girshick et al. (2015) [43] developed a model which worked on single-stage training and utilized multitask loss. During the training, the model parameters were updated, and no memory was required to extract the features. The author claimed that the proposed model trained the VGG16 model nine times faster than R-CNN and ten times from SPPnet. The suggested model also provided a better mAP value on PASCAL VOC 2012.

Dong et al. (2015) [44] utilized semi-supervised CNN for vehicle classification. A sparse Laplacian filter was used to collect the required features of vehicles from the unlabelled dataset. Softmax classifier was trained on the labeled dataset using multitask learning to get output. The author built his vehicle dataset, the BIT-Vehicle dataset, for model evaluation.

The given model achieved the highest accuracy of 96.1% in the daytime and 89.4% at night light.

Fan et al. (2016) [45] evaluated the performance of the Faster R-CNN model on the KITTI dataset. Performance was evaluated at different scales, and the numeral of recommendations and absolute configuration was selected iteratively. The proposed configuration achieved a maximum average precision of 95.14% while it was 95% for default faster R-CNN.

Fu et al. (2016) [46] integrated current computer vision approaches to gather information to assess thermal sensors' speed, identification, and classification capabilities in various temperature and illumination conditions. Data using thermal and visible cameras were collected simultaneously from various places. The author claimed that visible cameras had a better performance than thermal during the day, but thermal cameras outperformed during low visibility and shadow cases mainly for cyclists and pedestrians. Also, the speed performance of thermal cameras was constantly better than visible cameras in the daytime and at night.

Redmon et al. (2016) [47] presented a regression-based method for object detection and classification, which separated bounding boxes (BB) spatially as well as related class probabilities. Only one NN was used to predict BB and the probability of classes from the whole image in one phase and optimized network performance. The author evaluated the proposed model on three public datasets: PASCAL VOC 2007, Picasso, and People-Art. The author claimed that the proposed network could process the images at 45 frames per second in real time.

Liu et al. (2016) [48] presented an approach known as a Single Shot Detector (SSD) in which BB of output space was discretized into several predefined boxes of various aspect ratios and scales. The proposed model made predictions by generating scores based on the object present in the predefined box and made appropriate alterations to the boxes to match highly with the object's shape. The author claimed that the proposed model was more straightforward than available models as it did not use a proposal generation network or feature resampling phase. The proposed model was tested on PASCAL VOC, COCO, and ILSVRC datasets, achieving an mAP of 74.3% on the PASCAL VOC dataset. Additionally, the proposed model performed better than Faster R-CNN.

Kong et al. (2016) [49] developed a deep hierarchical model named HyperNet for dealing with both region proposal generation and object detection. The proposed model aggregated hierarchical features maps collected from hyper features and compressed into uniform space. The hyper feature deeply incorporated the semantics and high-resolution features of an image and helped in proposal generation as well as object detection by end-to-end training scheme. The author claimed that the proposed model achieved the highest recall and accuracy on the PASCAL VOC dataset when 100 proposals per image were considered. Also, the proposed model processes five frames per second (fps) on Graphics Processing Unit (GPU).

Bell et al. (2016) [50] presented a model named as Inside-Outside Net (ION) in which inside and outside information of the region of interest (RoI) was extracted. Spatial Recurrent Neural Networks were used to integrate the RoI's outside contextual information. And skip pooling method was utilized to capture inside's information at varying scales and abstraction levels. Experimental results depicted that the proposed model had 77.9% mAp on PASCAL VIC 2012 dataset and 33.1% on the MS COCO dataset.

Dai et al. (2016) [51] developed an object detection system in which position-sensitive score maps were used to resolve the issue of translation invariance in classification and detection problems. The author tested the proposed model on the PASCAL VOC dataset and achieved an mAP was 83.6% when used with ResNet-101 architecture.

Kang et al. (2017) [52] proposed a multi-stage pipeline method for detecting objects from videos (VID) based on a still-image detection strategy. The dataset from the YouTube videos was collected to conduct the study. The author developed a novel temporal CNN to deal with temporal consistency and proved that performance was consistently improved over still-image object detection.

Pinheiro et al. (2016) [53] presented an approach known as SharpMask based on a top-down refinement method for performing object segmentation with augmented feed-forward networks. With the proposed top-down and bottom-up architecture, high-fidelity masks were generated effectively. It was a two-phase model. During the first phase, coarse 'mask encoding' was returned, and results were refined in the top-down phase by using features at successively lower layers. The author claimed that the proposed method improved the accuracy by 10-20% in average recall for multiple configurations.

Dai et al. (2016) [54] proposed a model for instance-aware semantic segmentation in which multitask networks were cascaded. The proposed model consisted of three sub-models that differentiate instances, identify masks, and classify objects. Cascaded networks could share their convolutional features. The model was tested and evaluated on PASCAL VOC and MS COCO datasets and provided acceptable results.

Zhou et al. (2016) [55] developed a deep learning model for detecting and annotating vehicles that is DAVE. It comprised two CNN models: first, to extract objects, and second, Attribute Learning Network (ALN) was designed to verify generated proposals and predict the vehicle's color, type, and pose simultaneously. Both networks were optimized collectively to collect maximum latent information learned from ALN. The proposed model was evaluated on three public datasets and provided improvements over previous methods.

Yan et al. (2016) [56] proposed a model based on hypothesis generation and hypothesis verification. During hypothesis generation, possible vehicles were detected considering shadows under vehicles. In the other phase, the generated hypothesis was labeled vehicles and non-vehicles. In this study, two types of HOG descriptors were utilized to extract features of vehicles, and then these were integrated for the final set. The AdaBoost classifier was trained on the integrated features set. The author claimed that the developed system had acceptable accuracy and could be used in real-time.

Ren et al. (2017) [57] introduced a Region Proposal Network (RPN) in which convolutional features of the whole image were shared with the detection model and enabled the cheapest region proposals. Further, using the 'attention' method, the trained RPN network was merged with the Fast R-CNN to create a single model by sharing the convolutional features. Experimental testing on PASCAL VOC and MS COCO datasets provided state-of-the-art accuracy.

Ullah et al. (2017) [58] presented a method to extract the features from moving vehicles, such as model, make and type. The detection model was designed using deep neural networks (DNN). The frame difference technique was utilized to predict moving vehicles, and then a symmetrical filter was used to extract the front part of the vehicle. The extracted frontal part was given as an input to DNN for identification. A custom dataset was collected to carry out the study, and the proposed model provided 96.31% top-1accuracy.

Tayara et al. (2017) [59] introduced a vehicle detection and traffic estimation model by utilizing convolutional regression neural network. In this study, a regression model was used to infer the traffic count of an input image. The proposed model was tested on two public datasets: Munich and OIRDS. The author claimed that the proposed model was more effective and yielded higher recall, precision, and F1 score. However, the inference time of the proposed model was high.

Li et al. (2017) [60] designed a Scale-Aware Fast R-CNN (SAF R-CNN) in which large-size and small-size sub-networks were incorporated into a fused design to process pedestrians available in the image at different sizes. The proposed model was able to train the fused architecture by sharing early layers of convolutional filters for extracting mutual features and integrated the results of both sub-networks with scale aware weighing scheme. Experimental results depicted that the proposed model was efficient in detecting small-size pedestrians and achieved better accuracy on various benchmark datasets.

Krizhevsky et al. (2017) [61] presented a neural network with 60 million parameters and 6,50,000 neurons in which five convolution layers were used, and max pooling layers were added to avoid overfitting. In the end, three dense layers with 1000 neurons at output layers were implemented. Softmax activation function was utilized at the output layer. Dropout layers were used to minimize the chances of overfitting in dense layers. The proposed model attained top-1 and top-5 error rates of 37.5% and 17.0%, respectively.

Redmon et al. (2017) [62] introduced YOLOv2 and YOLO9000 for real-time detection. The author claimed that YOLOv2 was the fastest compared to other detection models and provided a trade-off between accuracy and speed. In YOLO9000, WordTree was utilized to merge data from different sources. The model was trained with their joint optimization method on COCO and ImageNet datasets.

Lin et al. (2017) [63] investigated the primary reason for the class imbalance problem and proposed a method to address this issue. This study used the cross-entropy loss to minimize the loss values given to correct classified samples. The author proposed focal loss, which focused on training a sparse set of complex samples and avoided many easy negatives. Experimental results had shown that the proposed model trained on focal loss was comparable with one-stage detectors in terms of speed and having better accuracy than two-stage detectors.

Bodla et al. (2017) [64] proposed soft- Non-Maximum Suppression (NMS) in which detection scores of all detected objects were decayed as a continuous function of overlapping maximum detection value. Accordingly, no complete object was removed. The suggested method showed cutting-edge accuracy using the PASCAL VOC and MS COCO datasets for training and testing.

Huang et al. (2017) [65] investigated the three main object detectors: Faster R-CNN, R-FCN, and SSD, and showed that if fewer proposals were considered in Faster R-CNN, the speed of the detector could be increased without much loss in accuracy as well as author claimed that feature extractor had less impact on SSD performance. Furthermore, the author found a trade-off between the speed and accuracy of any object detector.

Yu et al. (2017) [66] utilized Faster R-CNN and a fine-grained detection and classification model to recognize the vehicles' make and model. Faster R-CNN was used to detect vehicles from images. In the next step, an image comprising a single vehicle was passed to CNN to extract features, and a joint Bayesian belief network was utilized for classification. The author created a custom dataset to perform experimental results and showed acceptable accuracy.

Wang et al. (2017) [67] implemented Faster R-CNN for vehicle detection and classification. The author utilized the PASCAL VOC dataset for experimental results and claimed that the model provided 90.51% and 90.65% accuracy if considered only trucks and cars.

Zhuo et al. (2017) [68] proposed a two-step vehicle detector in which the first model was pre-trained on GoogleNet on the ILSVRC-2012 dataset, and then the trained model was fine-tuned on the vehicle dataset. All images of the vehicle dataset were categorized into six classes. Experimental results depicted that the proposed model had an accuracy of 98.26%, which was 3.42% greater than machine learning models.

Sindagi et al. (2018) [69] surveyed crowd counting and density estimation based on CNN and traditional methods. The author categorized CNN-based techniques into the training process and network topology. Experimental results were performed on various datasets using traditional and CNN methods and proved that CNN models had better handling capacity to deal with crowd detection and density estimation at different scales and scenes.

Suhao et al. (2018) [70] utilized the Faster R-CNN method with an improved RPN network to detect and classify vehicles effectively. The proposed model was tested and trained using data from MIT and Caltech, demonstrating its effectiveness.

Arinaldi et al. (2018) [71] implemented Faster R-CNN and a Mixture of Gaussian (MoG) + SVM for counting the vehicles, identifying the type of vehicles, and estimating the speed of vehicles. From the experimental results, the author claimed that Faster R-CNN outperformed MoG in detecting vehicles in different environmental conditions.

Tsai et al. (2018) [72] improved and optimized the existing Faster R-CNN method for transportation applications. The dataset had a total of seven classes. The proposed model achieved an accuracy of 90% when three classes, small vehicles, big vehicles, and trucks, were considered.

Han et al. (2018) [73] applied a CNN-based approach to classifying vehicles. The author also proposed an unsupervised pre-training technique to initialize the parameters of CNN to enhance the classification capabilities of the model. The accuracy of the experiment was 93.5 percent.

Sheeny et al. (2018) [74] utilized a polarised LWIR (POL-LWIR) camera to collect data from moving vehicles. The author evaluated the two popular object detectors, SSD and Faster R-CNN, using different architectures. Models were evaluated on collected data as well as on the KITTI dataset. Experimental results showed the mAP of 80.94% on Faster R-CNN with 6.4 FPS, while the mAP of SSD was 64.51% with 53.4 FPS.

Cai et al. (2018) [75] developed a multi-stage model for object detection, which dealt with training overfitting issues and a quality mismatch in predictions. The proposed model comprised of series of detectors trained with varying IoU thresholds. The Cascade R-CNN model was implemented on the COCO dataset. Experiments have shown that the proposed model achieved good gains.

Chen et al. (2018) [76] addressed the problems of object detection with less number of training examples and proposed a low-shot transfer detector (LSTD). In the given model, a deep flexible configuration of LSTD was designed to minimize the transfer difficulties of low-shot detections. Then, the proposed model was trained with new regularization parameters, Transfer Knowledge (TK), and Background Depression (BD) to fine-tune with fewer target samples. The author claimed that the proposed model outperformed PASCAL VOC and ImageNet datasets.

Nam et al. (2018) [77] developed a vehicle detection and classification model in which vehicles were classified by type. The author utilized visible and thermal images and extracted headlight and grill areas. Texture characteristics were extracted from given images and used to classify moving vehicles. Experimental results were performed with six-category and three-category datasets. When considering six categories, the proposed model provided an accuracy of 92.7% on visible images and 65.8% on thermal images. Similarly, accuracy was

95.9% and 70.5% on visible and thermal images, respectively, when used with three categories.

Liu et al. (2018) [78] addressed the issue of imbalanced class datasets and proposed a semi-supervised pipeline consisting of DNN with data augmentation methods based on generative adversarial nets (GANs). The suggested model was comprised of three phases. During the first phase, many GANs were trained on the actual dataset to produce adversarial examples from the rare categories. In the second phase, an ensemble was trained on the unbalanced dataset with different CNN configurations. A sample selection technique was applied to figure out the low-quality adversarial examples.

At last, the proposed ensemble was trained on the augmented images. The proposed model was evaluated on MIOvision Traffic Camera Dataset (MIO-TCD). Experimental results depicted that the proposed model increased some rare classes' performance and maintained overall high accuracy compared with base estimators.

Murugan et al. (2018) [79] designed a system that consisted of background subtraction, vehicle detection, and tracking using structural matching, extracting features, and classifying images. Initially, images were converted from true color to grey color. Then, the Gaussian mixture model (GMM) and morphological operations were utilized to extract foreground and moving objects respectively. Kalman filter was used to track the vehicles in multiple frames. The proposed ANFIS model was used to classify the detected vehicles. Experimental results have shown the superiority of the proposed model as compared to traditional neural networks.

Huang et al. (2019) [80] discussed the issue of scoring instance segmentation and proposed a CNN-based Mask Scoring R-CNN. The proposed model attempted to align the mask score with Mask IoU, which was earlier disregarded in maximum instance-based segmentation methods. On the MS COCO dataset, the proposed model had shown consistent performance and outperformed existing Mask R-CNN.

Law et al. (2019) [81] introduced an object detection architecture in which only one CNN was used to detect BB as a pair of key points, top-left corner and bottom-right corner. This study eliminated the concept of anchor boxes utilized in earlier single-stage models. Furthermore, a new corner pooling layer was proposed to improve corner detection efficiency. The author showed 42.2% average precision on the MS COCO dataset and performed better than existing one-stage detectors.

Duan et al. (2019) [82] addressed the issues of CornerNet to judge the internal cropped area with the least cost. The proposed model detected objects with triplets consisting of one central key point and two corners. The significant contribution of the proposed work was to provide the ability of a two-stage detector into one stage detector other than an effective discriminator. The proposed model was evaluated on the MS COCO dataset and provided an average precision of 47%.

Ghiasi et al. (2019) [83] introduced a model to optimize the working of designing a Feature Pyramid Network (FPN) with Neural Architecture Search (NAS) and known as NAS-FPN. The suggested model was more flexible and provided better detection results on the MS COCO datasets. Additionally, the supplied model significantly improved the trade-off between accuracy and speed.

Tian et al. (2019) [84] introduced a one-stage detector that was free from generating anchors and proposals, known as the Fully Convolutional One-Stage (FCOS) object detection model. The proposed model solved the detection task in a per-pixel prediction manner and eliminated computation and hyperparameter tuning corresponding to anchor boxes. FCOS model achieved the highest accuracy among already existing one-stage detectors. The author also claimed that the proposed model could be utilized as RPN in the two-stage detector to improve their performance.

Azimi et al. (2019) [86] proposed a vehicle detection model known as ShuffleDet in which speed performance was improved by utilizing shuffling channels and grouping convolutions. The inception module was used to recognize the size and shape of the vehicle. Experimental results were performed on CARPK and PUCPR datasets, and the proposed model processed the images at 14 FPS.

Kim et al. (2019) [87] introduced a traffic surveillance system in which vehicles were detected, tracked, and classified with image processing methods and CNN models. Custom data was collected by installing a video camera on the road. The proposed model utilized transfer learning for training the CNN model. The developed model was able to track multiple vehicles, classify them, and calculate their speed.

Biswas et al. (2019) [87] implemented two models (SSD and MobileNet-SSD) to determine the traffic density. The author analyzed the pros and limitations of both models based on the collected dataset. Experimental results depicted that SSD achieved 92.97% average accuracy while the accuracy of MobileNet-SSD was 79.30%.

Sun et al. (2020) [88] introduced a lightweight CNN in which features were optimized, and a joint learning scheme was utilized to classify vehicles based on type. In this study, depth-wise separable convolution was used to minimize the parameters of the network. Softmax loss and contrastive center loss were merged to increase the model's capacity for classification. Experimental results were performed on the Car-159 dataset, and the author claimed that model had less complexity while maintaining accuracy.

Kumar et al. (2020) [89] combined the feature values with the bat optimization method to find the optimum feature set. SVM was integrated with the local binary pattern to design bounding boxes with a confidence score. Enhanced Convolutional Neural Network (ECNN) was utilized to remove interference area vehicles and moving objects. Experimental results showed 96.63% accuracy.

Wang et al. (2020) [90] presented a method to classify small vehicles in the wild by using GANs. Discriminator consisted of two classification modules that could classify whether there was a car, van, or non-vehicle. Furthermore, the author proposed a novel mixed objective function to improve the comprehensive and perceptible information. The proposed model achieved the highest precision of 92.97%.

Shvai et al. (2020) [91] proposed an ensemble model in class probabilities obtained from CNN that were fused with continuous class probability values obtained from Gradient boosting-based method. The given model was evaluated on a custom real-world dataset with an accuracy of 99.03%.

Awang et al. (2020) [92] proposed an enhanced feature extraction approach based on Sparse-Filtered CNN with the Layer-Skipping method (SF-CNNLS). Three channels of SF-CNNLS were used to extract main and unique features. The proposed model was tested on the BIT benchmark and the custom SPINT datasets. The model showed the highest accuracy of 93%.

Grents et al. (2020) [93] merged Simple Online and Real-time Tracking (SORT) method with Faster R-CNN to detect and classify vehicles by type. The proposed model also estimated the speed of vehicles with an accuracy of 78%.

Zhu et al. (2020) [94] proposed MME-YOLO, consisting of two sub-models: the improved inference head and the LiDAR image composite model. The first sub-model could identify duplicate visual clues by feature selection blocks and anchor-based or anchor-free ensemble models. In other sub-models, the actual point data was analyzed deeply and combined with the visual backbone architecture at different levels, enabling vehicle detection under unusual lighting conditions. Experimental results indicated that the proposed model achieved accurate and reliable vehicle detection results.

Jagannathan et al. (2021) [95] worked on two public datasets that are MIOvision traffic dataset and the BIT vehicle dataset. Initially, images from datasets were pre-processed to improve their quality using adaptive histogram equalization and GMM. After that, Steerable Pyramid Transform and Weber Local Descriptors were implemented for feature extraction from the detected vehicles. At last, extracted feature vectors were passed as input to the proposed ensemble for vehicle classification.

Hu et al. (2021) [96] proposed an improved YOLOv4 model to detect vehicles from video streams. This paper suggested an algorithm to speed up detection and conducted experimental tests. Simulation results showed that the proposed model had good accuracy and could be used for safe vehicle driving decision-making.

Yang et al. (2021) [97] introduced feature fused SSD model and Tracking-guided Detections Optimizing (TDO) method for accurate and fast vehicle detection from videos. In the feature fused SSD, TDL replaced NMS through which inter-frame vehicles were linked by a fast-tracking method. Hence, propagated inferences could compensate for missed detections, and final results confidence scores were optimized. Experimental results on highway datasets showed an mAP of 8.2% greater than the base estimator.

Jamiya et al. (2021) [98] proposed a lightweight model, LittleYOLO-SPP, based on a YOLOv3-tiny network. In the proposed model, spatial pyramid pooling layers were introduced, which comprised pooling layers at different scales for feature concatenation to improve the learning abilities of a network. Further, network performance was improved by considering MSE and generalized IoU (GIoU). Experimental results on the PASCAL VOC dataset achieved an mAP of 77.44%, while 52.95% on the MS COCO dataset.

Wang et al. (2022) [99] proposed a method for vehicle detection from the UAV video. Hue, saturation, and value (HSV) spatial brightness operations were performed on video frames to enhance the adaptability of a model under various lighting conditions. After that, vehicle detection was done using the SSD model. The traditional SSD model was optimized by considering the focal loss function.

A summary of the different object detection models developed till now based on their architecture and performance is given in table 2.1. Table 2.2 summarizes various research done on vehicle detection and classification.

Table 2.1: Major milestones in object detection research based on the deep convolutional neural network

| Researcher | Model | Year | Type | Observations |
|---|---|---|---|---|
| Krizhevsky [61] | AlexNet | 2012 | Backbone Architecture | It was a large and complex architecture for computer vision tasks consisting of 650,000 neurons with 60 million parameters. In this, the ReLU activation function was used rather than Tanh, increasing the speed six times with the same accuracy. The dropout was used to avoid over-fitting. |
| Permanent [100] | OverFeat | 2013 | One Stage Detector | The original classifier was extended into the detector by considering the last fully connected layer as 1 X 1 convolutional layer to allow arbitrary input. It had shown significant speed strength as compared to two-stage detectors. |
| Simonyan and Zisserman [38] | VGGNet | 2014 | Backbone Architecture | In this, many 3X3 convolutional layers were used in increasing depth. The features were scaled down by using the max-pooling layer. Finally, the softmax classifier was applied to two fully connected layers consisting of 4096 nodes. The major limitation was its slow training speed and a large number of weights. |
| Girshick [101] | R-CNN | 2014 | Two-Stage Detector | Training and testing time was very high. Hard to get a globally optimal solution. |
| He [41] | SPP-net | 2014 | Two-Stage Detector | Feature maps were calculated from the whole image, and fixed-length feature vectors were extracted. Detection performance was good, even when objects were at different scales and aspect |

| | | | | |
|---|---|---|---|---|
| | | | | ratios. |
| Girshick [43] | Fast R-CNN | 2015 | Two-Stage Detector | Features were extracted using the RoI pooling layer. The optimal solution, high accuracy, and better training and testing speed were significant advantages. |
| Szegedy [100] | GoogleNet | 2015 | Backbone Architecture | In this, the inception module was used. |
| Ren [57] | Faster R-CNN | 2016 | Two-Stage Detector | Proposals were generated using Region Proposal Network (RPN). Hard to observe small targets. |
| He [102] | ResNet | 2016 | Backbone Architecture | It reduced training difficulties. So, it got a more optimal choice. |
| Li [51] | FPN | 2016 | Backbone Architecture | It was a feature detector that integrated with object detectors. |
| Redmon [47] | YOLO | 2016 | One Stage Detector | Object detection was considered a regression problem. Difficult to detect small and crowded objects. |
| Newell [103] | Hourglass | 2016 | Backbone Architecture | It captured both local and global information. It first down-sampled the input image and then up-sampled the feature map. |
| Liu [48] | SSD | 2016 | One Stage Detector | Hard negative mining was used to avoid negative proposals. Data augmentation also helped in improving detection accuracy. Capable of performing real-time inference. |
| Dai [51] | R-FCN | 2016 | Two-Stage Detector | Relative position information was provided by a position-sensitive score map of different classes, and features were extracted using ROI pooling. |
| Lin [63] | ResNet | 2017 | Backbone Architecture | It reduced computation and memory costs. Backbone accuracy is also improved. |

| | | | | |
|---|---|---|---|---|
| Huang [104] | DenseNet | 2017 | Backbone Architecture | Spatially robust features were retained, and the flow of information was improved by mixing the input with the residual output. |
| Chen [105] | DPN | 2017 | Backbone Architecture | It had the benefits of the ResNet and DenseNet models. |
| Lin [106] | RetinaNet | 2017 | One Stage Detector | The focal loss was used to subdue the negative samples gradient rather than discarding them. A feature pyramid network was used to detect different size objects. |
| Howard [107] | MobileNet | 2017 | Backbone Architecture | In this, the number of channels in each feature map was the same as the coordinates. Computational cost and number of parameters were reduced significantly. It was specially designed for mobile platforms. |
| Cai [75]] | Cascade RCNN | 2018 | Two-Stage Detector | It worked similarly to RefineDet, and proposals were refined in a cascaded manner. |
| Law and Deng [81] | CornerNet | 2018 | One Stage Detector | Objects were detected as a pair of corners |
| Google Brain Team [108] | Efficient Det | 2020 | One Stage Detector | In this, ImageNet pre-trained EfficientNet was used as the backbone model. Its computations speed is high than YOLO and AmoeabeaNet. |

Table 2.2: A summarized Literature review of Vehicle Classification and Detection

| **Author** | **Images / Videos** | **Camera Type** | **Object Type** | **Model/Technique** | **Metric Used** | **Dataset** | **Value (%)** |
|---|---|---|---|---|---|---|---|
| Ozkurt et al. [28] | Videos | Visible RGB | Vehicles | Neural network + Image Processing | Accuracy | Istanbul traffic management company | 94 |
| Kembhavi et al. [30] | Images | Visible RGB | Vehicles | Partial Least Square Methods + HoG | Precision vs., Recall | Google Earth San Francisco Data set and Overhead Imagery Research Data set(OIRDS) | Achieved the highest curve. |
| Arróspide et al. [31] | Images | Visible RGB | Vehicles | HoG | Accuracy | GTI vehicle image database | 96.94 |
| Arróspide et al. [32] | Images | Visible RGB | Vehicles | Log-Gabor Filter | Accuracy | GTI vehicle image database | 95.84 |
| Zhang et al. [33] | Images | Visible RGB | Vehicles | Ensemble of machine learning models with a majority voting scheme. | Accuracy | Custom data collected from the local police department | 98.65 |
| Arróspide et al. | Images | Visible RGB | Vehicles | Ensemble of SVM, | Accuracy | GTI vehicle image | 98 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [34] | | | | HoG and Gabor filter. | | database | |
| Qi et al. [35] | Images | Thermal | Pedestrian | Sparse representation-based classification and boundary box shrinking. | Log-average miss rate | LSI Pedestrian Dataset | 26 |
| Dollar et al. [36] | Images | Visible RGB | Pedestrian and general object detection | Fast feature pyramids based on visual recognition system. | Log-average miss rate | INRIA | 17 |
| | | | | | | Caltech | 43 |
| | | | | | | TUD | 50 |
| | | | | | | ETH | 50 |
| Karpathy et al. [37] | Videos | Visible RGB | General object detection | CNN | Accuracy | Sports-1M | 80.2 |
| | | | | | | UCF-101 | 65.4 |
| Simonyan et al. [38] | Images | Visible RGB | General object detection | CNN | Top-1 error | ILSVRC-2012 | 23.7 |
| | | | | | Top-5 error | ILSVRC-2012 | 6.8 |
| John et al. [39] | Images | Thermal | Pedestrian | Fuzzy C-Means + CNN | Log average miss rate | LSI Pedestrian Dataset | 34 |
| Zangenehpour et al. [40] | Videos | Visible RGB | Pedestrian and bicycle | HoG + SVM | Accuracy | Custom dataset | 88 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| He et al. [41] | Images | Visible RGB | General object detection | SPP-net | mAP | Pascal VOC2007 | 82.44 |
| | | | | | | Caltech | 93.42 |
| Angelova et al. [42] | Videos | Visible RGB | Pedestrian | Deep Neural Network | Average miss rate | Caltech Pedestrian | 32.52 |
| Girshick et al. [43] | Images | Visible RGB | General object detection | Fast R-CNN | mAP | PASCAL VOC | 66.9 |
| Dong et al. [44] | Images | Visible RGB | Vehicles | CNN+ Sparse Laplacian filter | Accuracy | BIT-vehicle | 96.1(Day) |
| | | | | | | | 89.4(Night) |
| Fan et al. [45] | Images | Visible RGB | Vehicles | Faster R-CNN | Accuracy | KITTI vehicle dataset | 95.14 |
| Fu et al. [46] | Videos | Visible RGB + Thermal | Vehicles + Pedestrian | HoG + SVM | Accuracy | Custom dataset (Thermal) | 96.1 |
| | | | | | | Custom dataset (RGB) | 96.8 |
| Redmon et al. [47] | Images | Visible RGB | General object detection | YOLO | mAP | PASCAL VOC | 63.4 |
| | | | | | | Picasso | 53.3 |
| | | | | | | People Art | 45 |
| Liu et al. [48] | Images | Visible RGB | General object detection | SSD | mAP | PASCAL VOC | 76.9 |
| | | | | | | MS COCO | 80 |
| | | | | | | ILSVRC | 43.2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Kong et al. [49] | Images | Visible RGB | General object detections | Hypernet | mAP | PASCAL VOC | 76.3 |
| Bell et al. [50] | Images | Visible RGB | General object detections | Inside-Outside Net | | PASCAL VOC | 79.2 |
| Dai et al. [51] | Images | Visible RGB | General object detections | R-FCN | Accuracy | PASCAL VOC | 83.6 |
| Kang et al. [52] | Videos | Visible RGB | General object detections | CNN | mAP | YouTube object dataset | 76.8 |
| | | | | | | ImageNet VID | 41.7 |
| Pinheiro et al. [53] | Images | Visible RGB | General object detections | Segmentation with feed-forward network | mAP | MS COCO | 33.5 |
| Dai et al. [54] | Images | Visible RGB | General object detection | Semantic segmentation with deep learning | mAP | PASCALVOC | 75.9 |
| | | | | | | MS COCO | 44.3 |
| Zhou et al. [55] | Images | Visible RGB | Vehicles | CNN | mAP | PASCAL VOC2007 | 64.44 |
| | | | | | | LISA2010 | 79.41 |
| | | | | | | Urban Traffic Surveillance (UTS) | 62.85 |
| Yan et | Images | Visible | Vehicle | HoG + | Accura | GTI vehicle | 97.24 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| al. [56] | | RGB | s | AdaBoost | cy | image database | |
| Ren et al. [57] | Images | Visible RGB | General object detection | Faster R-CNN | mAP | Pascal VOC | 78.8 |
| | | | | | | MS COCO | 42.7 |
| Ullah et al. [58] | Images | Visible RGB | Vehicles | Deep neural network | Accuracy | Custom dataset | 96.31 |
| Tayara et al. [59] | Images | Visible RGB | Vehicles | Convolutional regression neural network | Accuracy | Munich and Overhead Imagery Research Data Set (OIRDS) | 92 |
| Li et al. [60] | Images | Visible RGB | Pedestrian | Scale aware Fast R-CNN | Log average miss rate | Caltech | 9.32 |
| | | | | | | INRIA | 8.04 |
| | | | | | | ETH | 34.64 |
| | | | | | Average Precision | KITTI | 77.93 |
| Krizhevsky et al. [61] | Images | Visible RGB | General object detection | CNN | Top-1 error rate, Top-5 error rate | ILSVRC-2012 | 37.5<br>17.0 |
| Lin et al. [63] | Images | Visible RGB | General object detection | Feature pyramid network | Average Precision | MS COCO | 59.1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Yu et al. [66] | Images | Visible RGB | Vehicles | Faster R-CNN + Joint Bayesian belief network | Accuracy | Custom dataset | 89 |
| wang et al. [67] | Images | Visible RGB | Vehicles | Faster R-CNN | Accuracy | PASCAL VOC | 90.65 |
| Zhuo et al. [68] | Images | Visible RGB | Vehicles | CNN pre-trained on GoogleNet + trained on Vehicle dataset | Accuracy | ILSVRC-2012, vehicle dataset | 98.26 |
| Suhao et al. [70] | Images | Visible RGB | Vehicles | Faster R-CNN with improved RPN | Accuracy | MIT + Caltech car dataset | 84 |
| Arinaldi et al.[71] | Videos | Visible RGB | Vehicles | Faster R-CNN + MoG + SVM | Accuracy | Indonesian Toll Road dataset | 67.2 |
| | | | | | | MIT traffic | 69.4 |
| Han et al. [73] | Images | Visible RGB | Vehicles | CNN | Accuracy | Custom dataset | 93.5 |
| Nam et al. [77] | Images | Visible RGB + Thermal | Vehicles | Texture features | Accuracy | Custom dataset | 95.9 (Visible) 70.5 (Thermal) |
| Liu et al. [78] | Images | Visible RGB | Vehicles | GANs + ensemble of CNN | Average Precision | MIO-TCD dataset | 93.55 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Murugan et al. [79] | Videos | Visible RGB | Vehicles | GMM + Kalman filter + ANFIS | Accuracy | Custom dataset | 92.6 |
| Kim et al. [86] | Images | Visible RGB | Vehicles | Image processing + CNN | Accuracy | Custom dataset | 98 |
| Biswas et al. [87] | Images | Visible RGB | Vehicles | SSD, MobileNet-SSD | Accuracy | Custom dataset | 92.97 (SSD) 79.30 (MobileNet-SSD) |
| Sun et al. [88] | Images | Visible RGB | Vehicles | CNN | Precision | Car-159 | 85.34 |
| Kumar et al. [89] | Videos | Visible RGB | Vehicles | Bat optimization + SVM + ECNN | Accuracy | Custom dataset | 96.63 |
| wang et al. [90] | Images | Visible RGB | Vehicles | GANs | Precision | KITTI dataset | 92.97 |
| Shvai et al. [91] | Images | Visible RGB | Vehicles | Ensemble of CNN | Accuracy | Custom dataset | 99.03 |
| Awang et al. [92] | Images | Visible RGB | Vehicles | Sparse-filtered CNN with layer skipping | Accuracy | BIT vehicle dataset, custom dataset | 93 |
| Grents et al. [93] | Videos | Visible RGB | Vehicles | SORT + Faster R-CNN | Accuracy | Custom dataset | 78 |
| Zhu et al. [94] | Images | Visible RGB | Vehicles | MME-YOLO | Precision | Custom dataset | 91.18 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Jagannathan et al. [95] | Images | Visible RGB | Vehicles | Image processing + ensemble learning | Accuracy | MIO-TCD | 99.13 |
| Hu et al. [96] | Videos | Visible RGB | Vehicles | YOLOv4 | Detection speed | Custom dataset | 16FPS |
| Yang et al. [97] | Videos | Visible RGB | Vehicles | Feature fused SSD + TDO | mAP | ImageNet VID | 83.5 |
| Zhu et al. [98] | Images | Visible RGB | Vehicles | LittleYOLO-SPP | mAP | PASCAL VOC | 77.44 |
| | | | | | | MS COCO | 52.95 |
| Wang et al. [99] | Videos | Visible RGB | Vehicles | HSV + SSD | Accuracy | | 96.49 |

For a literature review on vehicle detection and classification, research papers have been taken from 2009 to 2022. Figure 2.3 provides the statistics of the number of research papers taken year-wise.

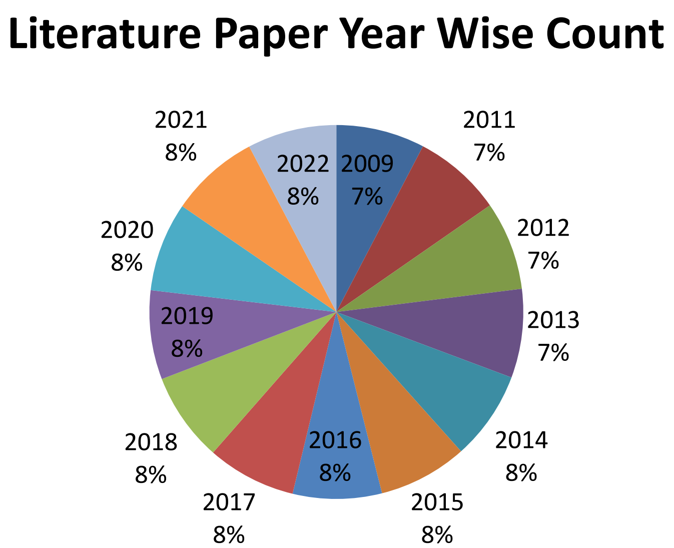



Figure 2.3: Number of papers reviewed year-wise for vehicle detection.

Figure 2.4 shows the ratio of image-dataset vs. video dataset taken in collected research papers. It has been observed that most researchers used image-based datasets over video-based datasets. Similarly, Figure 2.5 shows the camera type, thermal or visible cameras for collecting datasets. Figure 2.5 shows that 96% of the datasets have been collected from visible cameras, and research on thermal images and videos is relatively less. Figure 2.6 shows the various datasets taken for vehicle classification and detection research works and observed that PASCAL VOC and custom datasets are major sources. Various performance measures utilized to evaluate the model performance were shown in Figure 2.7 and analyzed that precision, accuracy, mAP, and precision-recall curve are the most utilized metrics.

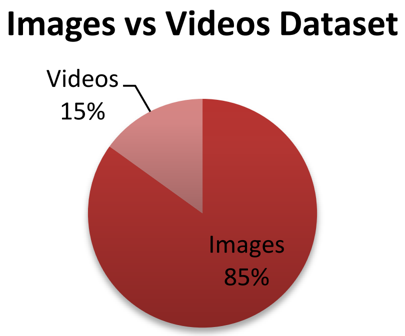



Figure 2.4 Videos vs. Images used in Literature

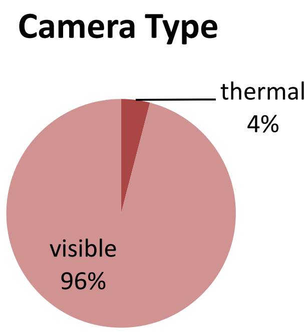



Figure 2.5: Thermal vs. visible images used in previous studies

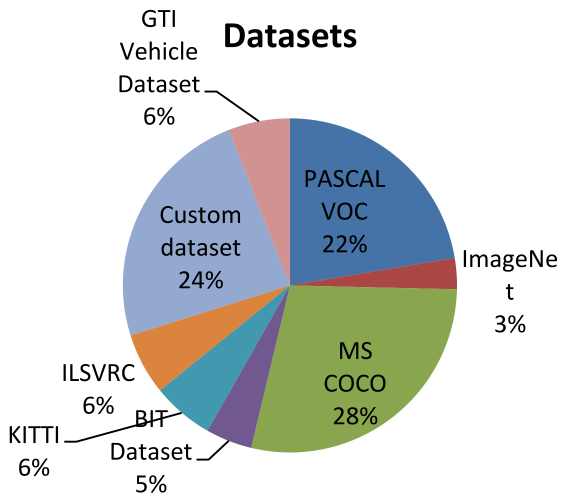



Figure 2.6: Datasets used in previous studies

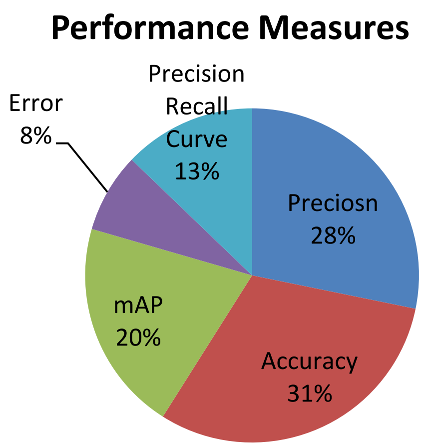



Figure 2.7: Performance measure used in previous studies

## 2.2 Literature Review Based on Emergency Vehicle Detection

Fazenda et al. (2009) [109] presented a method for incoming emergency vehicle detection. Ambisonic technology and an array of 4 ubiquitous speakers were used to provide information to the drivers. Simulation results demonstrated the efficiency of the suggested system.

Liaw et al. (2013) [110] proposed a system for ambulance recognition based on siren sounds in Taiwan. Initially, frames were extracted from the given sound. Every extracted frame was categorized into high-frequency and low-frequency classes. Further, the Longest Common Subsequence (LCS) was utilized to compare the organization of frequencies in the extracted frames. Simulation results showed 85% accuracy on realistic sounds.

Schroder et al. (2013) [111] presented Part-based models (PBMs) for detecting siren sounds of emergency vehicles in noisy conditions. The author proposed two major improvements. Rather than random initialization, spectro-temporal part extraction was initialized. Secondly, I preferred the discriminative training method over standard generative training. In the study, one hand-labeled and two ML-learned PBMs existed and were evaluated with standard Hidden Markov Models (HMMs) having mel-spectograms and MFCCs in noise-free and multi-condition training configurations. Experimental results showed that PBMs provided acceptable models for acoustic-based emergency vehicle detection.

Miyazaki et al. (2013) [112] proposed a method to detect ambulance sound and programmed it on a microcontroller. This study utilized Fast Fourier Transform (FFT) twice, and siren sounds were converted into numerical values. The proposed system provided effective results under the Doppler Effect even when the signal-to-noise ratio was 0 dB.

Sundar et al. (2015) [113] presented an intelligent traffic management system to prioritize emergency vehicles. Every vehicle was equipped with Radio Frequency Identification (RFID) tags. The author used NSK EDK-125-TTL and PIC16F877A tag readers to read RFID tags. The proposed system estimated vehicle density and determined network congestion. Green signal timing at an intersection was adjusted based on traffic. If an ambulance was detected, the traffic controller was informed to turn on the green signal. Information was provided to police with GSM SIM300 if the RFID of the stolen vehicle was identified. The proposed model provided the expected results under different inputs.

Dobre et al. (2015) [114] used yelp and wail signals and designed siren detection with minimal computational cost without digital signal processing. The proposed model was implemented and simulated on the SPICE simulator, and the results were acceptable.

Saad et al. (2016) [115] designed an intelligent traffic controller using passive RFID. The proposed model worked in harsh environments very well. Elliptic curve cryptography was used to maintain the system's security. Also, security was implemented to deal with attacks. The experimental results showed that the designed system minimized the congestion on road and emergency conditions.

Islam et al. (2016) [116] presented a traffic controller based on video processing methods and RFID. The frames extracted from videos were utilized to estimate the traffic density. For law enforcement, RFID sensors were implemented so that any motorist not following traffic rules could be caught easily.

Amir et al. (2017) [117] designed an automatic traffic controller with emergency vehicle control. RFID was used and programmed on a Programmable Logic Controller (PLC) to detect emergency vehicles. The proposed model retained the green signal on the road on which the emergency signal was identified and kept it on until emergency vehicles passed the intersection.

Meghana et al. (2017) [118] presented a traffic management system that utilized RFID to estimate traffic density. Traffic density was not estimated as the number of vehicles present; each type of vehicle was assigned several units. Also, priority was given to emergency vehicles with the help of RFID.

Naik et al. (2018) [119] addressed the issue of the emergency vehicle waiting time at the intersection. Hence, an RFID-based system was proposed to resolve the issue of allowing emergency vehicles to pass when detected on a particular road. The author implemented the proposed model with Arduino and LED display. Also, The author adjusted the road's green signal based on its traffic density.

Bhate et al. (2018) [120] utilized the Internet of Things (IoT) to reduce traffic congestion and prioritize emergency vehicles like ambulances, police vans, and fire brigades. The proposed model was implemented with Raspberry PI, Mode MCU, and RFID. Experimental results depicted that emergency vehicles were handled effectively with the proposed model.

Dutta et al. (2018) [121] utilized WSN, IoT, Data analytics, and cloud computing to design an effective traffic light management system. The proposed model could find the optimal route and suggest users find it. The proposed model also managed accidental events. The given model considered precipitation level, accident, green corridor concept, fuel consumption, and traffic flow rate with the help of machine learning algorithms.

Ebizuka et al. (2019) [122] detected approaching emergency vehicles towards the intersection based on siren sounds. From the input signal, FFT was applied for spectral analysis, and the kind of emergency vehicle was detected. The proposed model gave acceptable accuracy even in the presence of noise.

Sara et al. (2019) [123] developed a system for ambulance detection using computer vision. Images were captured using static cameras. First, features were extracted from input images using HOG, Local Binary Patterns (LBP), and Gabor filters. After that, extracted features were used to classify SVM and kNN. The proposed model was estimated using many performance measures and showed acceptable performance.

Goel et al. (2019) [124] compared various models based on deep learning, such as YOLO, R-CNN, and SSD, to detect emergency vehicles on the road. Several experimental results were conducted, and the author claimed that YOLO performed better.

Roy et al. (2019) [125] proposed an automatic system to recognize emergency vehicles from CCTV footage. Model training was performed on the MS COCO dataset, vehicles were divided into emergency, and regular vehicles, as well as CCTV footage images, were collected. For detection, the YOLO-V3 model was considered. Experimental results showed promising results.

Raman et al. (2020) [126] designed a hybrid model in which image processing and acoustic-based detection were performed to detect ambulances and fire trucks. For object detection, SSD Mobilenet was utilized while an algorithm was proposed to detect the siren sounds of these vehicles. The proposed model was tested on a dataset of 100 images and provided an accuracy of 86%.

Tran et al. (2020) [127] presented an ensemble-based approach to classify vehicle horns, siren sounds, and background noise. The first model was aimed at processing raw waveforms, and the second was working with features extracted by MFCC and log-mel spectrogram. According to the author, the suggested model had detection accuracy for siren sounds of 98.24% and worked well with variable input length.

Shirabur et al. (2020) [128] proposed a system using piezo-based traffic density estimation and utilized RFID to collect information about unique tags given to ambulances and fire trucks. Experimental results demonstrated the efficiency of the suggested system.

Karmakar et al. (2020) [129] presented a system to prioritize emergency vehicles. First, a priority code was assigned to emergency vehicles based on their types. Then, the proposed model computed lane clearance time to find several interferences required to minimize the travel time of emergency vehicles. The proposed model was implemented using SUMO, and results proved that travel time decreased with increased interventions.

Supreeth et al. (2020) [130] proposed a system for classifying ambulance sirens from noisy signals and vehicle horns in the frequency domain. First, the input signal was captured using an onboard mic and pre-processed by the sound device module in Python. The recognized ambulance signal was compared with Wail, Horn, Yelp, and Hi-Lo ambulance sirens. After that, a statistical model was utilized to compute basis calculations, and classification was done.

Fatimah et al. (2020) [131] utilized features extracted from the siren to detect the ambulance. Audio sensors were used to record the sounds of sirens and processed by bandpass filters. This study used MFCC and statistical features extracted using Fourier decomposition in the frequency domain. Various machine learning techniques were trained on a selected feature set, including SVM, KNN, and ensemble models. The proposed model showed an accuracy of 98.49%.

Baghel et al. (2020) [132] utilized an extended YOLO model to detect emergency vehicles. This model used two phases. The first phase was used to create BB across objects, and the final classification was done in the second phase. Image tensors were passed to the phase 2 classifier to classify classified vehicles into subclasses further.

Tran et al. (2021) [133] presented a system for detecting emergency vehicles based on audio and visual. In this, the first YOLO model with cross-stage partial connections was trained to classify vision-based emergency vehicle detection. The proposed model achieved an accuracy of 95.5%. The second model was proposed based on audio in which siren sounds were considered to classify emergency vehicles from others. Both models were also integrated to design an audio-vision-based model. The proposed model provided a misdetection rate of 1.54%.

For detecting emergency vehicles, research papers were collected from 2009 to 2021. Figure 2.8 shows the statistics of research papers taken for the literature survey over a given period and observed that maximum research on emergency vehicle detection was done in 2019 and 2020. Figure 2.9 shows the different techniques employed for emergency vehicle detection. It has been analyzed that maximum research was done considering the siren sounds of the vehicles. RFID is also a competing technology for detection applications.

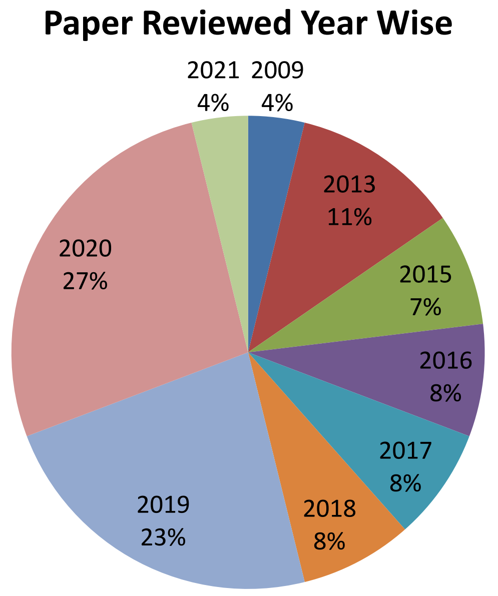



Figure 2.8: Year-wise percentage of Paper Reviewed for Emergency Vehicle Detection.

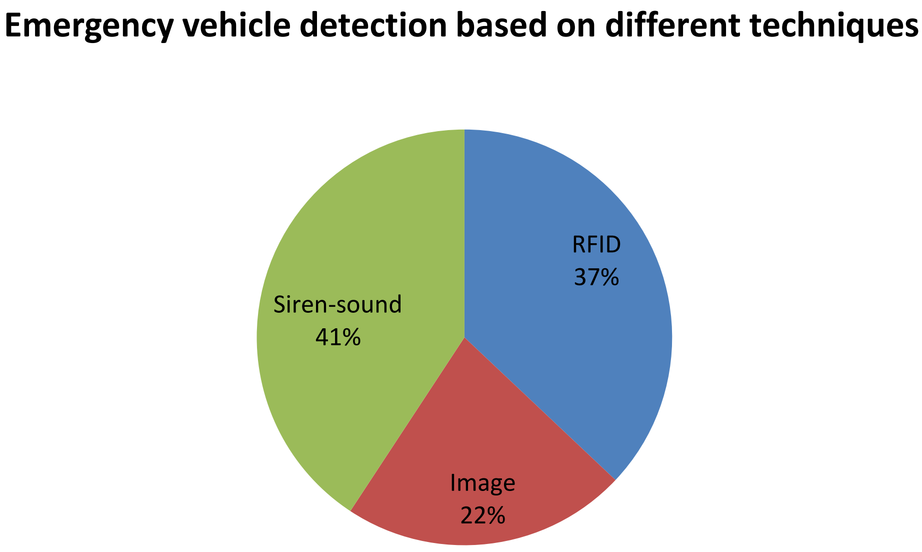



Figure 2.9: percentage of work done on different techniques for emergency vehicle detection

## 2.3 Literature Review Based on Green Signal Optimization

Nakatsuyama et al. (1984) [134] made use of fuzzy logic to design phase (offset) controllers even during rush hours also. Fuzzy control statements were used to control traffic on a one-way arterial road. Experimental findings demonstrated the efficiency of the suggested system.

Chinu et al. (1992) [135] presented a distributed method to control traffic signals in which local traffic conditions at present and adjacent intersections were considered parameters for signal optimization functions. Cycle time, offset, and phase split was used to define the signal timing at an intersection. The author made fuzzy rules to compute the signal timing based on intersection traffic conditions. The author performed several experiments, and simulation results showed the model's effectiveness.

Kok et al. (1995) [136] utilized fuzzy logic to extend traffic controllers' green signal timing to reduce vehicles' overall waiting timing. Extension values were computed based on fuzzy rules. The author performed experiments to compare fixed-time controllers with the proposed system. The author claimed that the proposed method based on fuzzy logic was better in terms of moving time and waiting time.

Trabia et al. (1999) [137] proposed a traffic light controller based on fuzzy logic for an isolated intersection. In this study, loop detectors were placed upstream of the intersection to estimate incoming traffic flow and queue length. Fuzzy rules were used to extend the green timing or terminate the green phase depending on the collected information. Experimental results were performed at a four-way intersection to compare the performance of the proposed system with traffic-actuated controllers.

Mirchandani et al. (2001) [138] designed an adaptive traffic controller known as RHODES. The system used four steps to design the controller. Firstly, the traffic control problem was divided into sub-problems and arranged hierarchically. Then traffic flow was predicted at a different level to activate proactive control. After that, different optimization methods were utilized to solve decomposed problems. Finally, an appropriate data structure was utilized to compute the solutions of sub-problems. Simulation results showed the effectiveness of the proposed system.

Chou et al. (2002) [139] designed a coordinated intersection green signal optimization system. Various parameters used as input were adjacent intersections, vehicle queue length, number of lanes, and street length. The proposed model was compared with previous research and had various characteristics like variable input variables, only nine fuzzy rules, lower inference frequency, and coordinating adjacent intersections. The simulation results with three linguistic variables, low, medium, and high, showed better performance.

Li et al. (2003) [140] utilized traffic flow information and diffusion at multiple junctions to design fuzzy based traffic light controller. A weighing mechanism was introduced to assign the weights to analyze the impact of traffic flow at an adjacent intersection on the green signal at the present intersection. In this study, average delay time was used to evaluate the model efficiency. The various parameters optimized were cycle time, phase, and green split. Experimental results proved the effectiveness of the proposed model.

Chiou et al. (2004) [141] used a hybrid approach of genetic algorithm and fuzzy logic for adaptive traffic light controllers. The proposed model considered queue length and flow rate as input variables, green time extension as an output variable, and vehicle delay determined by fluid estimation as an evaluation parameter. Simulation results proved that the proposed model was efficient and robust and could be applied to adaptive controllers.

Conglin et al. (2004) [142] presented a neuro-fuzzy system to estimate the vehicles waiting in the queue. The hybrid system was having an accuracy of 90%. The author also used genetic algorithms and fuzzy logic to design traffic light controllers. Experimental results showed that the delay time of vehicles and the queue length of waiting vehicles was reduced.

Murtal et al. (2005) [143] proposed a model for isolated intersection based on fuzzy logic known as Fuzzy Logic Multi-phased Signal Control (FLMuSic). The model had two components: one for estimating the phase's green time and the other for phase sequencing based on traffic density. Simulation results were performed on three and four-phased intersections. In both scenarios, the proposed system showed effective results, significantly when traffic density varied at a high rate.

Zeng et al. (2007) [144] designed a control strategy in which per-unit delayed vehicles were considered for signal optimization. Initial values of the controller were estimated by analyzing previous traffic flows. Simulation results depicted the effectiveness of the system.

Nair et al. (2007) [145] presented a fuzzy-based system to control traffic under normal as well as abnormal traffic scenarios. Sensors were deployed at the incoming and outgoing lanes to collect traffic-related data. Collected information was used to optimize the signal timing. The author designed a simulator and carried out various experiments. Experimental results depicted that the proposed model had the same efficiency as other fuzzy-based systems in normal conditions and performed better than other controllers under exceptional scenarios.

Wannige et al. (2008) [146] designed a neuro-fuzzy-based traffic light controller in which inflow rate and traffic density were input parameters. Density was computed by subtracting the inflow rate from the outflow rate and divided by the distance between sensors. The model's output was considered an extension in green signal timing. A four-way intersection was considered to carry out experiments. A few fuzzy rules were added to the model by the author, and the neuro-fuzzy system generated other rules. The collected dataset was split into testing and validation sets. To evaluate the performance, linear membership functions were utilized. Experimental results proved that the neuro-fuzzy inference system was far better than a fuzzy-based system.

Zheng et al. (2008) [147] developed an adaptive signal control model to estimate the optimal parameter values. The parameter values were optimized based on signal timing information to satisfy the demands of real-time vehicles on a cycle-by-cycle basis. The author performed experiments on a network of thirty-eight actuated signals with microscopic simulation. The author claimed that the proposed model was able to improve the performance of traffic light controllers even during peak hours.

Wannige et al. (2009) [148] presented a coordinated control signal for two four-way intersections using a neuro-fuzzy inference system. Membership functions and fuzzy rules were automatically designed by a neuro-fuzzy system based on the provided data set. The main input parameter to the proposed model was the inflow rate. The green signal's duration was determined based on the inflow rate. Experimental results showed that the proposed model minimized vehicle delays under different traffic scenarios. And signal timings were controlled and adjusted at both intersections depending on the traffic scenario at the first junction.

Franceshinis et al. (2009) [149] utilized a wireless sensor network (WSN) for traffic monitoring. The major objective of the research was to design a robust, flexible, low-maintenance, and cost-effective system. Sensor nodes were deployed along the roads to collect vehicle count, direction, and speed. Collected data was sent to the gateway node and further given to the roadside unit to integrate it with other data generation mechanisms from alternative sources.

A wireless sensor network-based solution for typical Indian city traffic was put up by C et al. (2009) [150]. The goal was to comprehend and address the chaotic traffic congestion. The author suggested regulating and optimizing how long a signal is green and how many vehicles pass through the intersection in a certain amount of time. The goal was to optimize the number of cars going through the signal by using WSN to make the traffic signal adaptive to the dynamic traffic flow. Following a simulation-based initial experiment and examination of the proposed algorithm, the authors found that the same infrastructure could accommodate 7% more vehicles.

The geometric fuzzy multi-agent system (GFMAS), which was based on a geometric type-2 fuzzy inference system, was created by Gokulan et al. (2010) using a distributed multiagent-based technique [151]. The various degrees of uncertainty contained in the inputs and rule base of the traffic signal controller can be addressed using GFMAS. A virtual road network simulating a portion of Singapore's financial district was used to test the simulation models of the agents created in PARAMICS. Green Link Determining (GLIDE) and Hierarchical Multi-agent Systems (HMS) were two traffic-control algorithms that were in-depth analyzed and compared (HMS). When tested for realistic traffic-flow circumstances, the suggested GFMAS signal control performed better than both benchmarks. Additional testing demonstrated the proposed GFMAS's more remarkable performance in dealing with planned and unforeseen accidents and impediments. The favorable outcomes illustrated the effectiveness of the suggested multi-agent architecture and the potential for further advancement.

Azimirad et al. (2010) [152] provided an innovative model for a single signalized intersection and a fuzzy logic controller. To guarantee a smooth traffic flow with a minimum waiting time and line length, the controller managed the timings and phase sequence of the traffic lights. Under typical traffic conditions, In general, fuzzy traffic controllers are made to maximize traffic flows and reduce waiting times. Therefore, these were not the best traffic controllers in unusual traffic situations like roadblocks and accidents. The average waiting time for a vehicle in a traffic network under a given time control was formulated using state-space equations. Additionally, the author put forth a fuzzy model and fuzzy traffic controller that could regulate how traffic moves in both familiar and unexpected scenarios. Results indicated that the suggested traffic controller performed better under normal and abnormal traffic situations than traditional fuzzy traffic controllers using a novel fuzzy model.

Wu et al. (2010) [153] created a fuzzy control system in which a sensor measured the number of vehicles in all lanes. The phase with the highest number of vehicles was classified as having the highest priority. The highest priority was transferred as the phase changed from the previous one to the next. The best green light delay duration was then calculated using fuzzy rules reasoning based on the length of the current waiting formation and the overall formation length. The simulation results showed that the fuzzy control method significantly outperformed the conventional timed control method regarding vehicle delay time.

Yousef et al. (2010) [154] proposed a traffic control system that adapts based on new traffic infrastructure and cutting-edge techniques for regulating traffic flow sequences using wireless sensor networks (WSN). The controller implemented the traffic signals time management algorithm and the traffic system communication algorithm (TSCA) (TSTMA). As shown by dynamic changes in the flow sequence of traffic signals and traffic variance, both techniques were capable of giving the system an adaptive and effective traffic estimation. The simulation results showed that the proposed plan was effective in relieving traffic congestion on the isolated (single) intersection based on the typical length of the line and the typical duration of the wait, as well as effective global traffic flow control on many intersections.

Zhou et al. (2010) suggested an adaptive traffic light control technique that modified the order and number of traffic lights in line with the amount of real-time traffic detected [155]. The proposed method determined the ideal green light sequence and length by considering several traffic variables, such as traffic volume, waiting time, vehicle density, etc. Simulation findings demonstrated that the suggested algorithm achieved a substantially better throughput and a much shorter average waiting time for the vehicle compared to a fixed-time control method and an actuation control technique. The results of the proposed technique's application on the author's transportation testbed, iSensNet, showed that it was successful and workable.

A novel application to predict the position and speed of a vehicle utilizing a wireless sensor network was presented by Saqib et al. (2010) [156]. Two Anchor nodes were utilized along the roadside as readers, and the distance between them was calculated. When a moving vehicle with a tag arrives within the expected operational range of two anchor nodes, information is exchanged using the Symmetric double-sided two-way ranging algorithm, which provides us with location information. Velocity might be easily calculated using position data at many time intervals. Kalman filtering was applied to estimate position and velocity from noisy observations.

A method for image-based traffic signal control was proposed by Choudekar et al. (2011) [157]. The technology used photographs mounted next to the traffic light to detect automobiles. Using image matching, the recorded images were successively matched. Edge detection was performed using the Prewitt edge detection operator. The predicted proportion of matched vehicle density was used to measure the duration of traffic lights.

Soh et al. (2011) developed an ANFIS traffic signals controller for multilane intersections to reduce traffic jams at traffic intersections [158]. This study presented a novel idea for producing sample data for ANFIS training. Fuzzy rules were used to generate the sample data, which was then analyzed using a tree diagram. The performance of this controller was evaluated against that of conventional controllers and fuzzy controllers using a multilane traffic intersection model created using the M/M/1 queuing theory. The ANFIS traffic signal controller, out of the three controllers, had the least average waiting times, queue lengths, and delay times, according to the simulation data.

According to the degree of traffic, Zaied et al. (2011) [159] created a fuzzy logic system that considers the two two-way crossings and can vary the time intervals of a traffic light. The proposed approach was evaluated on real-time data collected from a signalized intersection in the Hawalli governorate of the State of Kuwait. The findings demonstrated that the proposed fuzzy logic traffic system performed better regarding total waiting time, moving time, and vehicle queue after 27 iterations. The findings suggested that the suggested method might shorten the cycle's duration and give other phases a better opportunity to capitalize on the lost green time.

Balaji et al. (2011) [160] presented a multi-agent system based on a type-2 fuzzy decision module for traffic signal regulation for a complicated metropolitan road network. The distributed agent architecture with a type-2 fuzzy set-based controller was created to optimize the time when a traffic light is green to decrease vehicles' overall delay. The proposed agent architecture for the signal control was put to the test on a portion of the Singapore Central Business District that was simulated using the PARAMICS program. A hybrid neural network-based hierarchical multi-agent system (HMS) controller and a real-time adaptive traffic controller (GLIDE), both now in Singapore, were used to compare the performance of the proposed multi-agent controller. The present mean speed of vehicles on the road network and the overall mean delay that cars experienced when traveling from point A to point B served as the performance measures for evaluation. The road network's traffic conditions were improved significantly under the proposed multi-agent signal control, reducing the total trip time for simulated cars operating in dual and multiple peak traffic scenarios.

An adaptive traffic light control system was put forth by Zhou et al. (2011) [161] based on real-time traffic information such as waiting time, traffic volume, vehicle density, and the number of stops, and modified the sequences of green lights at various junctions. The length of the ideal green signal can be determined by considering local current intersection traffic and the traffic at adjacent intersections. Experimental results showed that the proposed approach gave promising results.

Zade et al. (2012) [162] offered a simulation of a fuzzy traffic control system to modify the length of the green light for adequate traffic flow. This system's expression of traffic-responsive signal control included two crucial elements: observation of the nearby intersection's current traffic status and proper regulation of the traffic signals. The controller is made based on flow rate and traffic density. This FIS module was created using the MATLAB tool's SIMULINK environment, which produced satisfactory traffic signal control results. The Adaptive Traffic Signal Controller could make decisions based on the Fuzzy Inference system to shorten wait times at intersections.

In a multi-intersection ITS, a distributed algorithm that determined the order and duration of green lights was reviewed by Faye et al. (2012) [163]. The author revealed the design of a WSN placed at junctions that made local judgments independently without the assistance of a centralized body. This sensor network collected data using an adaptive algorithm called TAPIOCA (distributed and Adaptive Intersections Control Algorithm), which made decisions about the green light sequences dynamically while taking into account three goals: (i) decreasing users' average waiting times while lowering the likelihood of starvation; (ii) prioritizing movements with the best load discharge potential; and (iii) synchronizing subsequent lights, for example, to create green wavy lines. In comparison to alternative dynamic methods and pre-determined schedules, simulation results using the SUMO simulator demonstrated that TAPIOCA produced a low average waiting time for cars and responded swiftly to increases in traffic load.

According to Bhuvaneswari et al. (2012) [164], a wireless sensor network can be used to make traffic signals adaptive to changing traffic flow. The proposed method was tested against the current fixed time control system using LabView simulation software.

Dynamic time restrictions at traffic signal junctions are a proposed improvement to the traffic control system made by Bharadwaj et al. (2013) [165]. The proposed system made use of sensors to ascertain the conditions of the traffic in order to control it dynamically. Due to traffic congestion, the existing static traffic control system may impede emergency vehicles like ambulances. The planned Efficient Dynamic Traffic Control System (EDTCS) included Road Side Units (RSU), Traffic Control Units (TCU), and Monitor Units (MU) (RSU). The unique RFID code for an emergency vehicle could be read by an RFID reader at RSU and sent to MU. MU relayed detected data to TCU to count regular and emergency vehicles using proximity switches and RFID tags. By comparing the counts gathered from various lanes, TCU got the regular and emergency vehicle counts and adjusted the signal dynamically. The proposed EDTCS reduced travel time and gave emergency vehicles, such as ambulances, a particular required priority.

The average amount of time that vehicles spend waiting at a junction was decreased because of Srivastava et al. (2013) 's [166] analysis of approaches to construct an intelligent system that was able to blend and support some of the existing technologies of traffic control. The proposed algorithms were adaptable to traffic flow at every road intersection point. Real-world traffic conditions were simulated on the Green Light District Simulator (GLD) platform to create the graph of average waiting time versus cycles. The outcomes demonstrated that the suggested approach worked well for managing traffic at an actual road intersection.

Hussian et al. (2013) [167] presented a wireless sensor network-based system that could route traffic based on the amount of traffic near any circle or intersection. This system could be simply adopted in any traffic system with less effort and expense because it didn't require any systems in the vehicles. This system featured a microcontroller-based routing algorithm developed for superior traffic management and wireless sensor network technology to detect automobiles.

Ahmed et al. (2013) [168] proposed a WSN-based roadside communication architecture and system used for the intelligent control and management of vehicle traffic at road intersections. According to the suggested architecture, data was sent to the coordinator module at the intersection from the end nodes by vehicles that interface with roadside units. The author developed a dependable and durable channel-switching method that improved packet delivery dependability while reducing reaction time, energy use, and connectivity latency. The author conducted a sensitivity analysis of the suggested system architecture to find the best system configuration by changing various communication parameters. The outcomes demonstrated the integrity and feasibility of implementing our suggested architecture.

Zhou et al. (2013) [169] created an adaptable architecture for gathering regional traffic data. Future investigation into developing and putting into practice traffic monitoring solutions was based on this framework. In the context of a WSN environment, a two-layer network architecture was built for the collecting of traffic information. A user-customizable data-centric routing method was also suggested for traffic information distribution, in which various routing-related data was taken into account for decision-making to fulfill diverse user requirements. Compared to other conventional routing algorithms on a real-world urban traffic network, simulations demonstrated the suggested routing scheme's strong performance.

Bodenheimer et al. (2014) [170] presented a strategy to reduce pointless stops, $CO_2$ emissions, and fuel usage. The author turned the traffic light controller's state graph into a transition graph that concentrated on signal changes and the likelihood that they would occur. The system was additionally adjusted for computational speed and storage requirements. According to the author, 80 percent of all cases could accurately identify signal alterations that would occur 15 seconds in the future.

Bi et al. (2014) [171] suggested a multi-agent type-2 fuzzy logic control (FLC) system optimized by differential evolution (DE) for the regulation of multiple intersection traffic signals. Due to their three-dimensional membership functions, type-2 fuzzy sets effectively handled model uncertainties, but choosing the correct membership function and rule base parameters was difficult. The type-2 fuzzy system's parameters were chosen using DE because it was simple to understand and use and had a low level of spatial complexity. The membership functions (MF) and the expert rule base parameters were alternately modified to avoid computational complexity. A traffic network with eleven intersections was investigated, and the suggested controller was used to control each intersection. A secondary layer controller was placed at each intersection to choose the appropriate phase order.

Furthermore, a multi-agent system was used to implement communication among the nearby intersections. Simulation experiments were developed to contrast communicative type-2 FLC optimized by DE with type-1 FLC, fixed-time signal control, etc. According to testing results, the suggested approach might successfully increase vehicle throughput rate while decreasing delay, queue length, and parking rate.

Lin et al. (2014) [172] created a two-stage fuzzy control system for the traffic signals for a single intersection. Based on the amount of traffic in each lane and the wait duration, the fuzzy controller calculated the traffic intensity of each phase and then decided whether to continue operating the current signal phase or halt it. MATLAB was used to simulate the intersection of four phases, and the control system's effectiveness was assessed using typical vehicle delays. The simulation results demonstrated that the two-stage fuzzy control system beat the induction control and conventional timing control systems to reduce the typical vehicle delay.

Chao et al. (2014) [173] utilized Radio frequency identification (RFID) to transfer traffic flow data directly to a control system via an RS232 interface. RFID was proposed as a method of traffic flow detection. In parallel, the sensor evaluated and assessed the data using an extension algorithm created to regulate traffic flow. Furthermore, using ZigBee wireless network connection technology, the traffic flow state was also sent to a remote monitoring control system. This study's traffic flow control technology was capable of remote transmission and decreased traffic accidents. Additionally, it could efficiently manage traffic flow while minimizing delays and preserving a constant traffic flow.

Biswas et al. (2015) [174] suggested a model gathering data about traffic and the presence of high-priority vehicles that utilized infrared proximity sensors and a microcontroller positioned in the center of the system. An intelligent traffic system was created to ease traffic and give emergency vehicles priority.

Collota et al. (2015) [175] proposed a real-time traffic monitoring IEEE 802.15.4 Wireless Sensor Network (WSN) with multiple concurrent fuzzy logic controllers, one for each phase, as part of a traffic light dynamic control system. Each fuzzy controller managed the phase and green time of the traffic signals dynamically while taking into account the turning movements of the cars. The proposed system combined the benefits of using four parallel fuzzy controllers, such as better performance, fault tolerance, and support for phase-specific management, with the benefits of the WSN, such as simple deployment and maintenance, flexibility, low cost, non-invasiveness, and scalability. According to simulation findings, the proposed method worked better than previous alternatives in the literature and significantly decreased vehicle waiting times.

Jagadeesh et al. (2015) [176] introduced a low-cost real-time dynamic traffic light control system using sensors to lower travel time while ignoring the fixed delay in signals. Dynamic time management was tested by dynamizing the traffic lights, measuring the results using an infrared sensor, and transmitting the data via a microcontroller.

Odeh et al. (2015) [177] introduced a hybrid approach that combines Fuzzy Logic Controllers (FLC) with Genetic Algorithms (GAs). In this work, GA was used to modify the FLC decision rules that established an intelligent traffic signal system, outperforming a conventional FLC-based control in terms of performance. According to the simulation findings produced by the hybrid algorithm, performance might be improved by up to 34% compared to a conventional logic controller (FLC) and up to 31% compared to a regular traffic signal controller (CTC).

Chakraborty et al. (2015) [178] developed a system that extended the current dynamic traffic signal control algorithm to reduce the average waiting time. The author also included the worst-case scenario for managing emergency vehicles.

Hung et al. (2016) [179] proposed a method in which a CCD camera was mounted to view moving cars as they appeared in the distance from the camera's view span to the crossroads' stop line, which was designated by the letter L and statistically determined. The traffic signal timing was adaptively altered following the estimated traffic flow using the video sequence that the camera captured in its span on the road, where the gaps between the vehicles were used to calculate the density of vehicles appearing in the camera span.

Nellore et al. (2016) [180] thoroughly analyzed the current urban traffic control programs. The main challenges with congestion control, average waiting time reduction, giving emergency vehicles precedence, and intelligent traffic system design requirements were reviewed to understand urban traffic management's goals better. The author concluded that using cloud computing to construct an intelligent traffic cloud is necessary to address real-time related issues.

Dubey et al. (2017) [181] presented an adaptive traffic system with an internet connection to monitor various lanes continuously. The Central Traffic Control Office evaluated and managed the data collected from various lanes from a single location. This collected data provided a number for the amount of traffic congestion in a specific lane according to which traffic lights were designed to operate. The signal lights would be chosen based on shorter wait times and less pollution if the first lane had less traffic than the other lanes. This technology also guides drivers in selecting a less congested route. This technique was also helpful for traffic surveys, VIP clearance, and emergencies. It made traffic clearance more effective. It also reduced pollution and traffic congestion.

Cruz-Piris et al. (2018) [182] suggested using a graph's centrality measurement as a starting point to identify the ideal spots to deploy sensors in a traffic network. After incorporating these sensors into a simulation scenario, three unique types of agents—traffic light management agents, traffic jam detecting agents, and agents regulating traffic lights at intersections—were identified as part of a multi-agent system. These Multi-Agent Systems' fundamental goal was to reduce networked vehicles' distance travel time. The necessary building blocks for modeling the sensors and agents in the simulation environment were constructed to test the suggested method. The Travel and Activity PAtterns Simulation (TAPAS) Cologne traffic scenario and the Simulation of Urban MObility (SUMO) traffic simulator were both employed in the author's research. The collected data showed that the idea permitted the decrease of the sensor network while still obtaining crucial information to have a thorough grasp of the environment. Finally, the author's experiments demonstrated that the proposed work outperformed other solutions already in use, such as traditional traffic light management systems (static or dynamic), reducing the length of vehicle trips and the overhead associated with message exchange in sensor networks.

Hartanti et al. (2019) [183] suggested a fuzzy Mamdani logic to improve traffic signal regulation at crossings. Based on various real-time parameters, including the length of each row's queue, the number of vehicles entering the line, the speed of the vehicles, and the width of each lane, each row's duration or green light period was optimized.

For efficient traffic flow, Pughat et al. (2019) [184] provided a Green Light-time Estimation (GLE), and delays are tracked by the traffic light controller (TLC). This adaptive TLC based on a fuzzy inference system (FIS) was used to monitor the current traffic situation surrounding the intersection in order to facilitate efficient traffic flow. Two TLC techniques were examined and reported on in this work. The first technique predicted the amount of time using traffic density and flow rate. The second method monitored the length of the green light based on data from vehicle communications, queue size, and traffic flow rate.

Balta et al. (2020) [185] proposed a three-stage fuzzy-decision tree for traffic control which considers information like road construction as well as environmental factors such as accidents or the presence of emergency vehicles. The suggested model generated/sent input to SDN agent vehicles and Road Side Units in an intersection for VANET routing protocols to be automatically selected based on traffic conditions. Under different traffic and network situations, the Adapazar City Center Model was used to assess the performance of the proposed system. According to the results of performance tests, the proposed 3-stage fuzzy-decision model outperforms fixed-time signaling by 15% to a maximum of 17%.

Ali et al. (2021) [186] concentrated on constructing an adaptive traffic signal control system with updated Webster's formula. These calculations were used to determine the ideal cycle time based on the traffic conditions that would apply to the following cycle. The fuzzy logic system was used to monitor and manage the change in traffic conditions between two sequential cycles. Using the SUMO traffic simulator, the proposed adaptive control approaches were compared to fuzzy logic-based traffic control, fixed-time Webster traffic control, and modified Webster traffic control. The simulation results demonstrated that the proposed methods beat the fixed time and fuzzy logic-based traffic control systems regarding the average vehicle delay, speed, and journey time.

Chabchoub et al. (2021) [187] created an intelligent traffic light controller that can govern movement in two different ways using a camera and auto sensors using fuzzy logic and image processing with MATLAB. The console input for the fuzzy logic was the number of vehicles on each road, and the timing of the assumed red, yellow, and green signal based on the level of traffic. The fuzzy logic was created with two inputs and six outputs. The simulation's outcome was comparable to the suggested control unit in that it dealt with the lights concurrently based on the number of vehicles on each road branch, which required constant operation of the stoplights.

In the Vietnamese capital of Hanoi City's isolated intersection with mixed traffic, Vuong et al. (2021) [188] used an adaptive neuro-fuzzy inference system (ANFIS) as a traffic signal control method. The proposed ANFIS method was more efficient because it entirely used an artificial neural network and a fuzzy logic system, intelligently adjusting the green time for each phase of the traffic signal lights by the fluctuating traffic volume under mixed traffic conditions to increase vehicle throughput and decrease waiting time. An example signalized intersection in Hanoi City was studied to determine the efficacy of the suggested technique. Integrating the MATLAB programming language and the VISSIM traffic simulation model, a minor traffic simulator was made. Simulation results showed that the suggested approach using ANFIS outperformed both the fixed-time control strategy and the fuzzy logic method in terms of performance and adaptability.

Table 2.3 shows the summary of the literature review on green signal optimization techniques with respect to different parameters like single/coordinated intersections, performance measures, and outcomes of experimental work.

Table 2.3: A summarized review of Green Signal Optimization Techniques

| **Author** | **Technique** | **Single/Coordinated Intersection** | **Metric Used** | **Results** |
|---|---|---|---|---|
| Nakatsuyama et al. [134] | Fuzzy Logic controller and phase controller | Coordinated Intersection | Delay (secs/ vehicles) | Overall delay of vehicles had been reduced. |
| Chinu et al. [135] | Fuzzy Logic | Single Intersection | Waiting time, number of stops | Overall waiting time and the number of stops had been minimized. |
| Kok et al. [136] | Fuzzy Logic | Single Intersection | Waiting time and movement time. | It was noted that the fixed-time controller was somewhat imbalanced, whereas the fuzzy logic controller offered approximately equal movement of autos in each lane. Using the fuzzy logic controller significantly decreased the overall waiting time for the automobiles in each lane. The fuzzy logic controller outperformed the fixed time controller in terms of cost, which reflected fuel costs, efficiency, etc. |
| Chou et al. [139] | Fuzzy Logic | Single and Coordinated Intersection | Queue length and average | The experimental results showed good performance with low, medium, and high traffic loads. |

| | | | | |
|---|---|---|---|---|
| | | | delay time | |
| Li et al. [140] | Fuzzy logic | Coordinated Intersection | Delay time | The proposed system showed better performance and controlled the real-time traffic and signals. |
| Chiou et al. [141] | Genetic algorithm-based fuzzy logic | Single intersection | Total vehicle delay. | The results suggested that the proposed model was effective, robust, and applicable for adaptive traffic signal control. |
| Conglin et al. [142] | ANFIS | Single intersection | Number of vehicles waiting in the queue | 90% accuracy for queue detection. |
| | | | Number of stops and average delay | Experimental results showed that the proposed model's number of stops and average delay were also smaller. |
| Murat et al. [143] | Two fuzzy logic systems | Single intersection | Signal timing and phase ordering are the two output parameters. | The proposed approach reduced vehicle delays by 23%. |
| Zeng et al. [144] | Fuzzy logic | Single intersection | Number of delayed vehicles | Comparing the proposed strategy to the conventional approach, fewer vehicles were delayed for each unit of time. |
| Nair et al. [145] | Fuzzy logic | Single intersection | Average delay | In unusual traffic conditions, the proposed approach successfully cut down on average delays. |
| Wannige et al. [146] | ANFIS | Single intersection | Average delay | The proposed system reduced the overall delay. |

| | | | | |
|---|---|---|---|---|
| Zheng et al. [147] | Adaptive signal control | Coordinated intersection | Average travel time, vehicle speed, mileage, and vehicle hours traveled. | The proposed model performed best under medium-intensity traffic conditions. |

## Chapter Summary

In this chapter, the literature review based on the research carried out by researchers on vehicle detection and classification, emergency vehicle detection, and green signal optimization based on fuzzy logic and hybrid systems of fuzzy systems has been documented. The selected papers offer an overview of research-based models and techniques for vehicle detection, emergency vehicle identification, and green signal optimization; datasets used, performance measures considered, and types of cameras taken for collecting datasets.

After an exhaustive survey of various sub-topics, the given literature survey will be a significant step in finding research gaps and limitations of existing works and identifying the future scope for carrying out the research.

# Chapter 3

# Vehicle Detection and Traffic Density Estimation

Humans are perfect at identifying multiple objects and detecting obstacles accurately in no or very little time existing in an image. But to perform the same tasks using computers, high processing time, hardware infrastructure, and complex algorithms are required. Nowadays, with the progression of technology, i.e., faster GPUs and convolutional neural networks, and by collecting a large amount of data, machines can train to identify and categorize various objects in an image with high precision.

Object recognition refers to a collection of related computer vision tasks recognizing objects in digital images. *Image classification* [41] is a supervised task that predicts the appropriate class of object available in an image. *Object localization [189]* refers to the determination of the position of one or more objects in an image and presents detected objects by drawing a bounding box around them. *Object detection [189]* is the combination of the above two tasks. An object detection algorithm distinguishes objects of confident, authentic classes from image backgrounds with the defined localization and predicts class labels of all objects. The task of the image classification model is to find the probability of an object belonging to a particular class.

In contrast, the object localization model aims to identify the location, that is, the coordinates of an object. Figure 3.1 shows the figure of object detection. Hence, a reliable and efficient detection algorithm should determine spatial information and a solid understanding of semantic cues about the picture.


Figure 3.1: Object Detection

Systems that use sensing, analysis, control, and communications technologies to improve mobility, efficiency, and security are known as intelligent transportation systems (ITS). Land transportation used ITS to improve traffic flow and reduce waiting time. Wide-ranging ITS applications process and transmit data to reduce traffic congestion, improve traffic management, and enhance quick responses to unexpected incidents.

One of the critical components of public transportation systems is video surveillance, which offers tremendous research potential because it helps with traffic network planning and management. The investigation of algorithms becomes more crucial and necessary as the number of cars on the planet rises. In the past, viewing and interpreting video data required a lot of labor, which was quite wasteful. However, in the last ten years, the price of surveillance cameras has dropped, allowing for the recording and using vast amounts of traffic data. Many computer vision techniques have been created to examine video surveillance data.

Specific surveillance tasks, like vehicle counting [190], license plate recognition [191], and incident detection [192], can be handled by sensor-based or vision-based algorithms in urban traffic systems. However, vision-based approaches may fully benefit from the plethora of visual patterns to identify target objects in a human-like manner. As an illustration, consider the application of vehicle detection. Radar sensor-based techniques can only detect vehicles in a small area. In contrast, vision-based approaches can use a camera to find all vehicles in a large visible area and simultaneously describe additional features of each detected vehicle.

## 3.1 Different techniques for vehicle detection

Vision-based vehicle detection algorithms are classified into three main divisions: motion-based, handcrafted feature-based, and CNN-based approaches.

### *3.1.1 Motion-based approaches*

Motion-based approaches comprise background subtraction, optical flow, and frame subtraction. Background subtraction separates foreground objects from the background using masks. The frame subtraction method subtracts two or three consecutive video frames to detect moving objects. In comparison, optical flow computes the motion vector of each pixel and tracks them. Among all these methods, optical flow is computationally complex and time-consuming. However, the above-mentioned techniques can only classify and detect moving vehicles by capturing images/videos from still cameras.

### 3.1.2 *Handcrafted feature-based approaches*

Handcrafted feature-based approaches contain Histogram of Oriented Gradients (HOG) [193], SIFT [194], and Harr-like [195]. These methods have low feature representation. Although these methods work well even if the data size is small and doesn't require any special hardware, in ML, features need to be extracted by some expert. The algorithm's efficiency depends upon how accurately features are identified and extracted.

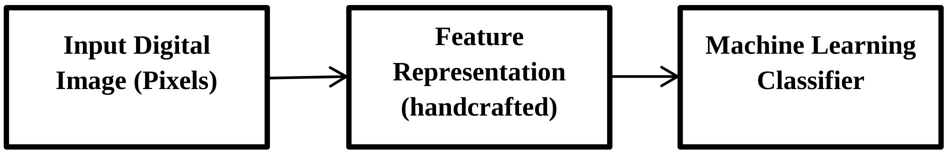



Figure 3.2: Object detection using handcrafted features

### 3.1.3 *CNN based approaches*

With the advancement of image classification algorithms using deep learning, object detection algorithms have also been widely used in recent years. Deep Learning models require a considerable amount of training data as it learns high-level features from it. It reduces the task of developing feature extraction algorithms. But these algorithms highly depend upon high-end machines as these algorithms perform a large number of matrix multiplication operations. Graphical Processing Units (GPUs) can very efficiently optimize these operations. Object detection frameworks based on deep learning are categorized into two classes: two-stage detectors and one-stage detectors.

A sparse set of proposals is generated in a two-stage detector, and features are taken out in the first phase. The motive is to propose high-recall regions so that every object in an image fits at least one of the proposed neighborhoods. Then region classifier predicts the class of each proposal. Two-stage detectors give high accuracy, although more time is taken to produce results. One stage detector directly predicts each location's category of the feature map, producing results in less time.

## 3.2 Convolutional Neural Network

Today's majority of machine learning professionals rely heavily on convolutional neural networks (CNN) [194]. Like a standard multi-layer neural network, a CNN model typically has one or more fully connected layers after one or more convolutional layers (sometimes with a sub-sampling step). Local connections and linked weights, along with some form of pooling, the architecture of a CNN is designed to take advantage of the 2D structure of an input image. CNNs also have fewer parameters and are easier to train than fully-connected networks with the same number of hidden units. The primary functions of a general CNN model are introduced one at a time in the following subsections.

### *3.2.1 Convolution*

Convolution filters are the first layers to accept an input signal. During convolution, the network identifies the input signal using what it has previously learned. The subsequent layer receives the generated output signal. Translational invariance is a convenient characteristic of convolution. It implies that each convolution filter represents an important feature, and the CNN algorithm learns which features make up the resulting reference image. The location of the features has no bearing on the output signal intensity; all that matters is if they are there. Therefore, an object might be present in various positions and yet be recognized by the CNN algorithm.

Other crucial variables are also adjusted, like channel depth, stride, and zero-padding. The number of filters employed for the convolution operation is correlated with the channel depth. As more filters are added, the network gets stronger at extracting picture attributes and identifying patterns in unseen images. The amount of pixels the filter matrix moves over the input matrix is known as the stroke. Filters are moved over the image one pixel at a time when the stride is 1.

### *3.2.2 Nonlinearity Activation*

The activation layer regulates the signal's progression through the network's layers by simulating how neurons activate. Strongly correlated output signals with prior references would activate more neurons, allowing for more effective signal propagation for identification. Rectified Linear Units (ReLU), the most common activation function for modeling signal propagation with CNN, are advised due to their faster training times. An extensive range of intricate activation functions is compatible with CNN.

The element-wise operation (applied per pixel) known as ReLU is used to replace any negative pixel values in the feature map with a value of 0. ReLU is used to bring nonlinearity into the CNN model since most real-world data we would like the network to learn from is nonlinear (Convolution is a linear operation - element-wise matrix multiplication and addition, so we account for nonlinearity by introducing a nonlinear function like ReLU).

### 3.2.3 *Pooling or Sub-sampling*

Convolutional layer inputs can be "smoothed" to lessen their sensitivity to noise and translational fluctuations. A technique known as pooling or sub-sampling can be accomplished by obtaining averages or the maximum over a signal sample. While reducing the dimensionality of each feature map, this spatial pooling still preserves the most crucial data. For the maximum pooling scenario depicted in Figure 3.3, the most significant component from the corrected feature map is chosen inside a spatial neighborhood made up of a 2X2 window. In such a window, it is also possible to compute the average (average pooling) or the total of all components rather than just the largest one. Max pooling has been proven to produce superior performance in real-world settings.

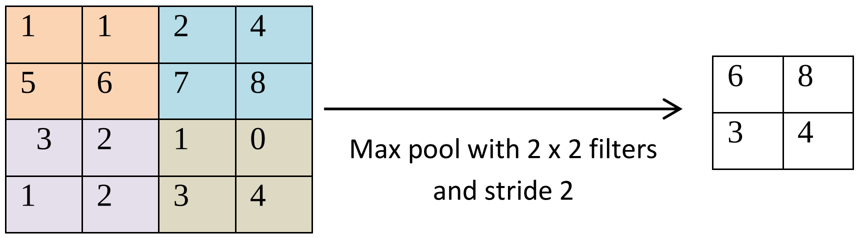



Figure 3.3: Max Pooling Operation

### 3.2.4 *Fully-Connected Layer*

A network's final layers are frequently fully linked, which means that each neuron in the layers before and after them is connected. Significant elements of the input image are shown in the output of the convolutional and pooling layers. The task of the Fully-Connected layer is to categorize the input image into several groups using these qualities and the training dataset. It is also possible to learn nonlinear combinations of these features for data classification if a fully linked layer is included. Most of the convolutional and pooling layer characteristics may be helpful for the classification task, but combining those features may be much more efficient.

### 3.2.5 *Back Propagation*

The backpropagation algorithm is typically the foundation for a CNN model's training procedure [197]. To reduce output error, utilize gradient descent to update all filter weights and parameter values after computing the gradients of the error concerning all network weights via backpropagation.

### 3.2.6 Common Loss Functions

In machine learning, optimization is frequently driven by a loss function that establishes the learning objective by mapping parameter values to a scalar number signifying the "badness" of these parameter settings. Learning aims to determine a weighting that minimizes the loss functions. An "information gain" matrix defines each label pair's " value " as sent to the information gain loss function. Softmax Loss computes the multinomial logistic loss by running real-valued predictions through a softmax to produce a probability distribution over classes.

### 3.2.7 Dropout Operation

A deep learning model can also be trained using the dropout operation approach. Dropout is the term used to describe neglecting neurons while training a specific set of randomly selected. When these units are "ignored," it signifies that during a specific forward or backward pass, they are not taken into consideration. At each training stage, a smaller network is left behind as individual nodes are either eliminated from the network with probability 1-p or kept in the network with probability p.  Dropout aims to prevent over-fitting because a fully-connected layer consumes the majority of the parameters and co-dependency between neurons during training.

### 3.2.8 Proposal Generation

Bounding boxes are generated around the potential objects and further refined. There are different methods for proposal generation, like computer vision-based methods, anchor-based methods, key point-based approaches, etc. In computer vision methods, proposals are generated with high recall using lower-level features. Anchor-based methods are supervised proposal generators. It generates the anchors at different scales and different aspect ratios. A feature map of 256 dimensions is extracted from each anchor and used for the classification and regression layer. The SSD was developed on a similar idea.

### 3.2.9 Learning Strategy

Optimization of localization and classification is challenging in object detection. Various learning strategies are used, like cascade learning, data augmentation, imbalance sampling, localization refinement, etc. Data augmentation plays a vital role in deep learning models as vast volumes of data are required to train these models. In Faster R-CNN, a horizontal flip is performed on training images to increase training data. Imbalance sampling is also a critical issue and is defined as most of the proposals generated are background images and actual objects are very low. In SSD, complex negative sampling was used to fix the ratio of foreground objects and background.

### *3.2.10 Testing Stage*

Predictions made by object detection algorithms are vast in number and duplicate, so they are not directly used for model evaluation. To remove duplicate predictions, i.e., false positives, non-maximum suppression (NMS) is used in the SSD. For each class, the predicted objects, i.e., bounding boxes, are arranged based on their confidence value and the predictions with maximum value are selected. Suppose selected bounding box is called M. After that the Intersection over Union (IoU) of all bounding boxes with M is computed. If IoU is more significant than some threshold value, it will remove these bounding boxes.

$$Score_B = \begin{cases} Score_B & IoU(B,M) < \theta_{test} \\ 0 & IoU(B,M) \geq= \theta_{test} \end{cases} \qquad \text{(i)}$$

NMS will miss an object if it lies within the threshold of M. To overcome these limitations, Navneet et al. [62] proposed soft-NMS which does not discard predictions but decreases the score of detections.

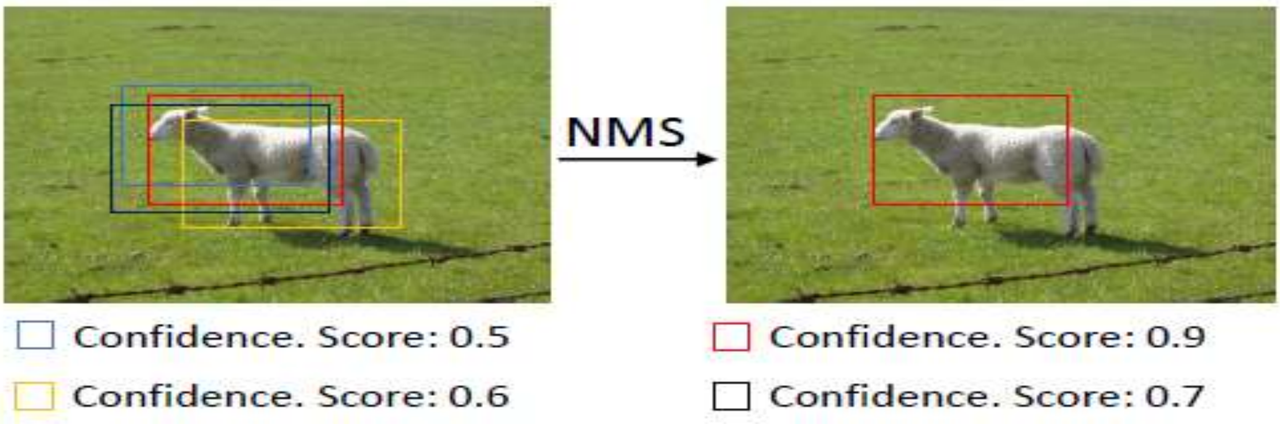



Figure 3.4: Non-Maximum Suppression [50]

Detection is done in two ways, i.e., vanilla object detection (bbox-level localization) and instance segmentation (pixel-level or mask-level localization). The vanilla object detection method uses bbox annotations while IoU is computed between the obtained bounding boxes and the target bounding boxes to compute the model effectiveness.

### 3.2.11 Transfer Learning

In the past, utilized collected training data with labels or without labels was to infer the data using traditional machine learning and data mining algorithms. Transfer learning, however, changes this process since it applies knowledge gained from at least one source assignment to the target work to enhance the experience. The fundamental idea to integrate transfer learning in machine learning was created at the NIPS-95 workshop on "Learning to learn." This workshop's primary focus is on the need for machine-learning techniques that reinforce and retain prior experience-based knowledge.

The main objective of transfer learning is to enhance the experience within the destination task by investing information from the supply task. With the help of transfer to improve learning, there are three standard methods: Firstly, initial performance accomplishable within the target task victimization solely the transferred information, before any more extended learning is executed, compared to the initial performance of an ignorant agent. Secondly, the measure of time it takes to become thoroughly familiar with the objective assignment given the transferred information contrasted with the measure of time to take it without any preparation. Thirdly, the last execution level reachable in the objective task contrasted with the last dimension without transfer.

Transfer learning can be implemented in two ways: Firstly, the extraction of features, which takes the very much prepared model from the source task as a component extractor and relearns the last few expanded layers without changing the first system parameters; Secondly, the calibrating method, which includes a couple of arbitrary statement layers to the pre-prepared systems, in addition, loads of unique layers will refresh in a little learning rate. Image categorization using Inception V3 as a starting point, transfer learning with a warm restart, VR-enabled imitation learning, leveraging modules from pre-trained models, and meta-learning: learning to learn, etc., are a few instances of transfer learning in action.

### 3.2.12 Evaluation Metrics

The task of object detection is typically used to predict the bounding box of the target item. An evaluation metric called Intersection over Union (IoU) is used to gauge how accurately an object detector performs on a specific dataset. This evaluation metric is frequently employed in object detection tasks. More specifically, the ground truth bounding boxes and the projected bounding boxes by a model are needed to apply IoU to evaluate an object detector.

Therefore, computing IoU can be defined as:

$$IoU = \frac{Area\ of\ overlap}{area\ of\ union} \qquad (ii)$$

The accuracy can be determined of the localization by specifying an IoU threshold. Each anticipated box is either a True Positive or a False Positive. The definition of precision (P) is the ratio of the number of True Positives (Tp) to the sum of the True Positives and the False Positives (Fp).

$$P = \frac{T_p}{T_p + F_p} \qquad (iii)$$

The number of True Positives (Tp) over the sum of the True Positives and the number of False Negatives is known as Recall (R) (Fn).

$$R = \frac{T_p}{T_p + F_n} \qquad (iv)$$

The accuracy-recall curve shows how precision and recall are traded off for different IoU thresholds. Low false negative rates are connected with high recall, and low false favorable rates are correlated with high precision. Excellent recall and high precision are both indicated by a high area under the curve. The precision values on the precision-recall curve are averaged to get the average precision when the recall is within the range [0, 0.1,..., 1].

Mean average precision (mAP) [198] is used to trade off accuracy and recall to assess the performance if numerous ground truths for each objects in the image are available.

## 3.3 Proposed Methodology for Vehicle Detection

To train a deep learning model, ‘N’ annotated images {$x_1$, $x_2$, …, $x_N$} are given, and for $i^{th}$ image $x_i$, there are $M_i$ objects belonging to C categories:

$$y_i = \{(c_1^i, b_1^i), (c_2^i, b_2^i), \ldots, (c_{M_i}^i, b_{M_i}^i)\ \} \qquad (v)$$

Where $c_j^i (c_j^i \in C)$ and $b_j^i$ signify categorical and spatial labels of the $j^{th}$ object in $x_i$, respectively. For $x_i$, the prediction shares the same format as $y_i$:

$$y_{pred}^i = \{(c_{pred1}^i, b_{pred1}^i), (c_{pred2}^i, b_{pred2}^i), \ldots\} \qquad (vi)$$

Over C+1 categories, a multi-class classification model is trained, where C refers to actual classes and one background.

### 3.3.1 Data Collection

Training the vehicle detection model with a standard dataset like PASCAL VOC 2007, 2012, and MS COCO 2014 is not an ideal choice [98]. These datasets do not contain all types of vehicle categories. PASCAL VOC 2007 and 2012 consists of only two class labels: car and bus. While MS COCO 2014 comprises three classes (car, bus, and truck). Therefore, four different datasets have been collected to perform experiments: FLIR thermal dataset, FLIR RGB dataset, MB7500, and KITTI dataset. FLIR dataset was released in 2018, consisting of approximately 14K Thermal, RGB images, and 10K videos [199]. The dataset comprises 60% daytime and 40% nighttime images From November to May when the weather is clear to overcast. The MB7500 dataset contains around 7500 photos obtained with a Phantom 4 drone and a high-definition camera in windy conditions. KITTI datasets are captured by driving around Karlsruhe's mid-size city, rural areas, and on highways. Figures 3.5 to 3.8 show the difference between thermal and visible camera images of the same scene at different times and in weather.

(A) (B)

Figure 3.5: Daytime image from the visible camera (A) and Thermal Camera (B) from FLIR Dataset [221]

(A) (B)

Figure 3.6: Sunlight-affected images from the visible camera (A) and Thermal Camera (B) from FLIR Dataset [221]

(A) (B)

Figure 3.7: Morning time image from the visible camera (A) and Thermal Camera (B) from FLIR Dataset [221]

(A) (B)

Figure 3.8: Nighttime image from the visible camera (A) and Thermal Camera (B) from FLIR Dataset [221]

### *3.3.2 Data Annotation*

Data annotation is the process of categorization and labeling of data. In this work, all the images are annotated using the labeling tool into six categories (cycle, two-wheeler, light vehicle, heavy vehicle, bus, and truck) [221].

### *3.3.3 Data Augmentation*

Data augmentation is also used to increase the data diversity for training models. In this work, three transformations have been applied to balance the dataset: horizontal flip, rotation, and Gaussian noise. Figure 3.9 shows an original image of a bus and augmented images.

(a) (b) (c) (d)

Figure 3.9: Augmented Images (a) original (b) flipping (horizontal) (c) rotation (d) gaussian noise [221]

### 3.3.4 Detection of Vehicles using the Ensemble Method

In this paper, two DL models (Faster R-CNN and SSD) have been investigated. In some scenes, both models show duplicate detections, which means the same object is detected in two categories simultaneously, leading to incorrect results. Figure 3.10 shows duplicate detections made by Faster R-CNN in which a total of five objects are present. Out of five objects, one object is detected by the model as a heavy vehicle and a light vehicle, while other objects are not detected at all. Base model predictions have been improved by eliminating duplicate detections, and an ensemble using a majority voting classifier has been implemented.

### 3.3.5 Removal of Duplicate Detections in Faster R-CNN and SSD

To remove the same predictions of the Faster R-CNN and SSD model, obtained predictions are compared to each other based on their bounding box coordinates. If the difference between two coordinates is less than 25 (threshold=25 taken using hit and trial method), then prediction with less confidence score is discarded.


Figure 3.10: Prediction with Duplicate Detections by Faster R-CNN model [221]

### 3.3.6 Ensemble of Faster R-CNN and SSD

The technique of training numerous machine/deep learning models and integrating their outputs is known as ensemble learning. It's typically utilized to boost prediction performance, function approximation, and classification model accuracy. The proposed study has implemented an ensemble of two deep learning models (Faster R-CNN and SSD). Faster R-CNN and SSD are explained as follows:

#### 3.3.6.1 Faster R-CNN

Faster R-CNN is a two-stage detector consisting of a total of three parts. (a) Convolution layers (b) Region Proposal Network (RPN) (c) Classes and bounding box prediction. The architecture of Faster R-CNN is shown in Figure 3.11. CNN layers are used to extract features from the images. RPN predicts the possibility of the presence or absence of an object. It works on the last feature maps of the CNN layers and predicts the bounding box around the possible objects. The last part of the model is used to predict class labels and final bounding boxes.

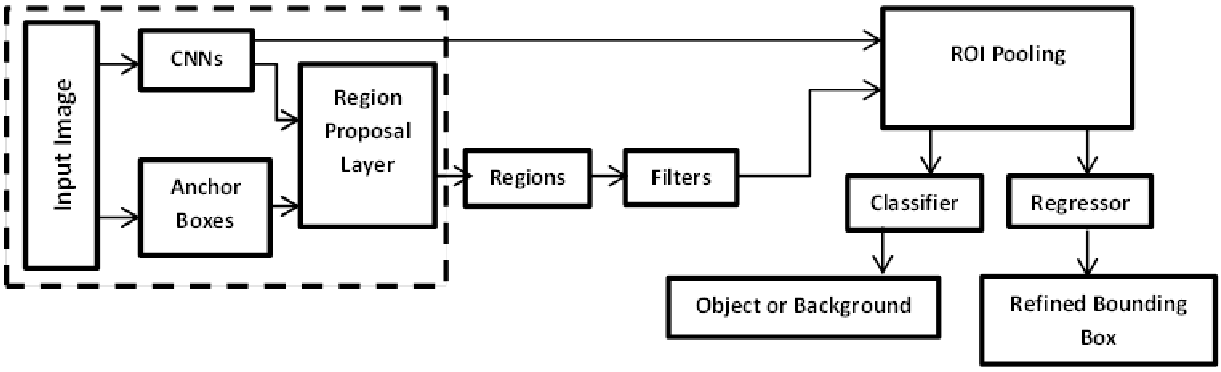



Figure 3.11: Architecture of Faster R-CNN

*3.3.6.2 Single Shot Detector (SSD)*

SSD is a single-stage detector that consists of VGG-16 as a base without a classification layer followed by multi-box convolutional layers. Figure 3.12 shows the architecture of the SSD model. The base model (VGG-16) is used to extract features. After the VGG-16 base model, various convolutional layers with decreasing sizes are placed to detect objects at varying scales and aspect ratios and to predict confidence scores. Initial layers are responsible for detecting small objects, while deep layers detect larger ones. Hard negative mining is also utilized to avoid many negative proposals.

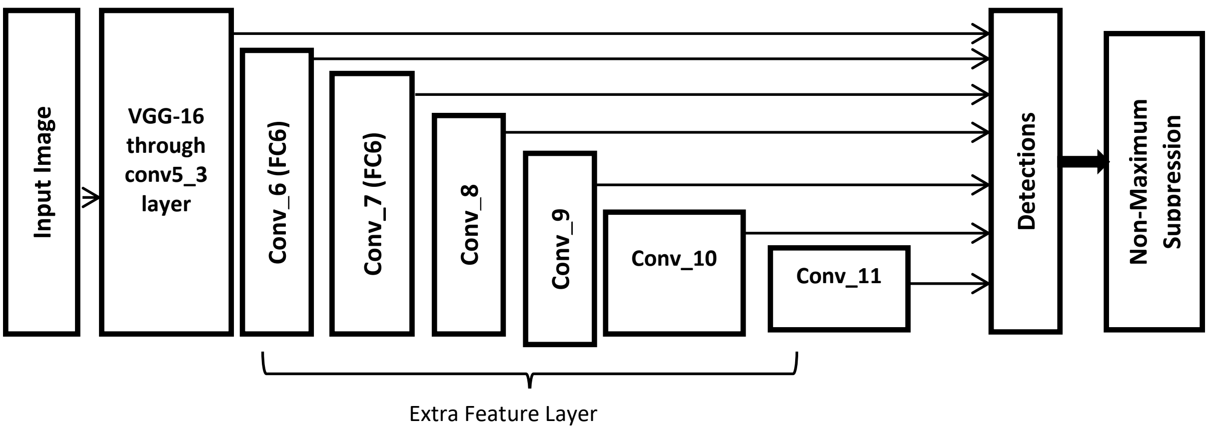



Figure 3.12: Architecture of Single Shot Detector

**Algorithm 1: Proposed Methodology**

**INPUT:** an image

**OUTPUT:** Predicted classes with bounding boxes and their confidence score

[$P_{Ensemble}$ : bbox , class, conf]

Initialize the threshold value. And provide input images to the model.

**Begin**

Obtain the results of base models. Suppose $P_{faster}$ and $P_{ssd}$ are the predictions obtained from faster R-CNN and SSD models, which return bounding boxes coordinates, class of vehicle, and confidence score.

[$P_{faster}$: bbox, class, conf]

[$P_{ssd}$: bbox, class, conf]

If $P_{faster}$. bbox - $P_{ssd}$. bbox <= threshold

If $P_{faster}$.conf1 > $P_{ssd}$.conf2

$P_{Ensemble}$.bbox = $P_{faster}$.bbox

$P_{Ensemble}$.class = $P_{faster}$.class

Else

$P_{Ensemble}$.bbox = $P_{ssd}$.bbox

$P_{Ensemble}$.class = $P_{ssd}$.class

End

End

**End**

Faster R-CNN and SSD model predictions are saved in Pfaster and Pssd, respectively. Predictions made by both models are compared based on bounding box coordinates to identify unique or duplicate detections. If the coordinate values of two predictions have a difference of less than or equal to the threshold, that means both the models have predicted a specific vehicle. The confidence score of $P_{faster}$ is compared to the confidence score of $P_{ssd}$. If the confidence score of $P_{faster}$ is greater than $P_{ssd}$, then the prediction made by $P_{faster}$ is saved to the final prediction, and the $P_{ssd}$ prediction is discarded otherwise, and vice versa. A flow chart of the proposed technique is given in Figure 3.13.

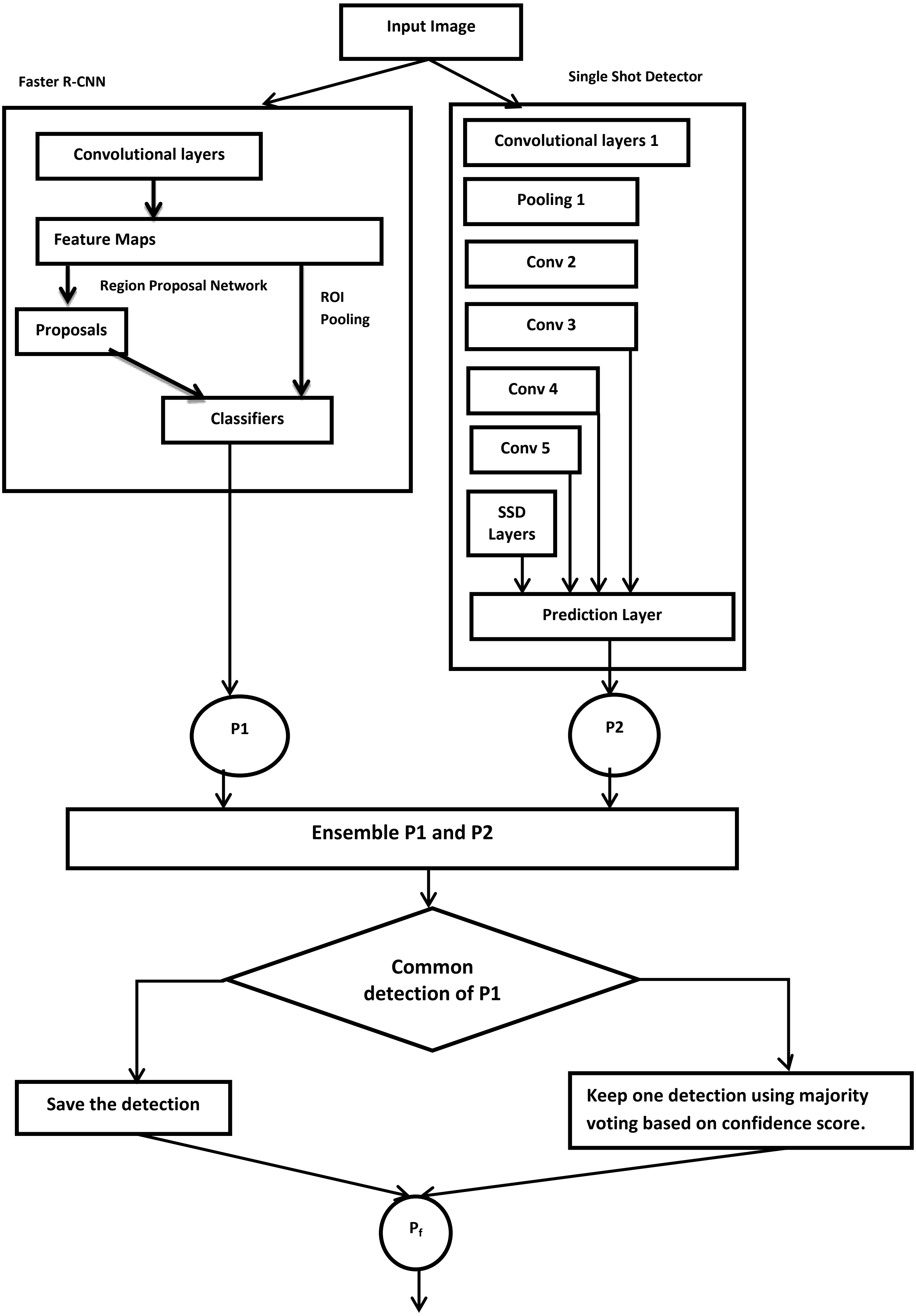



Figure 3.13: working of proposed DL ensemble [221]

## 3.4 Traffic Density Estimation

Road traffic density estimation is an essential factor that helps manage road traffic, road structure, vehicle routing, vehicular network traffic scheduling, reducing pollution, and making an efficient plan of transportation and related policies. Density is the number of total vehicles present on a lane over a time unit. Density is estimated by detecting and recognizing the vehicles. In late years, vehicle detection and classification has become a significant research arena and is used in many applications like intelligent parking systems, automatic toll collection, traffic statistics analysis, driver assistance systems, etc.

Density is calculated as the number of units present on-road rather than the number of vehicles, as each vehicle has a different size and shape. They occupy different areas and take different times to cross the intersection. This research categorizes vehicles into six classes: cycle, two-wheeler, light vehicles (small cars), heavy vehicles (big cars, vans, etc.), bus, and truck. Table 3.1 shows the number of units assigned to each type of vehicle [221].

Predictions made of each model are summed up by multiplying each vehicle with its corresponding units

$$density = \sum predicted\_category_{\mathrm{i}} * n\_units \qquad (vii)$$

Where $predicted\_category_i$ represents the vehicle category predicted by the model and n_units represents the number of units associated with that category.

Table 3.1: Number of units assigned to different types of vehicles

| Vehicle Type | Number of Units |
|---|---|
| Cycle | 1.0 |
| Two-Wheeler | 1.3 |
| Light Vehicle | 1.7 |
| Heavy Vehicle | 2.1 |
| Bus/Truck | 2.3 |

## Chapter Summary

This chapter gives an introduction to object classification, localization, and detection. Basic concepts of a convolutional neural network, transfer learning, and object detection have been covered. Then, different evaluation matrices for object detection are also presented in this chapter.

Finally, the chapter contains the design of an ensemble-based vehicle detection model, a combination of the Faster R-CNN and SSD models. The details of the working of the proposed model are illustrated in the algorithm. Also, the method of computing the traffic density is described, and the number of units assigned to each vehicle type is mentioned.

# Chapter 4

# Detection of Emergency Vehicles Using Radio Frequency Identification (RFID)

There has been a tremendous increase in vehicle traffic with urbanization, industrialization, and population. There is no robust traffic framework today; one approach to convey effective traffic frameworks is through unique control of traffic signals dependent on the traffic size. Moreover, for all need vehicles, for example, ambulances, police vans, and fire engines, no need administrations are given. Therefore, need vehicles must be furnished with specific administrations other than standard administrations.

A traffic signal that works on timers is the most common form of traffic control. These are scheduled to work a certain way at a particular time, regardless of traffic. When congestion occurs for any reason, passengers' traveling time will increase, and emergency vehicles may get stuck in a traffic jam. In this chapter, the study of the detection of emergency vehicles using RFID has been done.

## 4.1 Introduction to RFID Technology

Animal tracking and automated toll collection are just two RFID technology applications that offer automatic object identification [201]. There are two components to this wireless technology: a tag (also known as a receiver) and a reader (i.e., a transmitter). These components employ radio frequency signals to exchange information between a tag attached to an object and a reader incorporated into the environment. RFID tags are embedded computers that enable wireless storage and retrieval of an object's identification information by an RFID reader. They have a small number of features and onboard memory. An RFID reader is an embedded computer that has processing power comparable to a modern desktop computer and can communicate in real time with thousands of RFID tags simultaneously [201]. RFID tags and readers come in various shapes and sizes and can work across frequencies and distances. They also have two different communication methods.

## 4.2 Types of RFID

RFID tags come in three primary varieties: passive, semi-passive, and active. The size and shape of tags can vary from that of a stamp to that of a shoe box, and they can work with various radio signal frequencies (e.g., from a few KHz to a few GHz).

### *4.2.1 Passive RFID*

A passive RFID tag has no internal power source. The incident radio frequency signal emitted by the RFID reader gives passive tags their power for functioning and communication. These are the least expensive (costing only a few cents per tag), have the most extended lifespan, and can initiate conversations with readers independently.

### *4.2.2 Semi-passive RFID*

Semi-passive tags are semi-active or battery-assisted passive (BAP) tags. These operate on the same principle as passive tags but incorporate a battery to increase their communication range, provide memory to the tag, and sometimes add sensors.

### *4.2.3 Active RFID*

An internal battery powers an active tag. As a result, it constantly announced its existence even without a reader. The reader is most dependent on active tags, which have a shorter lifespan and are more expensive (costing a few dollars or more per tag). Therefore, it is only applied in narrowly specialized industrial applications.

## 4.3 Types of RFID Readers

An RFID reader is an embedded device that can function over a wide range of radio signal frequencies (for example, from a few KHz to a few GHz). It comes in various form factors to fulfill the requirements of various use cases (e.g., handheld, desktop, wall-mounted, etc.). An embedded operating system must manage the onboard hardware resources of an RFID reader because the device is embedded. Additionally, an RFID reader includes a variety of communication interfaces (like USB, serial, ethernet, etc.) that enable it to be programmed in a variety of programming languages (like C++, C#, Java, etc.) to read and write tags by the needs of the application. A variety of antennas with various properties can connect RFID readers. Various antennas with various emission patterns (i.e., the form of the radio signal produced by the reader's antenna) can be utilized to link RFID readers. Newer readers can connect to up to four antennae. An RFID reader can read a tag at a range of distances, from a few centimeters for near-field readers to several tens of meters for far-field UHF RFID readers. It is dependent on the radio signal's frequency and intensity.

## 4.4 Types of RFID Tag-Reader Communication Mechanisms

Two main communication methods are employed for tag-reader interaction based on the variation in the temporal radiofrequency electromagnetic fields concerning tag-reader distance. The charge and current components of the electric and magnetic fields are not significantly different when the tag-reader distance is up to two wavelengths of the reader's radio frequency signal. When tag reader distances are close, the combined effects create a near-field. Near-field communication is used to describe tag-reader communication through such a field interaction. The charge and current effects diverge to form a radiative field as the tag-reader distance surpasses the two wavelengths span limit. Far-field communication is the term used to describe this radiative field-based communication approach. Near-field communication is frequently used by RFID tags and readers when they operate at lower frequencies (i.e., a few KHz). In contrast, far-field communication is generally used when they run at higher frequencies (i.e., a few GHz).

## 4.5 Different Ranges of RFID

RFID labels and pursuers must be tuned at a similar recurrence. There are different frequencies that an RFID gadget can utilize. By and large, the most widely recognized are

a) Low recurrence, or LF (125 - 134 kHz)
b) High recurrence, or HF (13.56 MHz)
c) Ultra-high recurrence, or UHF (433 and 860-960 MHz)

Over the different frequencies, radio waves act contrastingly, so it is critical to pick the fitting recurrence for the application.

Low-frequency labels, for instance, have a frequency and can enter delicate metal surfaces all the more adequately. LF RFID frameworks are likewise ideal for perusing high water content articles, for example, natural products or beverages. However, the reading go is just centimeters or centimeters. Commonplace LF RFID applications incorporate creature labeling and get to control.

High-frequency labels work very well for metal items and can be utilized to deal with medium to high-water-content merchandise. HF RFID frameworks ordinarily work in inch ranges; however, they can have a most excellent perusing scope of roughly three feet (1 meter). Following library books, following the patient stream, and travel tickets are the average HF RFID applications.

UHF frequencies commonly offer better understanding and can move information rapidly (for example, peruse a lot more labels each second) than lower or high-recurrence frequencies (inch to 50 + ft. as per the RFID framework arrangement). Since, in any case, the frequency of UHF radio waves is shorter, their sign would be brought down (or debilitated) and not communicated through metal or water. As a result of their high information move rates, RFID labels for some things without a moment's delay are appropriate, for example, merchandise boxes when passing a dock to a distribution center or dashing when crossing an end goal. Other primary uses for UHF RFID incorporate electronic cost assortment and control of stopping access because of the more drawn-out understanding extent.

## 4.6 Proposed Methodology of RFID-Based Emergency Vehicle Detection

During an emergency, the RFID tag is a sensor that passes signals to the lighting system. RFID is a technology used to automatically identify an individual, packet, or item with radio signals. It relies on RFID days to do so. These tiny transponders provide identity information when requested over a short distance. There are at least two parts in most RFID tags. One is an integrated circuit that stores certain information and modulates and demodulates the signal for radio frequency and other special functions. The second antenna is used to receive and transmit the signal. Two RFID tags exist mainly: active battery-containing RFID tags and passive, battery-free RFID tags.

### *4.6.1 Proposed Methodology*

Every vehicle has passive RFID labels with explicit Electronic Product Code (EPC) RFID label numbers. No external force source should be connected. Subtleties, for example, the vehicle number, vehicle type, and proprietor data are likewise put away on each RFID tag. The Ultra-High Frequency (UHF) recurrence band recognizes labels in the RFID radio wire. The module peruses this information and is then sent for additional preparation to the worker. Table 4.1 shows the details of emergency vehicles taken for the experiment [219]. The proposed system includes two central units linked together: emergency vehicle detection and vehicle priority based on the type of vehicle.

Table 4.1: Details of Emergency Vehicles

| EPC | Vehicle Number | Vehicle Type | Owner's Details | Priority |
|---|---|---|---|---|
| 1011 | XXX | ambulance | XXX | 1 |
| 1012 | XXX | fire brigade | XXX | 2 |
| 1013 | XXX | ambulance | XXX | 1 |
| 1015 | XXX | ambulance | XXX | 1 |
| 1016 | XXX | police | XXX | 3 |
| 1017 | XXX | fire brigade | XXX | 2 |
| 1018 | XXX | ambulance | XXX | 1 |
| 1019 | XXX | police | XXX | 3 |
| 1020 | XXX | police | XXX | 3 |

#### *4.6.2 Detection of Emergency Vehicle*

This system is based on an RFID tag installed in emergency vehicles. During an emergency, it is used as a tracker. It means that the driver activates the RFID tag, which is then detected by RFID readers located a few meters away at the junction. These readers then continuously transmit the signals to the intersection where the traffic lights are controlled. The reader starts detecting signals, and the conditions return to normal as soon as the emergency vehicle is tagged into the junction.

Access control is used to detect IDs entering or leaving the RFID reader area. The signals are passed to the junction unit following identification by RFID. The entire detection unit is shown in Figure 4.1. Following are the steps performed in the proposed algorithm.

1. The transceiver collects signals sent by the RFID reader. After recognizing the received signals, the system identifies the type of emergency vehicle and assigns priority to it. For example, suppose more than one different or the exact emergency vehicle is present at different lanes, then according to their type. In that case, a numeric value, say 1 for an ambulance, 2 for the fire truck, and 3 for a police van.
2. The system then prioritizes the roadsides accordingly.

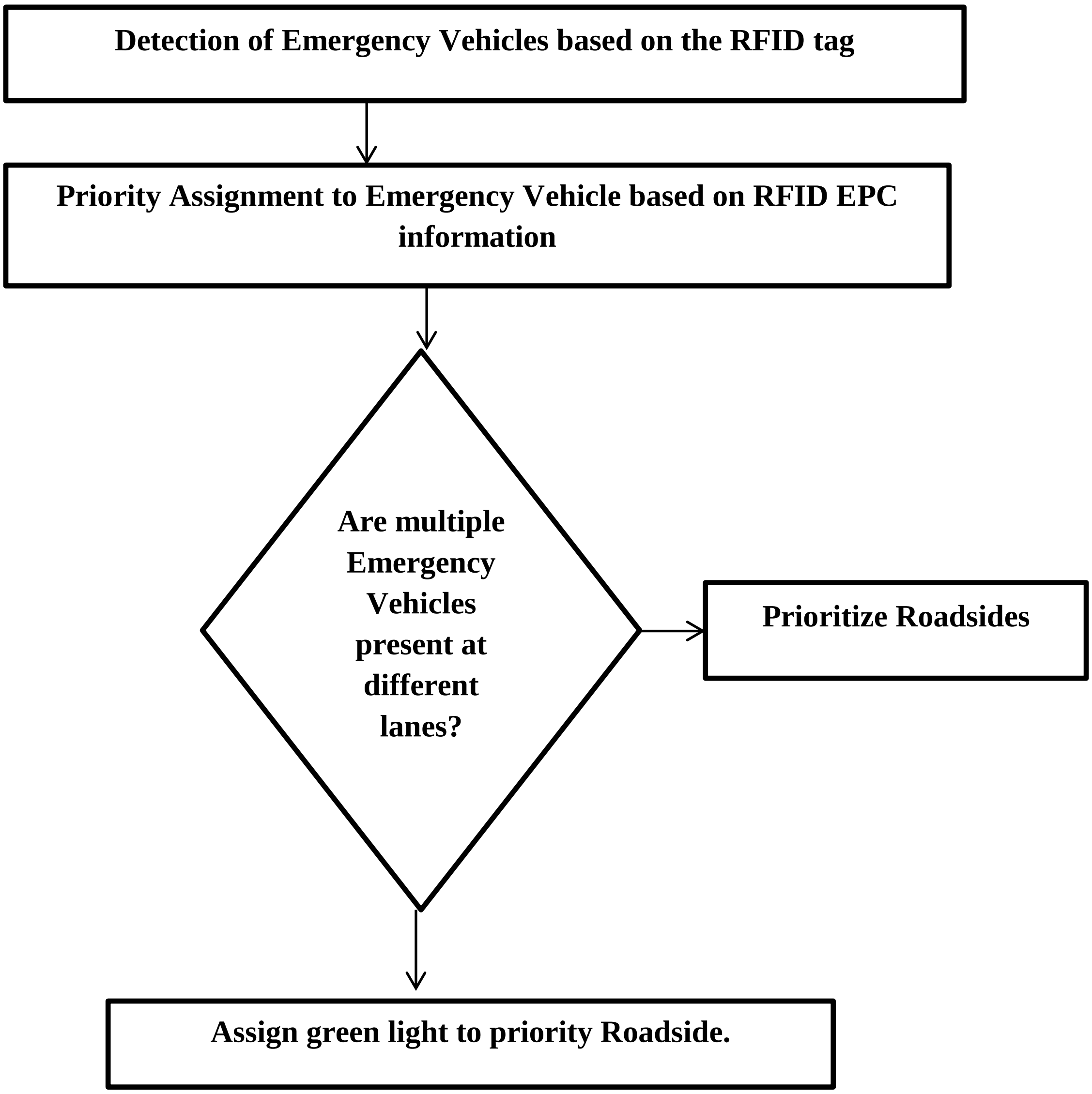



Figure 4.1: Flow Chart of Proposed Methodology for RFID-based Emergency Vehicle Detection [219]

## Chapter Summary

This chapter covers basic concepts of RFID technology, components of RFID, and RFID based on frequencies. Ambulances, fire trucks, and police vans are considered emergency vehicles. It is assumed that RFID tags are attached to each emergency vehicle. When an emergency vehicle approaches the intersection, the RFID reader reads the RFID tag and checks the priority of the emergency vehicle assigned to them.

This method presents a methodology for the arranging of traffic crisis vehicles. The proposed crisis location and organizing framework depend on radio recurrence ID. The crisis vehicle location module it will guarantee that the crisis vehicles will arrive at the goal as quickly as time permits. Each RFID tag is gathered and progressing by the system.

# Chapter 5

# Acoustic-Based Emergency Vehicle Detection

An audio segment that may be identified as a unique idea in an audio stream is referred to as a sound event [201]. We frequently encounter sound events in our daily lives, such as a doorbell, car engine, footfall, keyboard sounds, etc. Additionally, regardless of its substance, such as style, notes, or words, speech and music can be generally regarded as sound occurrences. Identifying the beginning and end of each sound event in an audio signal and connecting them to their appropriate textual labels is known as sound event detection (SED). SED's primary objective is to identify the sound events in the audio input accurately.

The two primary subcategories of SED are monophonic and polyphonic [202]. Regardless of the number of sound events occurring at any given time, monophonic SED systems can only identify a single sound event (typically the most prominent) at a time. Because simultaneous sound events frequently occur in real life, the limit on the number of detected sound events is a drawback for the practical applicability of such systems. Consider the simultaneous presence of car horns, sirens, and loud human speech in an audio signal captured in a crowded street. On the other hand, the objective of polyphonic SED is to identify numerous simultaneous sound events at any particular time, which is more appropriate for real-world applications.

## 5.1 Sound Representation

The sound events in a real-world setting or a recording studio are digitally recorded to provide the audio signals for SED. Since no processing is done on the signal before it is used to represent a sound event, the time domain representation of a sound event is regarded as the lowest-level representation. However, this representation is mainly unnecessary to determine the belongingness of an object with its sound event for classification. To represent audio data for SED, specific acoustic properties are frequently extracted. Since the signals from the same sound event frequently share components in the frequency domain, the acoustic features are mostly retrieved in the frequency domain.

Furthermore, compared to time domain representation, frequency domain representation is more noise-resistant and compact. The degree of abstraction of the sound representation depends on how many processing steps were applied to the time-domain input before acoustic features were obtained. For instance, the histogram of gradients (HOG) feature and the mel-frequency cepstral coefficients (MFCC) feature, described below, are regarded as higher-level representations because they require multiple frequency domain processing steps to calculate, making the representations more abstract.

## 5.2 Stages of Acoustic Feature Extraction

Frame blocking, windowing, and frequency spectrum calculation are the three primary phases of acoustic feature extraction in the frequency domain. The signal should be presumptively able to model a sum of stationary sinusoids for the Short-time Fourier Transform (STFT) to acquire the frequency spectrum. The signal is first divided into brief frequency intervals to calculate the frequency of audio signals; the spectrum calculates the frequency of the spectrum. Frame blocking is the practice of doing this. Based on the frame length, there is a trade-off between frequency and time resolution. The increase hampers time resolution in frequency resolution that comes with more extended frames. As a result, the choice of frame duration depends on the current machine hearing task. Frame lengths between 20 and 50 ms are frequently chosen for SED. An overlap between the frames 25% and 50% of the frame length is frequently chosen to achieve better results. Then, a window function is multiplied by each short-time frame signal. Windowing is the technique used to prevent discontinuities at the frame's borders from affecting the estimation of the frequency spectrum. Hamming, Hann, and Blackman functions are frequently used for SED windowing. Finally, the frequency domain representation of each short-time frame signal is obtained using the discrete Fourier transform.

### *5.2.1 Spectrogram*

The term "spectrogram" refers to the time-frequency domain feature matrix created by concatenating the frequency domain feature vectors for each recording's subsequent time frames. The foundation of a good representation of SED is frequently the spectrogram of an audio signal. The spectrogram has complex values because the Fourier transform is a complex-valued function. However, most machine learning techniques are only intended to function with input that has real values. Therefore, the phase information needs to be more frequently addressed in machine hearing since it is thought to be less informative [203], and only the magnitude of the spectrogram is employed.

Due to the linear frequency resolution of the Fourier transform and the fact that sound occurrences frequently have substantially higher energy levels at lower frequency levels, these lower frequency components predominate as valuable features. By using the logarithm to create log magnitude spectrograms, the dynamic range of the linear magnitude spectrogram can be reduced.

The advantages of using spectrograms as the audio representation for SED are as follows. Based on the relative energy distribution in the frequency domain, spectrograms offer more compact and comprehensive information about sound events when compared to raw audio signals in the time domain [204]. Additionally, because spectrograms are multi-dimensional like images, the extensive machine learning research conducted for tasks based on image categorization applies to SED. Because environmental noise is frequently restricted to lower frequencies, the resultant SED performance is better than raw audio signals. Spectrograms are more resilient to noisy environments than time-domain audio signals.

### *5.2.2 Mel spectrogram*

Numerous spectrogram representation techniques are grounded in auditory perception in people. According to empirical findings, humans do not perceive sounds through a linear frequency scale, and we are more sensitive to changes in the lower frequency range than the higher frequency range.

The Mel scale is a non-linear frequency scale in which the human ear adjusts the pitches to appear equally spaced [205]. The mel spectrogram and mel-frequency cepstral coefficients are two examples of mel scale-based sound representations (MFCCs). The magnitude spectrogram is applied across the mel filter bank at each time frame to produce the mel spectrogram, a matrix of mel band energy feature vectors concatenated for successive time frames. Mel filterbank is a collection of triangle filters that use the mel scale and whose bandwidths increase with higher central frequencies. Higher frequency resolution is produced in the lower frequency range, and vice versa. Taking the logarithm to compress the dynamic range transforms the mel spectrogram into the log mel spectrogram, which is frequently processed further. Log mel spectrograms are the most widely used sound representation for SED tasks. They have been included in numerous cutting-edge techniques for polyphonic SED, uncommon SED, and SED employing weakly labeled data. The range of 40 to 80 mel filterbanks is frequently chosen for SED, likely fewer than the frequency bins utilized in STFT. Therefore, compared to a magnitude spectrogram, a mel spectrogram offers a more condensed picture.

### *5.2.3 Mel Frequency Cepstral Coefficients (MFCC)*

Applying Discrete Cosine Transform (DCT) to the log mel spectrogram yields MFCCs [206]. Mel filters overlap, which causes the outputs of neighboring filterbanks to correlate. As a result, DCT is roughly used to decorrelate the log mel spectrogram features. The higher MFCC coefficients are frequently ignored because they need more information. Additionally, the first coefficient needs to be more frequently addressed because it just equals the average log energy and says nothing about the spectral properties. The top 10–16 coefficients for each short period are acoustic properties.

## 5.3 Detection of Emergency Vehicles using Siren Sounds

Special siren sound signals are used by emergency vehicles to identify them on roads. However, traffic jams or road emergencies can cause emergency services to be delayed. Thus, to avoid delays due to red signals at the traffic signal intersection, EVD has been focused on their siren sounds. Siren sounds have been separated into three categories: ambulance, fire truck, and police car. Each country has its regulations on the siren sound type and frequency band.

Generally, the sirens are warning signals issued in an emergency and standardized by the International Organization of Standards (ISO), and ISO 7731 [207] gives important guidelines for warning sirens. Audio recognition using an ensemble of deep learning models is the primary approach used in this paper. Before performing the recognition task, audio feature extraction methods are used to obtain valuable features in the time domain and frequency domain. Intelligent transport systems can use the application of the EVDs system. Traffic controllers can integrate siren detection to prioritize direction with emergency vehicles by altering the signal and estimating the timing of the green signal accordingly.

### *5.3.1 Data Collection*

The experimental dataset is collected from Google Audioset Ontology [208] which contains sound events in a hierarchal arrangement. This ontology consists of an animal, human, environmental, musical, and miscellaneous sounds. It includes the siren sounds of four vehicles, i.e., Police Car, Ambulance, Fire Engine, and Civil Defence Siren, in video format. Information about the video is available in a CSV file which contains the YouTube link to the video, starting time, ending time, and label. From the dataset, the video of three types of emergency vehicles (Ambulance, Police car, Fire truck) has been downloaded by using two python libraries, i.e., "pafy” and “youtube_dl." The whole audio file is of no use, so the siren sound of the vehicle is clipped from downloaded files using the "moviepy” library.

### *5.3.2 Feature Extraction*

Although many feature extraction methods are available to extract features from audio data, Mel Frequency Cepstral Coefficient (MFCC) is used in this work. It has extracted 39 different features from the dataset where the first feature corresponds to the audio pitch, and 12 of them are related to the amplitude of frequencies. The flow chart of the feature extraction is given in Figure 5.1. The "Librosa" [209] library is used to extract information from audio files. The final shape of the feature vector and target is (259169,40) and(1,301), respectively. Figure 5.2 shows the waveform of the audio of the police car siren sound.

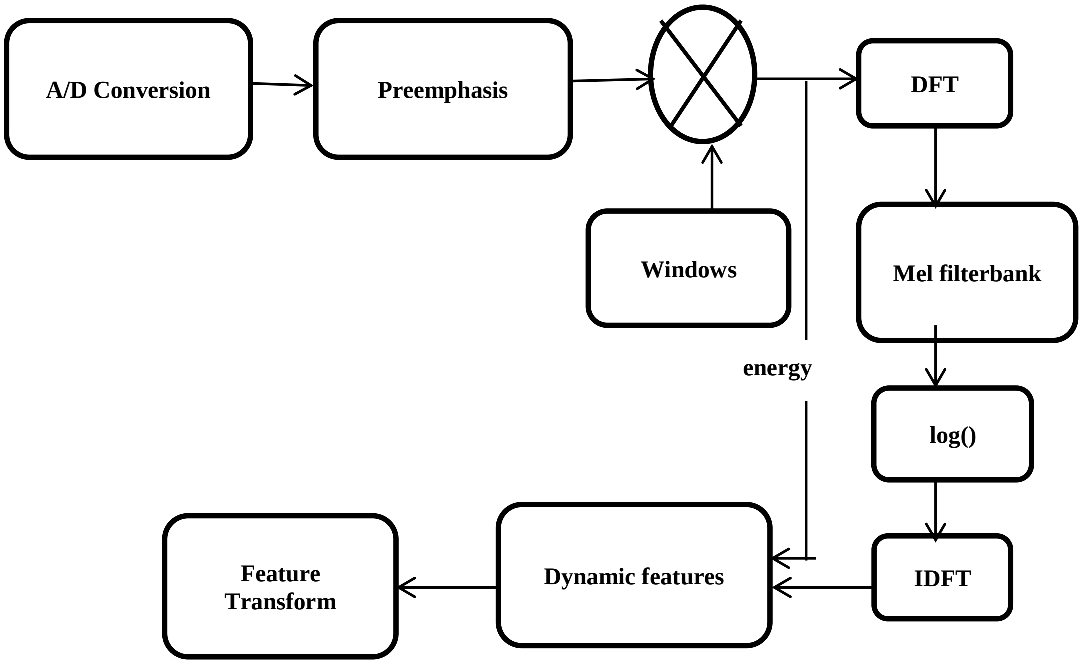



Figure 5.1: Flow Diagram of feature extraction

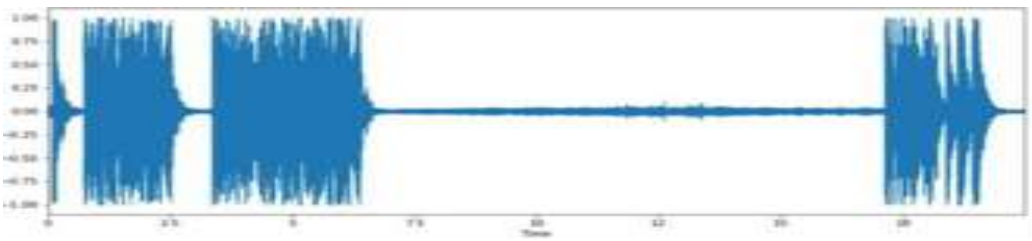


Figure 5.2: Waveform of police car siren [218]

### *5.3.3 Hyperparameter Tuning*

In this work, three types of deep learning models have been investigated and analyzed. When dealing with deep learning models, selecting the appropriate layers and network parameters to optimize its performance is necessary. Hence, a series of experiments have been conducted. The impact of the various layers and parameters is investigated in all three models. Following that, appropriate models from each configuration to acoustic-based EVD have been selected. Tables 6, 7, and 8 show the training and testing accuracy on different layers and parameters used in FCNet, CNN_Net, and RNN_Net models [218].

For implementing deep learning architectures, “TensorFlow" has been used. It is a free and open-source library for mathematically extensive programming, mainly focusing on machine learning and neural networks, developed by Google. All the models have been trained on Google Colaboratory, supporting GPU (Graphics Processing Unit), free for public use. In all the configurations, the Relu activation function is used at the hidden layer, and in the output layer, Softmax activation is applied. The models are trained using the Adam optimizer [210] with a learning rate of 0.001, decay of 0.0001, and categorical cross-entropy loss.

In this work, a recurrent neural network and an ensemble of three different deep-learning models have been created and evaluated to classify the siren sounds of emergency vehicles. Base estimators of the proposed ensemble consist of a fully connected neural network, CNN, and RNN. The details of the base estimators are as follows:

a) *Fully Connected NN (FCNet)*: This architecture is purely based on dense layers without convolutional layers. This network is evaluated with the different numbers of fully connected layers up to 8 with various parameters for selecting the best model.
b) *Convolutional NN (CNN_Net)*: This architecture consists of various 2D convolutional layers, filters, and a 4X4 kernel size. After convolutional layers, the max-pooling layer is used to prevent overfitting. Further, the dropout layer with a 0.25 parameter is applied after the dense layer.
c) *Recurrent NN (RNN_Net)*: This architecture is a recurrent neural network (RNN) that consists of a different number of long short-term memory (LSTM) layers with a different number of neurons.

Table 5.1: Layers and parameters in a multilayer fully connected neural network with different numbers of fully connected layers

| Layer | FC Layer-2 | FC Layer-3 | FC Layer-4 | FC Layer-5 | FC Layer-6 | FC Layer-7 | FC Layer-8 |
|---|---|---|---|---|---|---|---|
| Input | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| FC-1024 | 35267584 | 35267584 | 35267584 | 35267584 | 35267584 | 35267584 | **35267584** |
| FC-512 | 524800 | 524800 | 524800 | 524800 | 524800 | 524800 | **524800** |
| FC-512 | 262656 | 262656 | 262656 | 262656 | 262656 | 262656 | **262656** |
| FC-512 | ---- | 262656 | 262656 | 262656 | 262656 | 262656 | **262656** |
| FC-256 | ---- | ---- | 131328 | 131328 | 131328 | 131328 | **131328** |
| FC-256 | ---- | ---- | ---- | 65792 | 65792 | 65792 | **65792** |
| FC-128 | ---- | ---- | ---- | ---- | 32896 | 32896 | **32896** |
| FC-64 | ---- | ---- | ---- | ---- | ---- | 8256 | **8256** |
| FC-32 | ---- | ---- | ---- | ---- | ---- | ---- | **2080** |
| Output-3 | 1539 | 1539 | 771 | 771 | 387 | 195 | **99** |
| Total Parameters | 36,056,579 | 36,319,235 | 36,449,795 | 36,515,587 | 36,548,099 | 36,556,163 | **36,558,147** |
| Training Accuracy % | 100 | 99.58 | 100 | 99.58 | 100 | 100 | **100** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Testing Accuracy % | 60 | 70 | 75 | 84 | 92 | 94.6 | **96.4** |

Table 5.2: Layers and parameters in a convolutional neural network with various 2D convolutional layers

| Layer | Conv_Layers-2 | Conv_Layers-3 | Conv_Layers-4 | Conv_Layers-5 | Conv_Layers-6 |
|---|---|---|---|---|---|
| Input | 0 | **0** | 0 | 0 | 0 |
| Conv 4X4 – 32 | 544 | **544** | 544 | 544 | 544 |
| Conv 4X4 – 32 | 16416 | **16416** | 16416 | 16416 | 16416 |
| Conv 4X4 – 64 | ---- | **32832** | 32832 | 32832 | 32832 |
| Conv 4X4 –64 | ---- | **----** | 65600 | 65600 | 65600 |
| Conv 4X4 – 128 | ---- | **----** | ---- | 131200 | 131200 |
| Conv 4X4 – 128 | ---- | **----** | ---- | ---- | 262272 |
| FC – 512 | 564,265,472 | **281805312** | 70451712 | 35062272 | 6947328 |
| FC- 64 | 32832 | **16416** | 32832 | 32832 | 32832 |
| Output 3 | 195 | **99** | 195 | 195 | 195 |
| Total | 564315459 | **281871619** | 70600131 | 35341891 | 7489219 |
| Training Accuracy % | 100 | **99.58** | 93.3 | 95 | 95 |
| Testing Accuracy % | 61 | **92.4** | 85.3 | 88.6 | 84.4 |

Table 5.3: Layers and parameters in an RNN with different long short-term memory (LSTM) layers

| **Layer** | **LSTM_Layer-2** | **LSTM_Layer-3** | **LSTM_Layer-4** | **LSTM_Layer-5** | **LSTM_Layer-6** |
|---|---|---|---|---|---|
| Input | 0 | 0 | 0 | **0** | 0 |
| LSTM 32 | 9344 | 9344 | 9344 | **9344** | 9344 |
| LSTM 32 | 8320 | 8320 | 8320 | **8320** | 8320 |
| LSTM 64 | ---- | 24832 | 24830 | **24832** | 24832 |
| LSTM 64 | ---- | ---- | 33024 | **33024** | 33024 |
| LSTM 128 | ---- | ---- | ---- | **98816** | 98816 |
| LSTM 128 | ---- | ---- | ---- | **----** | 131584 |
| FC 128 | 4224 | 8320 | 8320 | **16512** | 16512 |
| Output 3 | 387 | 387 | 387 | **387** | 387 |
| Total | 22275 | 51203 | 84227 | **191235** | 322819 |
| Training Accuracy % | 84.07 | 89.6 | 92.2 | **98.7** | 90.4 |
| Testing Accuracy % | 61.29 | 75.7 | 85.2 | **94.5** | 84.1 |

### *5.3.4 Ensemble Model*

After evaluating and investigating various experimental results of different configurations of three models, an ensemble has been designed, making the prediction based upon majority voting. The base network of FCNet contains eight layers, CNN_Net consists of 3 layers, and RNN_Net comprises five layers. The proposed ensemble network's architecture is shown in Figure 5.3.

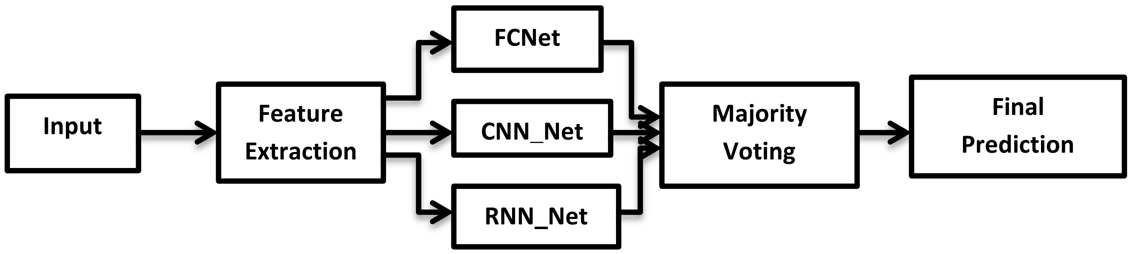



Figure 5.3: Architecture of Proposed Ensemble Model [218]

**Algorithm 5.1: Proposed Methodology for EVD**

**Input**: An audio file

**Output**: The predicted class of emergency vehicle.

**Begin**:

I. Extract features from the given audio file using the MFCC technique.
II. Provide extracted features to three base models, i.e., FCNet, CNN_Net, and RNN_Net, and store their predictions: y_FCNet, y_CNN_Net, and y_RNN_Net, respectively.
III. Apply majority voting on the obtained predictions: mode(y_FCC_Net, y_CNN_Net, y_RNN_Net) and return final prediction

**End**

## Chapter Summary

In this chapter, an introduction to sound-based event detection is covered. Different feature extraction methods from audio files are also discussed.

Acoustic-based models are also a better choice as compared to image-based detection models. As emergency models move at high speed, it is difficult to capture the image of the emergency vehicle using the cameras. But emergency vehicles give warnings from long distances that the system can easily capture and process.

This chapter introduces an ensemble of deep learning-based models for acoustic based emergency/high priority vehicle detection. The suggested model consists of fully connected layers, CNN and RNN models. Models have been trained on MFCC features extracted from collected data.

# Chapter 6

## Green Signal Optimization Using Adaptive Neuro-Fuzzy Inference System

A sound transport system is essential for a country's a trade and industry growth and development. But in many developing countries, inadequate transport facilities lead to road accidents and fidelities causing harm to life and environmental pollution.

Traffic congestion is a significant issue in such countries as India. Road congestion is a complex process in which vehicles are clogged, and there is very slow or no movement. India alone accounted for about 10% of the world's road casualties [211]. Generally, road environs and human and vehicle interaction are driving road systems. According to a survey, the road environment contributes 28%, vehicle factors 8%, and human factors 95% to road accidents with overlapping effects [1]. Traffic congestion not only wastes people's time but also leads to environmental pollution, health hazards, and the wastage of fuel. The critical issue is managing road capacity with the supply-demand equation, properly deploying traffic control devices, and establishing intelligent transportation systems. Nowadays, fixed timer controllers with three-color traffic signals are used at intersections. Although these systems eliminate a person's intervention, they cannot work on real-time data nor identify priority vehicles. The waiting time for the vehicles is more in this system. In image processing-based systems, cameras are used to analyze the traffic density and the presence of vehicle density. But camera resolution and taking images/videos at odd times, like at night, is the major challenge. Fuzzy systems use predefined fuzzy rules for activating green signals depending on vehicle density, flow rate, etc. Other than these techniques, Q-learning is widely used to improve traffic conditions. Q-Learning comprises an online learning method built upon the Markov Decision Process, which learns about the environment from previous experience without the involvement of a mathematical model.

In all the techniques discussed above, the model is designed by analyzing the traffic pattern at a particular intersection. But traffic patterns keep on changing with time. Thus, the model fails to give the expected performance after some time. In this work, an ANFIS system is proposed, which can adapt itself to changing traffic conditions. ANFIS is a fusion system using neural networks and fuzzy logic, which can self-learn and perform reasoning that is easy for humans to understand. A neural network is a mathematical model which learns from previous experiences. A fuzzy system can use human reasoning to deal efficiently with uncertain data. In fuzzy logic, linguistic variables are used, and rules are designed to train the system. Rules of the fuzzy system are fixed, and these rules are not adaptable. In the proposed method, the 4-lane intersection is considered, as shown in Figure 6.1, where the traffic can pass to its left side directly. For the right side direction and in the forward direction, vehicles need to wait for the green signal.

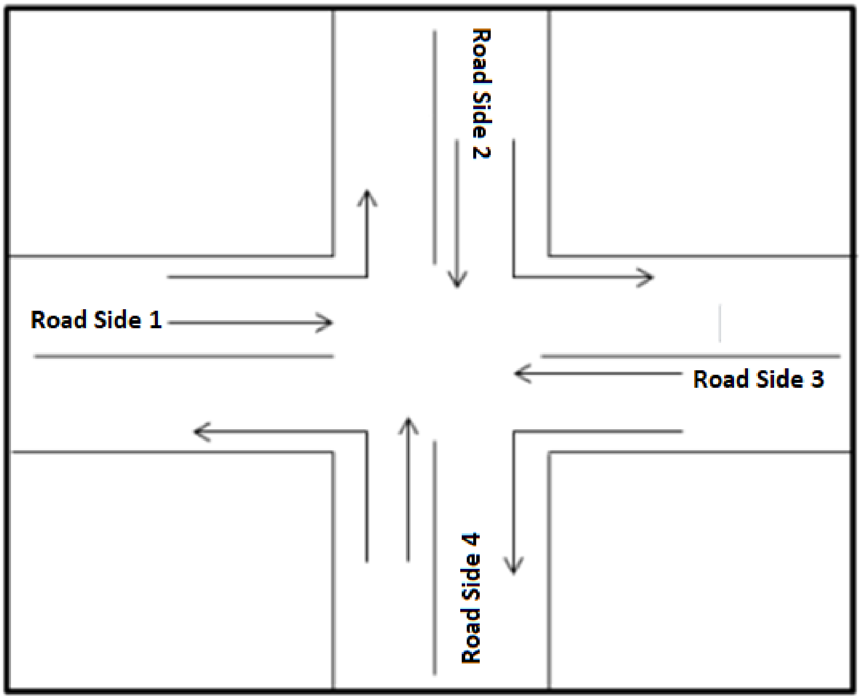



Figure 6.1 4-way isolated intersection [220]

## 6.1 Motivation

Intelligent traffic control algorithms can be designed to increase the transportation system's supply if it is known how congestion propagates with time and space. In this work, with flow rate and density at the current lane, density at the adjacent lane is also considered. If the density at the adjacent lane is very high, then there is no use in providing high green signal timing at the current lane as well as it increases the waiting time of the vehicles at the adjacent lane. Rather than giving priority to the dense lane, the round-robin algorithm is used in the proposed method so that each lane gets equal priority with an appropriate time stamp. If green signal timing is provided to the lane with maximum density, then waiting time for vehicles on the less dense lanes will increase. Emergency vehicles like ambulances, fire brigade, and police vans must go with zero or minimal delay. In the proposed system, priority is given to such vehicles. If any high-priority vehicle is present, then the working of the round-robin algorithm is prompted, and a green signal is provided at the lane having a priority vehicle. A minimum 15 seconds green signal is assigned to each lane. If no vehicle is present on a lane, then the green signal duration is reduced to zero for that lane.

## 6.2 Proposed System for Green Signal Optimization

Traffic congestion is a very complex and unpredictable problem. So it is challenging to manage the vehicles on the road and at intersections with a fixed timer system. Also, in case of emergencies like an accident or heavy jams, the intervention of traffic police is required, who manually handle the situation by taking the appropriate decisions. Soft computing techniques can be used to handle such issues as it exploits the tolerance for vagueness, ambiguity, and partial truth so that a robust, cost-efficient, and better understanding of reality can be accomplished. The role model for soft computing is the human mind. Neural networks, Fuzzy logic, and genetic algorithm are the principal partners of soft computing. As the neural network can learn and self-adaptability, a fuzzy system deals efficiently with the vagueness and fuzziness of natural systems by using if-then rules. A hybrid approach consisting of the NN and fuzzy logic (FL) has been considered, that is, Adaptive neuro-fuzzy inference system (ANFIS).

The proposed model considers the traffic on the current and adjacent roadsides. Lanes are selected using the round-robin (RR) algorithm. According to the round-robin algorithm, each roadside will periodically get the green signal: roadside 1, roadside 2, lane 3, lane 4, and then lane 1, respectively. A fixed green time signal of 15 seconds is assigned to each lane. If no vehicle is present on a particular lane, then a green timing signal is not provided to that lane and shifted to the next lane. The round-robin algorithm is not applicable in the presence of emergency vehicles. Instead, a green signal is provided at the lane on which such a vehicle is present. After the vehicle's passage, the round-robin algorithm's operation is resumed again.

### *6.2.1 ANFIS Model*

An integrating technique comprising of NN and FL called ANFIS is used as a principal tool in the proposed method. The proposed system is designed using the fuzzy rules extracted from the training data. Then ANFIS model is designed by tuning the rules of the fuzzy system using NN. Here, the Sugeno-based ANFIS system is implemented to compute the results.

In the proposed system, vehicle density at the current roadside, vehicle density at the adjacent roadside, and flow rate is taken as input, and output is the green timing extension.

Vehicle density gives the number of vehicles present in one kilometer.

$$k = \frac{m}{L}\frac{veh}{Km} \qquad (vii)$$

Where 'L' is the length of the lane and the 'm' the number of vehicles present in the area covered by 'L' .Flow rate defines the vehicles passing the intersection during the green signal phase.

$$f = \frac{n}{T}\ veh/\sec \qquad (viii)$$

Where 'T' on represents vehicles passing in time T seconds.

A triangular membership function is used to represent the actual values of input and output. Output is defined as a constant value. The proposed model defines input variables from 0 to 110 vehicles /km. Input variables are classified in five different ranges, from very low to very high, as shown in Table 6.1. According to the first-order Takagi and Sugeno's model, the output of each rule can be represented as follows:

$$y = a_i I_1 + b_i I_2 + c_i I_3 + d_i \qquad (ix)$$

Where i= 1, 2, 3…..125 and ai, bi, ci, and di are the coefficients.

Figure 6.2 shows the structure of the proposed system. The different layers of this structure are discussed below:

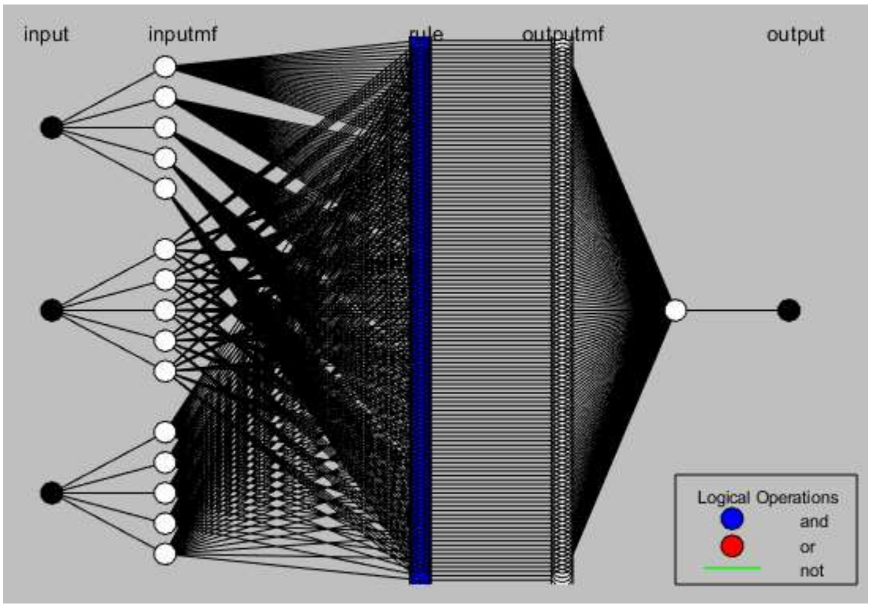



Figure 6.2: Structure of the proposed system [220]

Layer 1: The number of neurons at this layer equals the number of input variables, i.e., three equal to inflow rate, density at the current lane, and density at the adjacent lane.

Layer 2: In the proposed model, five membership functions are assigned to each input variable. Thus, the total number of neurons is 5+5+5=15.

Layer 3: 125 fuzzy rules have been designed in the proposed model. Thus, 125 neurons are used at this layer.

Layer 4: The number of neurons in this layer equals the number of membership functions used for the output variable. So, 125 neurons are used in this layer to represent output membership functions.

Layer 5: This layer consists of only one fixed node that sum up all incoming signals to generate final output.

Table 6.1: Membership Functions for Input Variables

| **Membership Function** | **Flow Rate (Vehicles/Sec)** | **Density at current lane(Vehicles/Km)** | **Density at adjacent lane(Vehicles/Km)** |
|---|---|---|---|
| Very Low | 0-30 | 0-30 | 0-30 |
| Low | 20-50 | 20-50 | 20-50 |
| Average | 40-70 | 40-70 | 40-70 |
| High | 60-90 | 60-90 | 60-90 |
| Very High | 80-110 | 80-110 | 80-110 |

The proposed algorithm is divided into three parts: (i) selection of the lane, (ii) extension in the green signal timing, and (iii) training of the system. The flow chart of the proposed system is shown in figure 6.3 [220].

*a) Selection of the lane*: Normally, lane selection is made using the round-robin algorithm. Four input values are provided to the system, i.e., traffic inflow rate, number of vehicles waiting on the current lane, the density of the vehicles on the adjacent lane, and the presence of the emergency vehicle. Values for the next cycle are provided to the system during the yellow light. If the emergency vehicle is present on any lane, then the round-robin algorithm halts, and a green signal is provided on the lane having a priority vehicle. After the priority vehicle's passage, the round-robin algorithm's operation is resumed.

*b) Optimizing green signal timing*: Green timing signal is provided based on three parameters: Inflow rate (number of vehicles passing through the junction in one cycle), waiting for vehicles at the current lane (number of vehicles), and density of the vehicles (number of vehicles) at the adjacent lane. The linguistic variables for the input values used are Very Low (0-30 vehicles), Low (20-50 vehicles), Medium (40-70 vehicles), High (60-90 vehicles), and Very High (80-110 vehicles). Depending upon the input variables and output variables, a total of 125 rules have been formulated, for example:

i. If the inflow rate is shallow, waiting vehicles are very low at the current lane, and density at the adjacent lane is very low, green timing extension is shallow.

ii. If the inflow rate is high, waiting vehicles are low at the current lane, and density at the adjacent lane is very low, green timing extension is shallow. If the inflow rate is low, waiting vehicles are high at the current lane, and density at the adjacent lane is low, green timing extension is high.

iii. If the inflow rate is medium, waiting vehicles are medium at the current lane, and density at the adjacent lane is medium, green timing extension is medium.

iv. If the inflow rate is high, waiting for vehicles at the current lane is very high, and density at the adjacent lane is low, green timing extension is medium.

*c) Training of system*: The efficient design of ANFIS-based models need practical parameter training for enhanced accuracy. Training data is the most influencing factor. Therefore, to train the proposed model, traffic patterns are analyzed by taking videos of a junction (Guru Nanak Mission Chowk) in Jalandhar, Punjab. A dataset consists of values for inflow rate, waiting for vehicles, and vehicle density. For training and updating the proposed system, traffic data for one month can be stored so that the system's rules will be updated periodically to make the system adaptive.

Pseudo code for the proposed methodology for adaptive traffic controller:

---

1. Initialize the current road side and next road side: Set CURR=Roadside1, NEXT=Roadside2
2. Initialize the algorithm parameters like the presence of priority vehicle (FLAG) and priority Road (Roadside_P).
3. Check priority vehicle is present.

   If FLAG==1:

CURR=Roadside1

If CURR==Roadside1 then

NEXT_P=Roadside2

Else if CURR== Roadside2, then

NEXT_P=Roadside3

Else if CURR==Roadside3, then

NEXT_P=Roadside4

Else

NEXT_P=Roadside1

End If

Initialize inflow rate (IRATE), number of waiting for vehicles at current lane (VCURR), and number of waiting for vehicles at adjacent lane (VADJ)

End if

4. Set green timing signals by evaluating the proposed system on IRATE, VCURR, and VADJ.

If VCURR==0 then

SIGNAL=0

Else

SIGNAL=15+EVAL(IRATE,VCURR,VADJ)

End if

5. Activate the green signal timing for the SIGNAL seconds duration.

6. Assign the current lane and next lane for the next cycle:

CURR=NEXT

If NEXT==Roadside1 then

NEXT=Roadside2

Else if NEXT==Roadside2, then

NEXT=Roadside3

Else if NEXT==Roadside3, then

NEXT=Roadside4

Else

NEXT=Roadside1

End if

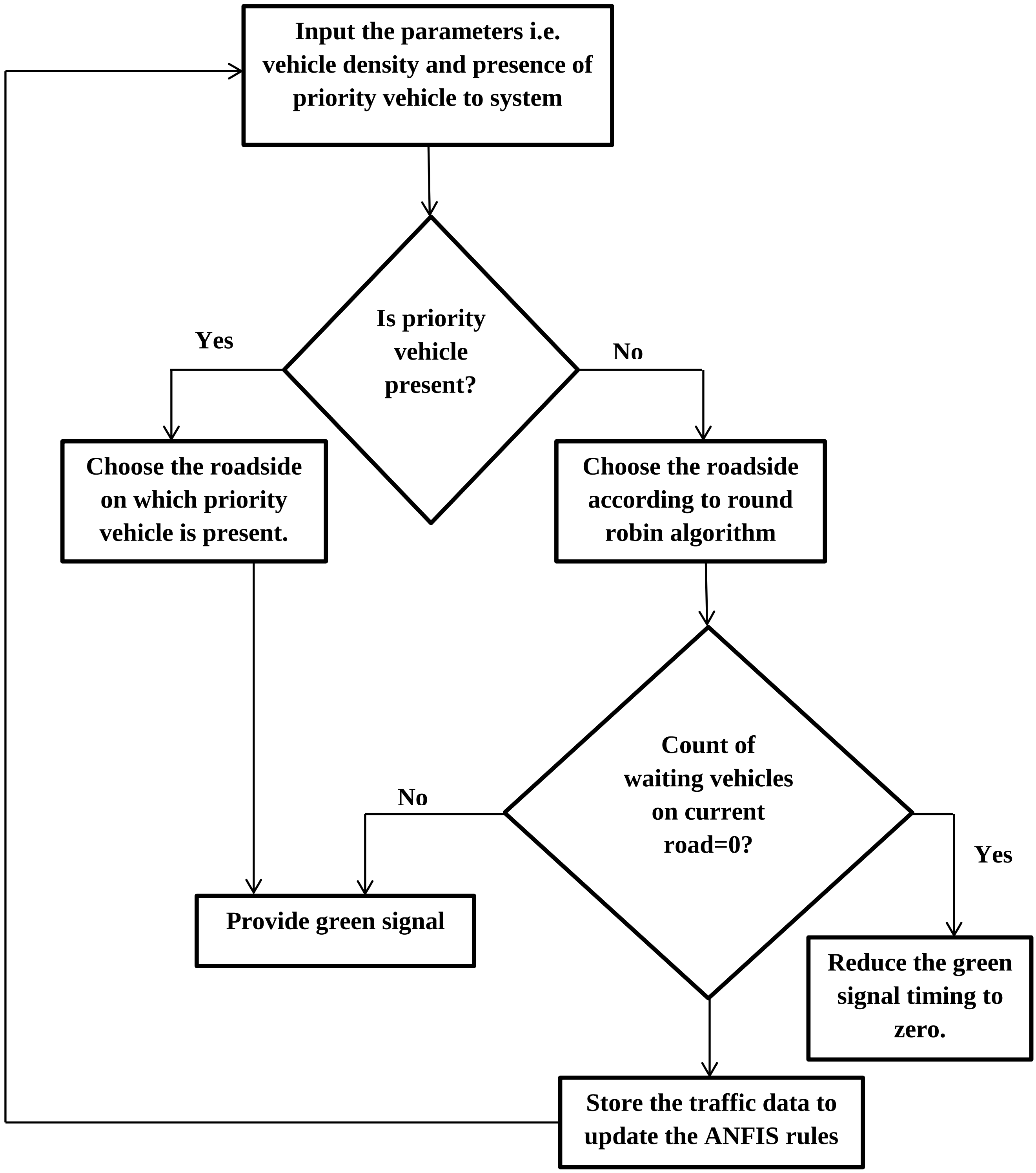



Figure 6.3: Flow chart of the proposed system [220]

## 6.3 Sending Data from One Intersection to Adjacent Intersection

There are broadly two options to deal with such critical data: Premise communication or Cloud services. WSN modules such as Xbee or Zigbee can be used to build a reliable connection of internodes around the traffic signals and other on-ground modules. Figure 6.4 and Figure 6.5 shows the receiver and sender circuit of WSN.

a) Motivation to select Xbee or Zigbee

i. IEEE 802.15.4-2003 standard designed for point-to-point and star communications
ii. A broad spectrum of frequencies to choose from for data transmission
iii. Relatively approachable for development and deployment as its long presence in solving WSN-related needs

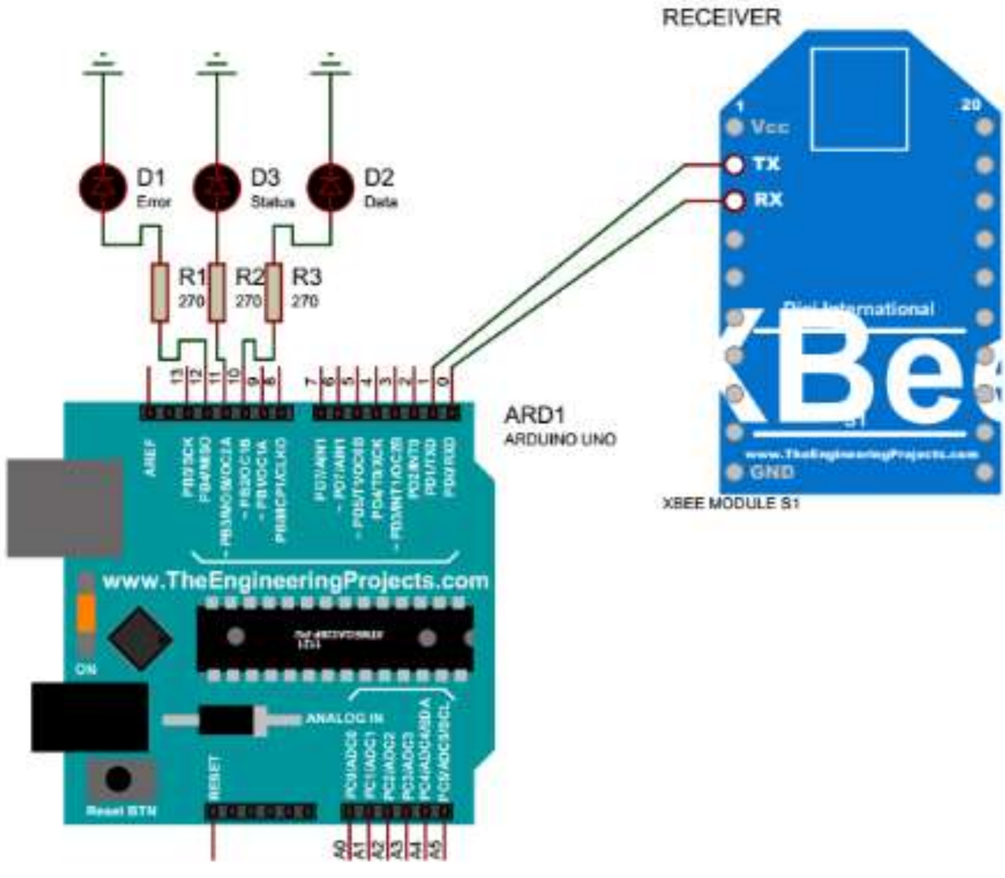



Figure 6.4: Receiver Circuit (WSN) [216]

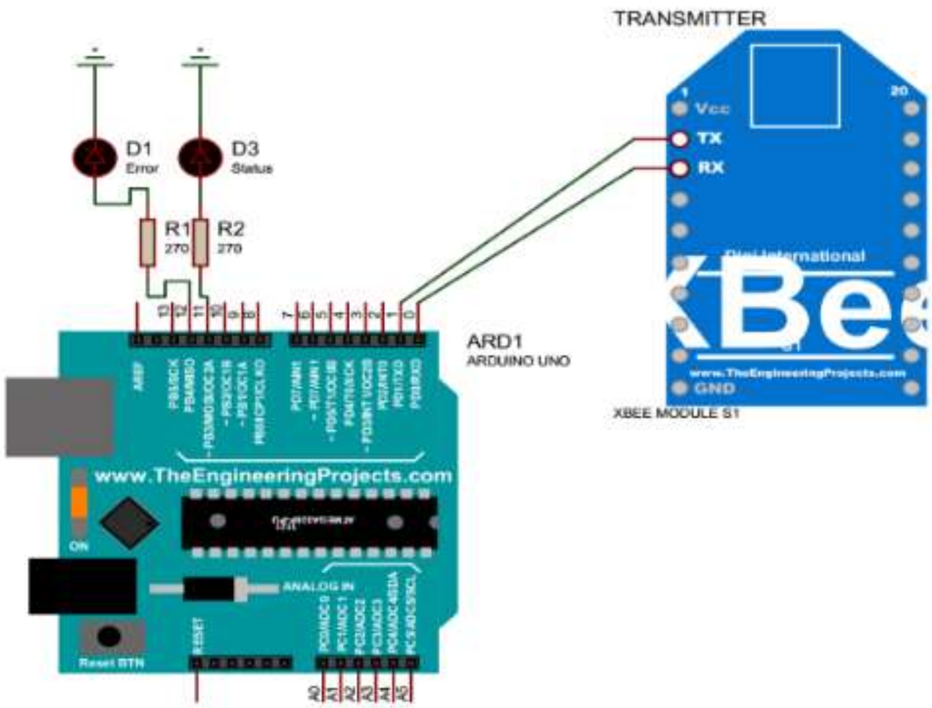



Figure 6.5: Sender Circuit (WSN) [216]

Even though with all these perks as well, Xbee fell short of meeting the requirement to serve as a viable option for data communication

b) Reasons to look for a better solution.

I) The Max range is limited to 3.2 Km, while Some traffic posts are farther away than this threshold.

II) Packet dropping is a well-known issue while working with radio-wave-based solutions.

III) Limited Baud rate to receive and process data which can bottleneck the whole workflow.

IV) It can become economically unfeasible to deploy in real-world scenarios.

The cloud-based solution seems a far more reliable alternative as it works around all the shortcomings of WSN in above mentioned case scenario. Even though there are many options, AWS S3 services fit the requirements perfectly. Figure 6.6 shows the dataflow of AWS S3.

I) Multiple servers across the globe solve threshold distance issues and open the avenue for further expansions

II) High bandwidth accessibility delivers real-time data storage and retrieval necessities

III) on-demand compute power, makes it cheaper to deploy in real-world situations

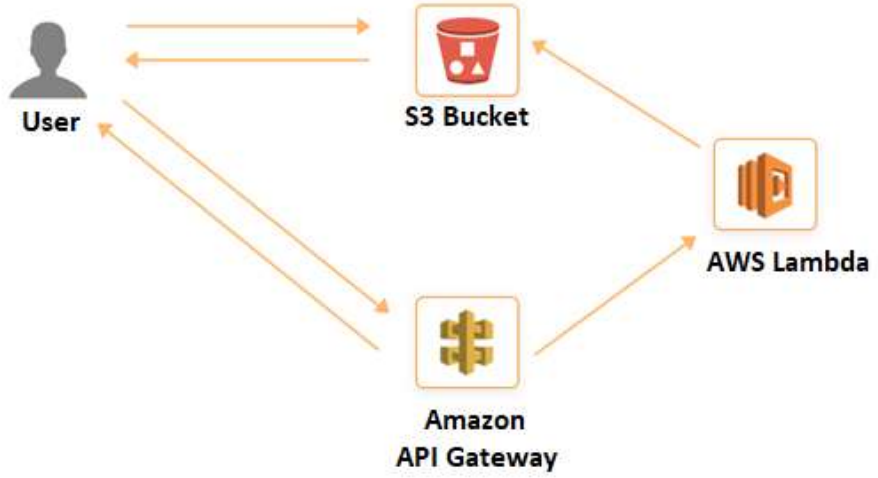



Figure 6.6 AWS S3 dataflow (Cloud) [217]

## Chapter Summary

Congestion management at the intersection is the primary step to solving traffic-related problems. Previous researches show that designing an adaptive traffic controller is the primary step that can work on real-time data. The research work presented in this paper studies the traffic pattern of an intersection at different time intervals of the day and models the system on actual input data using the soft computing methodology that is ANFIS. It was assumed that a large amount of training data is required to train ANFIS successfully. However, it may be challenging to get a massive amount of data. Hence, this research attempted to build the system using a small data pool.

# Chapter 7

# Results and Discussions

In this thesis, three models are implemented. One for vehicle detection and traffic density estimation, the second for emergency vehicle detection using RFID and siren sounds, third for optimization of green signals using the ANFIS model.

The experimental results of all the models have been discussed in this chapter.

## 7.1 Results of Vehicle Detection

Implementation of all three models has been done in Python using TensorFlow API. The experimental platform configuration of the machine is given in Table 7.1. The input size of the model is 300 X 300 with a batch size of 24. The learning rate and IoU threshold are 0.0002 and 0.6, respectively. Table 4 shows the difference between thermal and RGB images of the same scene taken from the FLIR dataset. As thermal images generate the images by sensing heat reflected by a body, the image quality is not affected by lighting and weather conditions. Thus, the visibility of objects is better in thermal images than in RGB.

To compare the SSD, Faster R-CNN, and proposed models, precision vs. recall curve and (Mean average precision) mAP matrices have been used. The precision-recall curve (PR Curve) helps report information retrieval results. A good PR curve has a greater area under the curve (AUC). The mAP helps measure the performance of object detection tasks. It compares the ground truth bounding box to the detected box. Figure 7.1 to Figure 7.4 shows the graphs of the precision-recall curve of the FLIR thermal dataset, FLIR RGB, MB7500, and KITTI dataset, respectively. All PR curves show that the proposed Ensemble performs better than SSD and Faster R-CNN models. Table 7.2 shows the computed mAP values on the different datasets for SSD, Faster R-CNN, and proposed Ensemble [221]. The maximum mAP achieved is 94% by the proposed Ensemble on FLIR thermal dataset that is 34% higher than SSD and 6% from the Faster R-CNN model. The presented ensemble model performs better and yields more acceptable results than individual estimators. Experimental results also show that thermal image detection is better than visible images. A comparison of different models on a collected dataset based on mAP is shown in Figure 7.5 [221].

Table 7.1 Experimental Platform Configuration

| **Computing Machine** | **Configuration** |
|---|---|
| Operating System | Windows 10 |
| GPU | NVIDIA GEFORCE GTX (4GB) |
| RAM | 8 GB |
| Processor | Intel Core i5 |
| GPU acceleration library | CUDA, CUDNN |

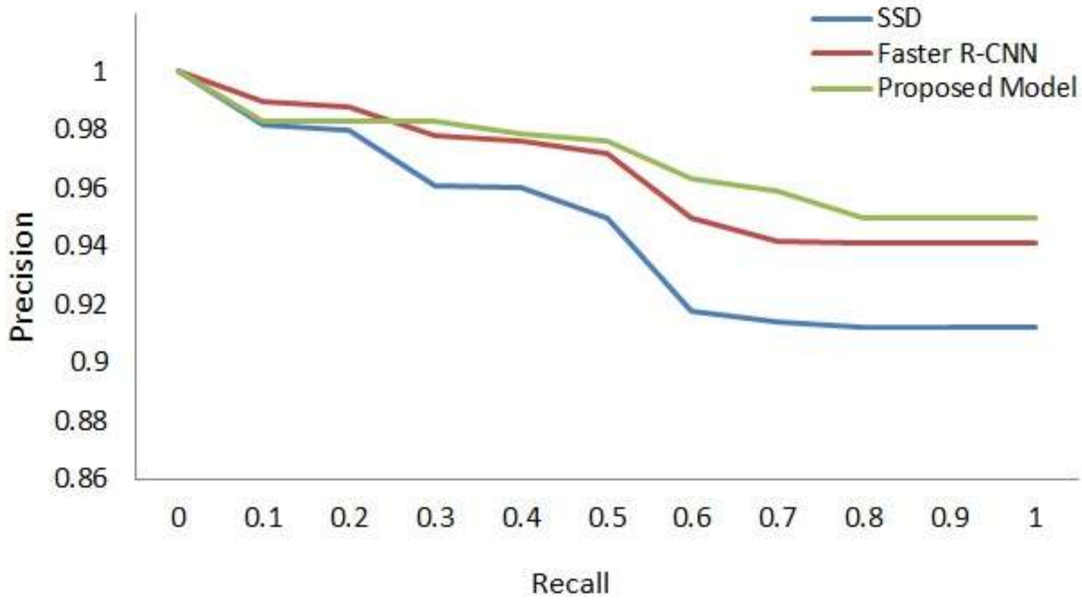



Figure 7.1: Precision vs. Recall Curve of SSD, Faster R-CNN, and Proposed model on FLIR Thermal dataset [221]

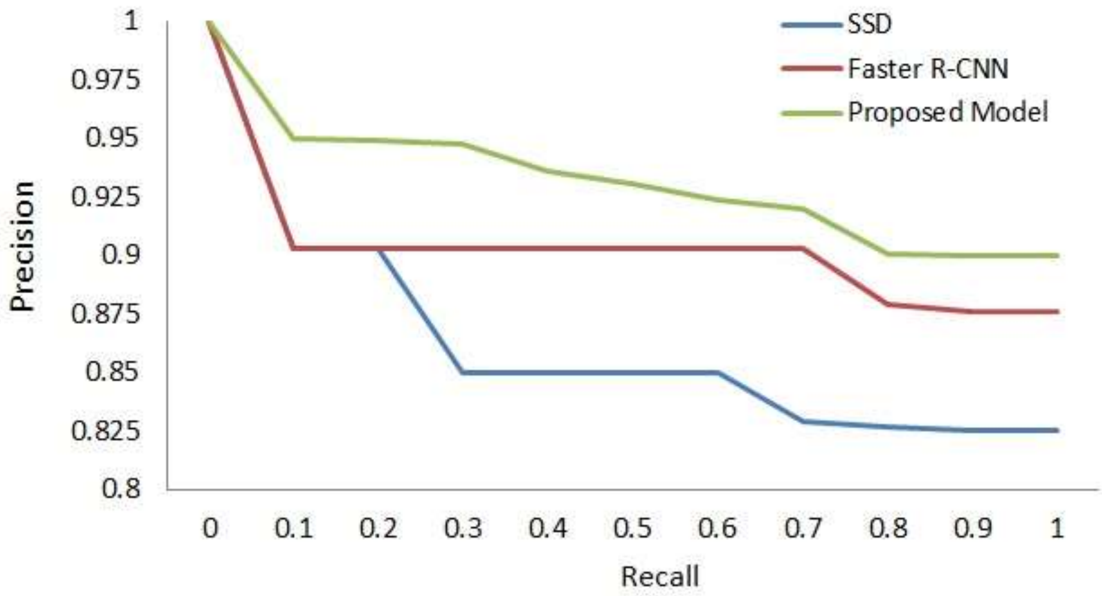



Figure 7.2 Precision vs. Recall Curve of SSD, Faster R-CNN, and Proposed model on FLIR RGB dataset [221]

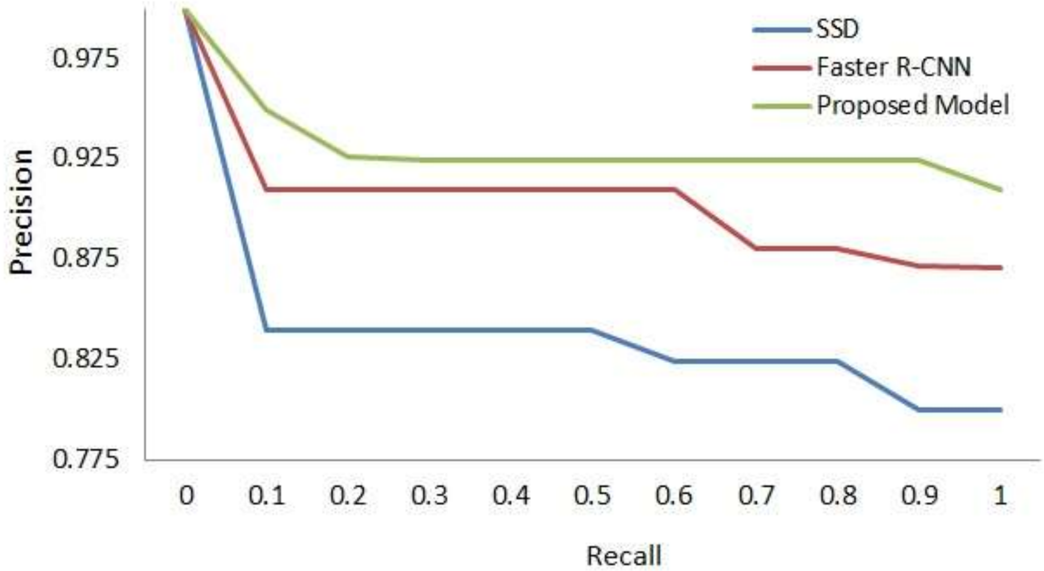



Figure 7.3 Precision vs. Recall Curve of SSD, Faster R-CNN, and Proposed model on MB7500 dataset [221]

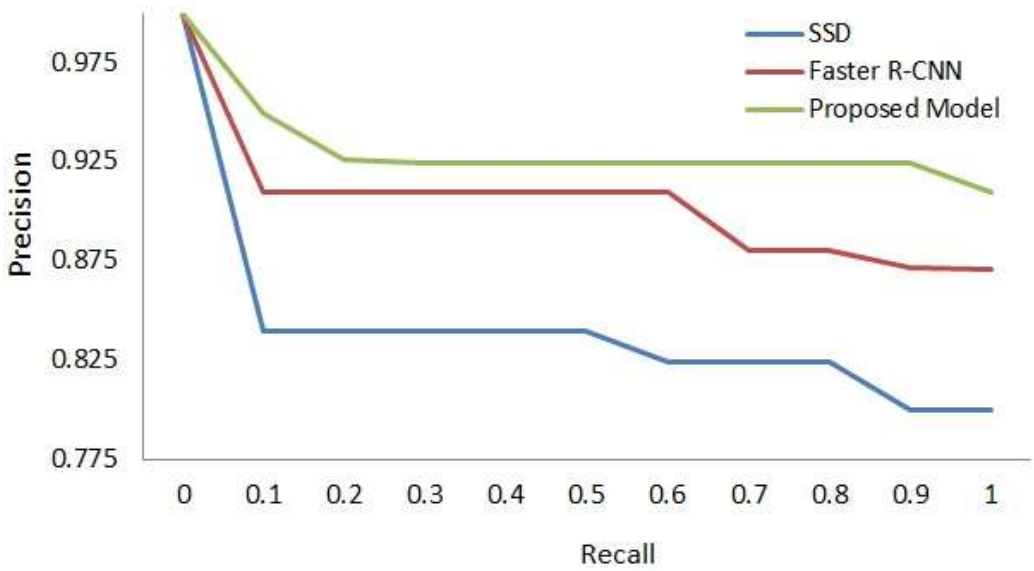



Figure 7.4 Precision vs. Recall Curve of SSD, Faster R-CNN, and Proposed model on the KITTI dataset [221]

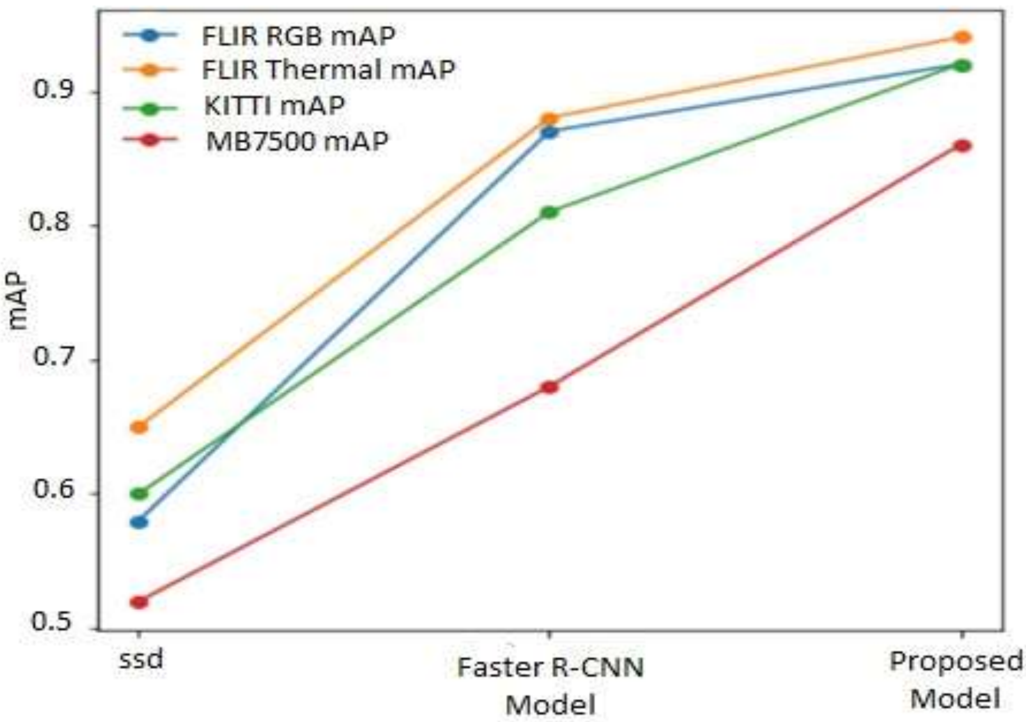



Figure 7.5 mAP of SSD, Faster R-CNN, and Proposed Ensemble on FLIR RGB, Thermal, KITTI, and MB7500 dataset [221]

Table 7.2: Comparative analysis of SSD, Faster R-CNN, and Proposed Ensemble based upon mAP on FLIR, KITTI, and MB7500

| **Model** | **Dataset** | **mAP** | **CY** | **TW** | **LV** | **HV** | **TR** | **BU** |
|---|---|---|---|---|---|---|---|---|
| SSD | FLIR RGB | 0.58 | 0.65 | 0.54 | 0.64 | 0.55 | 0.58 | 0.5 |
| | FLIR Thermal | 0.65 | 0.7 | 0.64 | 0.61 | 0.58 | 0.81 | 0.55 |
| | KITTI | 0.60 | 0.55 | 0.51 | 0.51 | 0.59 | 0.46 | 0.93 |
| | MB7500 | 0.52 | 0.52 | 0.47 | 0.59 | 0.41 | 0.57 | 0.56 |
| Faster R-CNN | FLIR RGB | 0.87 | 0.94 | 0.87 | 0.8 | 0.94 | 1 | 0.67 |
| | FLIR Thermal | 0.88 | 0.96 | 0.91 | 0.86 | 0.87 | 0.86 | 0.8 |
| | KITTI | 0.81 | 0.75 | 0.75 | 0.84 | 0.89 | 0.77 | 0.85 |
| | MB7500 | 0.68 | 0.68 | 0.6 | 0.59 | 0.94 | 0.6 | 0.67 |
| Proposed Ensemble | FLIR RGB | 0.92 | 0.94 | 0.93 | 0.91 | 0.91 | 1 | 0.83 |
| | FLIR Thermal | 0.94 | 0.95 | 0.96 | 0.89 | 0.97 | 1 | 0.9 |
| | KITTI | 0.92 | 0.85 | 0.87 | 0.96 | 0.92 | 0.97 | 0.98 |
| | MB7500 | 0.86 | 0.89 | 0.8 | 0.96 | 0.94 | 0.78 | 0.78 |

Table 7.3 shows the comparison of the proposed work with existing studies. Most of the work has been done for vehicle detection using visible images [221]. In this study, an ensemble of deep neural networks has been proposed and evaluated on thermal and visible images. The proposed model shows promising results on both types of images.

Table 7.3: Comparison of the proposed model with existing methods

| **Technique/Reference** | **Image Type** | **Accuracy** |
|---|---|---|
| Nam’s Approach [77] | Visible Images | 92.7 |
| Nam’s Approach [77] | Thermal Images | 65.8 |
| CNN-based Ensemble [78] | Visible Images | 93.2 |
| ShuffleDet [85] | Visible Images | 62.89 |

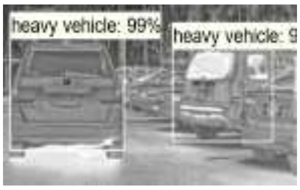
| | | |
|---|---|---|
| ECNN-SVM [89] | Visible Images | 93.63 |
| LittleYOLO-SPP [98] | Visible Images | 77.44 |
| Proposed Model | Visible Images | 92 |
| Proposed Model | Thermal Images | 94 |

## 7.2 Results of Density Estimation

Figure 7.6 to Figure 7.13 shows some of the experimental results obtained using SSD, Faster R-CNN, and the proposed model on the collected datasets [221]. An analysis is made based on the density estimated by each model. Computed density can be utilized for the designing of traffic light controllers.


(a)

Actual Density=7.6 units
Predicted Density= 3.8 units




(b)

Actual Density=7.6 units
Predicted Density= 4.2 units

(c)

Actual Density=7.6 units
Predicted Density =4.2 units

Figure 7.6 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t FLIR thermal dataset [221]

(a)

Actual Density=9.9 units
Predicted Density= 4.0 units

(b)

Actual Density=9.9 units
Predicted Density= 4.4 units

(c)

Actual Density=9.9 units
Predicted Density =6.1 units

Figure 7.7 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t FLIR thermal dataset [221]

Figure 7.6 consists of two heavy vehicles and two light vehicles. Figure 7.6 (a) depicts the results of the SSD model in which the exact vehicle is predicted as a heavy vehicle, and a light vehicle means duplicate detection, and SSD fails to detect other vehicles. While Figures 7.6 (b) and 7.6 (c) show the results of Faster R-CNN and the proposed method, respectively, in which two heavy vehicles are detected accurately, and models fail to detect light vehicles. Hence, the actual density in the image is 7.6 (2 heavy vehicles=4.5 units and two light vehicles=3.4 units). Vehicle density computed by SSD models is 3.8 units, and Faster R-CNN and proposed model computes 4.2 units, respectively. SSD models also predict one object as light and heavy vehicles. Figure 7.7 contains one truck, two light vehicles, and two heavy vehicles, and the total estimated vehicle density is 9.9 units. Figure 7.7 (a) depicts the results of the SSD model in which one truck and one light vehicle are detected, and the computed vehicle density is 4.0 units. Figure 7.7 (b) shows the results of Faster R-CNN in which one truck and one heavy vehicle are detected, and vehicle density is 4.4 units. Figure 7.7 (c) contains the predictions made by proposing method. Vehicles are predicted as one truck, a light vehicle, and a heavy vehicle. The computed vehicle density is 6.6 units, the closest value to the actual value among all models.

(a)
Actual Density=4.0 units
Predicted Density=4.0 units

(b)
Actual Density=4.0 units
Predicted Density=1.7 units

(c)
Actual Density=4.0 units
Predicted Density =4.0 units

Figure 7.8: Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t FLIR RGB dataset [221]

(a) Actual Density=5.5 units Predicted Density= 1.7 units

(b) Actual Density=5.5 units Predicted Density=5.5 units

(c) Actual Density=5.5 units Predicted Density =5.5 units

Figure 7.9 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t FLIR RGB dataset [221]

Figure 7.8 shows two vehicles in the image: a light vehicle and a truck. Thus, the total vehicle density is 4.0 units. Figure 7.8 (a) contains detections made by the SSD model, where it detects both the vehicles and label them correctly as light vehicle and truck. Thus, the computed vehicle density is 4.0 units. In comparison, Faster R-CNN predicts only light vehicles and misses a truck. So, computer density is only 1.7 units. Predictions made by the proposed method are correct, and the estimated density is 4.0 units. Figure 7.9 consists of two light vehicles and one heavy vehicle. Therefore, the actual vehicle density is 14 units. Figure 7.9 (a) depicts the results of SSD models showing only one vehicle as a light vehicle and vehicle density as 1.7 units. While Figures 7.9 (b) and 7.9 (c) show the results of Faster R-CNN and the proposed method, respectively, in which all vehicles are predicted accurately, and the computed vehicle density is 5.5 units.


(a) Actual Density=9.3 units Predicted Density=4.0 units

(b) Actual Density=9.3 units Predicted Density=1.0 units

(c) Actual Density=9.3units Predicted Density=5.0 units

Figure 7.10 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t MB7500 dataset [221]


(a)

Actual Density=6.6 units

Predicted Density= 1.7 units


(b)

Actual Density=6.6 units

Predicted Density=5.3 units

(c)

Actual Density=6.6 units

Predicted Density=5.3 units

Figure 7.11 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t MB7500 dataset [221]

Figure 7.10 has five vehicles one truck, one bus, two light vehicles, and one 2-wheeler. Thus, the actual density is 9.3 units. Results of the SSD model are shown in figure 7.10 (a), and SSD predicts one bus correctly. A truck is detected as a light vehicle. Thus, the total predicted density is 4.0. Faster R-CNN predicted a 2-wheeler as one cycle, and the calculated density is only 1 unit. Finally, the proposed model predicts three vehicles and predicted 5.0 units. Figure 7.11 has four vehicles: one light vehicle, one bus, and two 2-wheelers. Thus, the actual density is 6.6 units. Results of the SSD model are shown in Figure 7.11 (a), and SSD predicts only one light vehicle, and the estimated vehicle density is 1.7 units. Where predictions made by Faster R-CNN and the proposed method are given in Figures 7.11 (b) and 7.11 (c), respectively, these models detect one light vehicle, one bus, and one 2-wheeler as predicted as a cycle. Therefore, the estimated vehicle density is 5.3 units for both models.

(a)

Actual Density=3.1 units

Predicted Density= 0 units

(b)

Actual Density=3.1 units

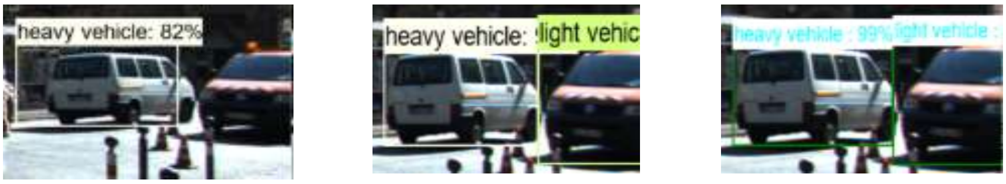
Predicted Density=3.1 units

(c)

Actual Density=3.1 units

Predicted Density =3.1 units

Figure 7.12 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t KITTI dataset [221]




(a)
Actual Density=3.8 units
Predicted Density=2.1 units

(b)
Actual Density=3.8 units
Predicted Density=3.8 units

(c)
Actual Density=3.8 units
Predicted Density =3.8 units

Figure 7.13 Actual and predicted density of (a) SSD model, (b) Faster R-CNN, and (c) Proposed Ensemble w.r.t KITTI dataset [221]

Figure 7.12 consists of one cycle and one heavy vehicle total of 3.1 units of density. SSD has not predicted any vehicle. Thus, the estimated density is 0 units. While Figures 7.12 (b) and 7.12 (c) show the results of Faster R-CNN and the proposed model, respectively, in which cycle and heavy vehicle are predicted correctly, and the computed density is 3.1 units. Figure 7.13 contains one heavy vehicle and one light vehicle. Thus, the actual vehicle density is 3.8 units. Figure 7.13 (a) shows the SSD model results in only one heavy vehicle being detected, and the computed vehicle density is 2.1 units. In comparison, Figures 7.13 (b) and 7.13 (c) show the results of Faster R-CNN and the proposed model, respectively, in which both the vehicle detected and computed density is the same as the actual density.

Other than the detection accuracy, the effectiveness of the detection model can also be estimated using inference speed. Table 7.4 shows the inference speed of different models SSD, Faster R-CNN, and the proposed method [221].

Table 7.4 Comparison of inference speed of Faster R-CNN, SSD, and Proposed Method

| **Model** | **Inference Speed** |
|---|---|
| Faster R-CNN | 15.67 Sec |
| SSD | 10.15 Sec |
| Ensemble | 22.37 Sec |

Considering the detection results and inference speed together, it is concluded that the proposed model provides better detection and density calculation results. However, the time taken to process the image is high compared to other models.

## 7.3 Results of Emergency Vehicle Detection using RFID

This method dramatically impacts the traffic problems faced in urban areas. Usually, the traffic is controlled by a predetermined signal light-controlled system. The working of the modules is based on the RFID tag information. Also, the license invalidation module is present to ensure the drivers obey the rules. Otherwise, a fine will be deducted from the account specified in the RFID tag. The only requirement of this paper to be implemented is to provide the vehicles with a unique RFID tag. The proposed method was executed five times, and the results are shown in Table 7.5 [219]:

Table 7.5: Results of Proposed Method for RFID-based Emergency Vehicle Detection

| **Traffic Light Switching Direction** | **Emergency Vehicle Present?** | **Number of vehicles present** | **Direction and details of vehicles from where vehicles coming** | **Signal to be provided** |
|---|---|---|---|---|
| North | Yes | 2 | From the west, Firebrigade (EPC: 1017)<br>From North, Police van (EPC: 1020) | Signal to be provided on West |
| West | Yes | 1 | From North, Police van (EPC: 1020) | Signal to be provided on North |
| North | No | NA | NA | Signal to be provided on South |
| South | No | NA | NA | Signal to be provided on East |
| East | Yes | 2 | From North, Police van (EPC: 1016)<br>From South, Ambulance (EPC: 1018) | Signal to be provided on East. |

Any RFID label on a path for over 20 minutes will distinguish an anomaly. It might be an accident in the vehicle or a mishap on the road. The vehicle with this EPC number will be tried for issues utilizing the phone number of the proprietor and, where pertinent, will give prompt help.

## 7.4 Results of Acoustic-Based Emergency Vehicle Detection

A comparative analysis of the proposed model with different deep learning architectures is given in Table 7.6. In this paper, four different models have been explored. FC_Net model, which consists of only dense layers, provides an accuracy of 96.4% and its inference time is 0.061s. CNN_Net model provides an accuracy of 92.4%, while RNN_Net and Ensemble have an accuracy of 94.5% and 98.7%, respectively. A comparison based upon inference time is also given in table 4, which clearly shows that the time taken by RNN_Net and FCNet is almost the same, while CNN takes longer than these models, and the response time of Ensemble is the highest, and its takes almost 1.5 seconds [218].

Table 7.6: Comparative Analysis of Different Models

| Model | Accuracy | Inference Time (s) |
|---|---|---|
| CNet | 96.4 | 0.061 |
| CNN_Net | 92.4 | 0.151 |
| RNN_Net | 94.5 | 0.061 |
| Ensemble | 98.7 | 1.5 |

Various machine learning models are also evaluated on the collected dataset, and results are compared with the deep learning models. Table 7.7 compares different machine learning models with proposed deep learning models [218]. Although decision trees and random forests provide higher training accuracy, their testing accuracy could be higher, and models are over-fitted. Compared to machine learning models, proposed deep learning models provide better accuracy and acceptable models.

Table 7.7: Comparative Analysis of deep learning models with machine learning models

| Method | Training Accuracy (%) | Testing Accuracy (%) |
|---|---|---|
| Perceptron (L1 Regularization) | 59.5 | 49.2 |
| Logistic Regression (L2 Regularization) | 65.8 | 44.3 |
| SVM (Kernel=Polynomial) | 61-7 | 52.4 |
| KNN (neighbors=10) | 61.7 | 46 |
| Decision Tree (Entropy) | 100 | 57.3 |

| | | |
|---|---|---|
| Random Forest (estimators=12) | 98 | 57.3 |
| Naïve Bayes | 65 | 42.6 |
| AdaBoost Classifier (base=Naïve Bayes, number of estimators=11) | 51 | 51 |
| Fully Connected Neural Network (8 FC layers) | 100 | 96.4 |
| Convolutional Neural Network (3 conv layers) | 99.58 | 92.4 |
| Recurrent Neural Network (5 LSTM layers) (Proposed) | 98.7 | 94.5 |
| Ensemble Model (Proposed) | 99.74 | 98.7 |

Several kinds of research based on microcontrollers [112] and circuit design [114] only stated the prospects of the siren detection system without evaluating its accuracy on a large dataset. As a result, works focused on ML and DL methods have been considered for comparison. Table 7.8 compares our proposed models to previous findings [212, 213, 214, 215, and 126] in terms of methods and functionality, and prediction accuracy. Table 7.8 depicts the classification accuracy of the CNN model proposed by L. Marchegiani [212], and the proposed model with RNN is almost the same. Machine learning models like KNN [212], HMM [214], and part-based models [214] had an accuracy of less than 90%. CNN models developed by Tran [128] achieved an accuracy of 98.24%. The proposed ensemble model provides 98.7% accuracy, the highest among all works. The proposed Ensemble yields promising results 98.7%, which is better than the results of RNN_Net (94.5%) and other related works [215] 94%, [213] 83%, [211] 85%, [214], 86%, [215] 98.24%. The performance of the ensemble model are more promising than other models.

Table 7.8: Comparative Analysis based on the accuracy of existing works

| **Work** | **Features** | **Approach/Model** | **Classification _Accuracy** |
|---|---|---|---|
| L. Marchegiani et al. [215] 2018 | Spectrogram (log-mel) | CNN | 94.00 |
| L. Marchegiani et al. [213] | Spectrogram | K-NN | 83.00 |

| | | | |
|---|---|---|---|
| 2017 | | | |
| J. J Liaw et al. [212] 2013 | Longest Common Subsequence (LCS) | LCS Comparison | 85.00 |
| J. Schroder et al. [214] 2013 | Hand-labeled<br>PBMs MFCC Spectrogram | Part-based Models (PBMs)<br>HMM | 86.00<br>(PBMs)<br>80.00<br>(HMM+MFCC)<br>74.00<br>(HMM+log-mel |
| V.T Tran et al. [127] 2020 | Raw-data<br>MFCC + Spectrogram<br>Aggregated features: Raw data, Spectrogram, MFCC | 1D-CNN (WaveNet)<br>2D- CNN<br>(MLNet)<br>CNN(SirenNet) | 96.51<br>(1D-CNN)<br>96.42<br>(2D-CNN)<br>98.24<br>(CNN) |
| Proposed work with RNN | MFCC | RNN | 94.5 |
| Proposed work with Ensemble | MFCC | Ensemble (FCNet, CNN_Net, RNN_Net) | 98.7 |

## 7.5 Results of Green Signal Optimization using Adaptive Neuro-Fuzzy Inference System

The simulation was done using MATLAB. Fuzzy Logic Toolbox provides the steps of designing a fuzzy inference system and functions for adaptive neuro-fuzzy learning. The developed ANFIS model is applied to solve the problem of congestion. The real-time data has been taken for modeling the proposed system. To generate the FIS, the grid partition method is used. This method partitions the input space into several fuzzy regions to form the antecedents of the fuzzy rules. To train the model, a hybrid learning algorithm is used. The hybrid learning algorithm works in two phases: the forward phase to compute the results and the backward phase to update the parameters to minimize the error. The performance of the ANFIS model is compared with the fuzzy and fixed timer-based systems. Table 7.9 shows different cases of the predicted output of ANFIS, fuzzy system, and fixed timer system [220]. Case 1 shows that if vehicles at the current lane are very low, then the green timing signal is also low, and if density at the current lane is very high, then green timing is high, say 93 seconds, as given in case 2. Case 3 indicates that the proposed model provides priority to emergency vehicles, and case 4 presents that no signal is provided at the current lane if waiting vehicles are zero at the current lane.

Table 7.9: Fixed timer system, fuzzy system, and ANFIS predicted results

| Case | Inflow Rate 80(*irate*) | Number of waiting for vehicles at current lane ($V_{curr}$) | Number of waiting for vehicles at adjacent lane ($V_{adj}$) | Priority vehicle present (*flag*) | Lane on which priority vehicle present | Fixed timer system output (in seconds) | Fuzzy system output (in seconds) | ANFIS output (in seconds) | Lane to be served by a fixed timer system | Lane to be served by ANFIS |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 18 | 7 | 40 | 0 | - | 60 | 10 | 3 | 1 | 1 |
| 2 | 89 | 118 | 49 | 0 | - | 60 | 55 | 93 | 2 | 2 |
| 3 | 102 | 82 | 44 | 1 | 2 | 60 | 55 | 42 | 4 | 2 |
| 4 | 25 | 0 | 17 | 0 | - | 60 | 18 | 0 | 2 | 3 |

Figure 7.14 and Figure 7.15 compare expected green signal timing vs. obtained green signal timing by the ANFIS and FIS models. ANFIS model has higher accuracy as compared to FIS.

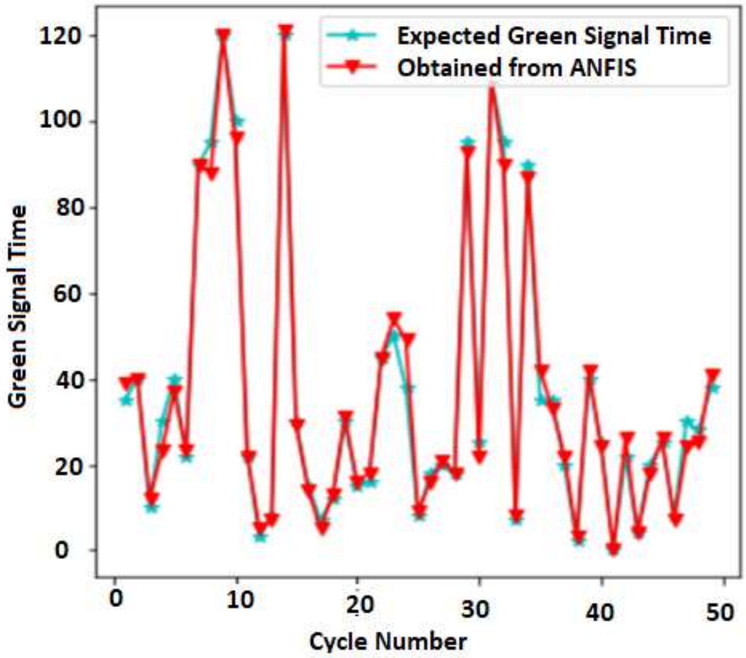



Figure 7.14 Expected green signal timing vs. obtained green signal by ANFIS [220]

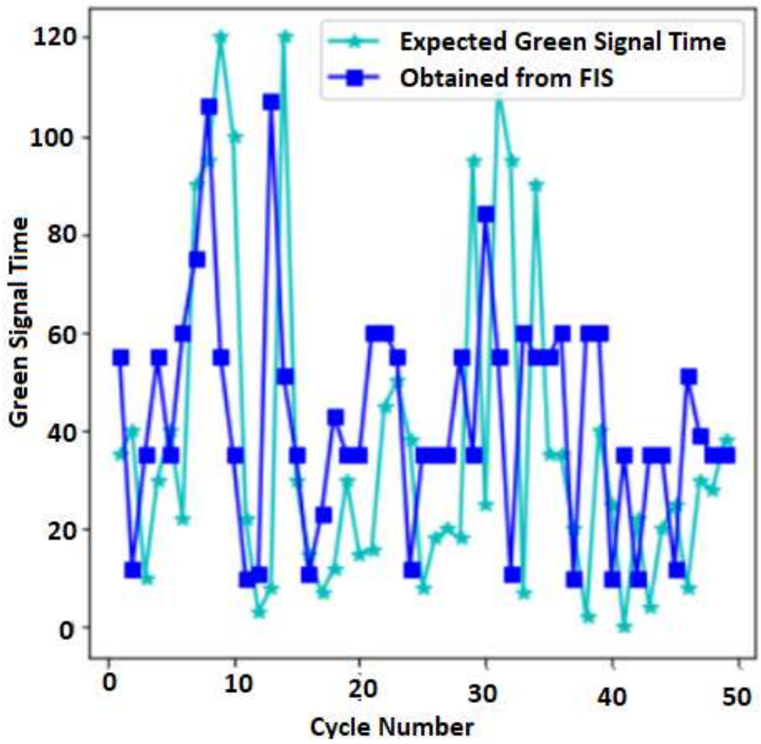



Figure 7.15 Expected green signal timing vs. obtained green signal by FIS [220]

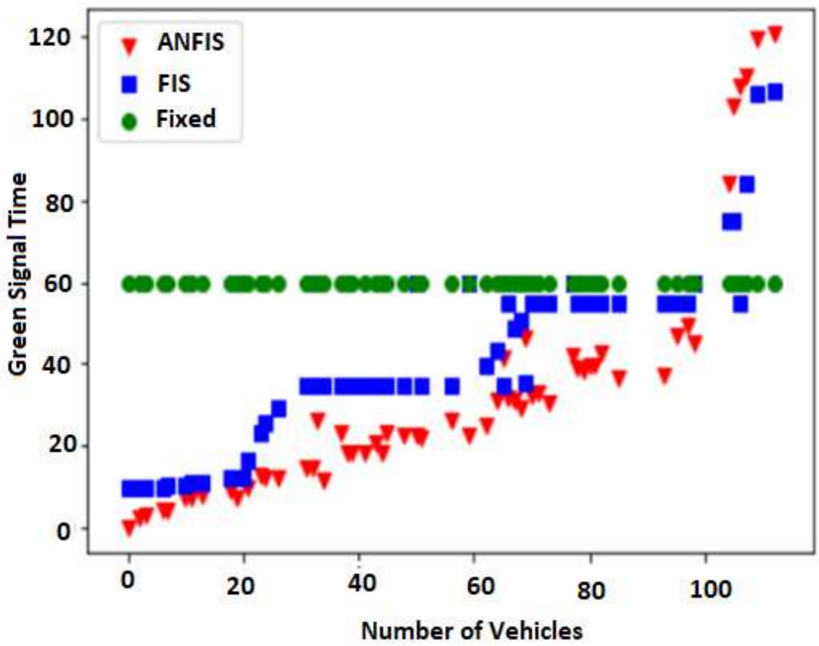



Figure 7.16: Number of vehicles vs. green signal timing of ANFIS, FIS, and Fixed Timer System [220]

Figure 7.16 compares the ANFIS model with the Mamdani-based FIS and fixed timer-based system based on the number of vehicles vs. green signal timing. From the comparison, it can be seen that green signal timing is less if the density at the current lane is less than 100. In the case of the ANFIS system resulting in less waiting time for the vehicles at the adjacent lane, signal timing is constant in a fixed timer-based system leading to higher waiting times. From the test data, it has been observed that the total number of vehicles passing through the intersection is 3147, and the time taken to pass these vehicles by ANFIS model, FIS model, and fixed timer based model is 1866 seconds, 2411 seconds, and 3480 seconds respectively. Thus, the overall waiting time is reduced.

Figure 7.17, Figure 7.18, and Figure 7.19 show the entire cycle length duration of the ANFIS model, Mamdani FIS, and Fixed timer system [220]. ANFIS model has the shortest duration, while the fixed timer system has the maximum duration.

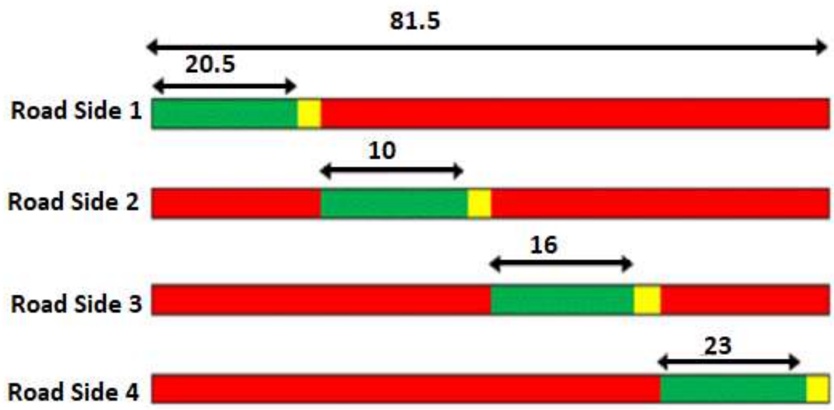



Figure 7.17 ANFIS signal timing in the first cycle [220]

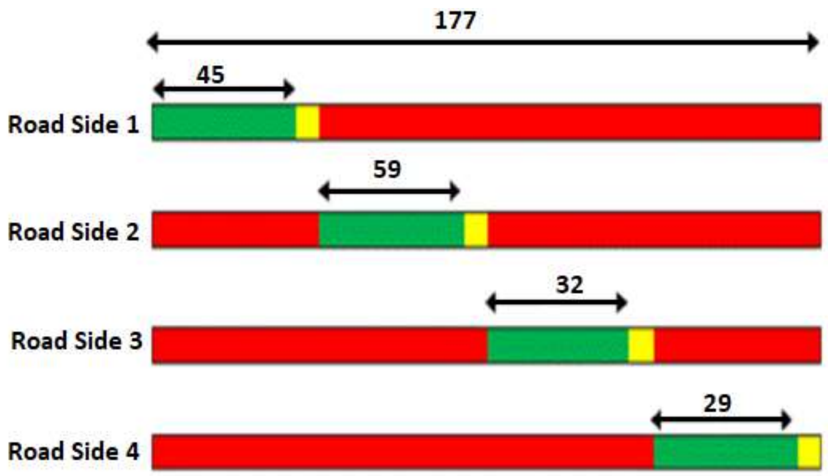



Figure 7.18 Mamdani FIS signal timing in the first cycle [220]

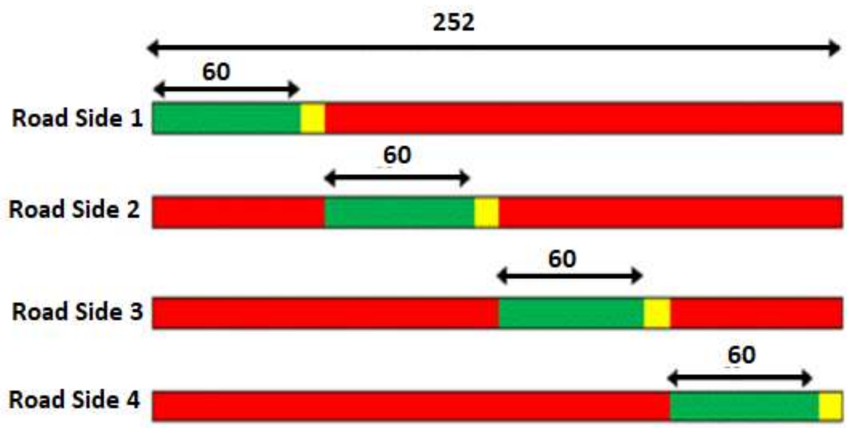



Figure 7.19 Fixed timer signal timing in the first cycle [220]

## Chapter Summary

In this chapter, the results of all the models have been discussed. From the results of vehicle detection and classification, it has been concluded that the proposed Ensemble provides the highest accuracy compared to its base estimators and also provides traffic density available on the road at a particular time. Emergency vehicles have been detected by using RFID and sound signals. RFID tags have less wavelength so vehicles can be identified at a shorter distance from the intersection. Acoustic-based models have been implemented using CNN, RNN, and fully connected neural Networks. An ensemble of three configurations is also designed. It has been concluded that the proposed model provides the highest accuracy. Finally, Green signal optimization has been done using the ANFIS model, and a comparative analysis is provided with a fixed timer system, fuzzy-based system, and proposed model. Two methods have been utilized to send traffic information from one intersection to an adjacent intersection. WSN is beneficial when the distance between two intersections is less. But if the distance between two intersections is high, then cloud-based services provide better results. AWS S3 service has been utilized to send traffic information over high distances.

# Chapter 8

# Conclusion and Future Scope

In this work, an artificial intelligence-based traffic light controller has been implemented, which detects and estimates traffic density and makes detections for the presence of emergency vehicles. Green signal optimization has been done using an adaptive neuro-fuzzy inference system based on vehicle density and the presence of an emergency vehicle.

To detect and classify the vehicles, Dataset has been collected from open source libraries. Two deep learning architectures, Faster R-CNN and SSD models are trained on the collected Dataset. Then, a hybrid model based on ensemble learning was implemented. From the experimental results, it has been concluded that the proposed model performs better in terms of accuracy than its base estimators, and traffic density estimation is done. Results of the density estimation are also better yielded by the proposed model as compared to Faster R-CNN and SSD.

To detect the emergency vehicle on the road near the intersection has been identified using RFID and siren sounds. RFID has two parts: an RFID tag and an RFID reader. But RFID technology has some limitations, like more distant objects can not be identified by using them. So, acoustic-based emergency vehicle detection has also been implemented. , siren sounds of emergency vehicles have been collected from the open source library. To perform sound-based detection, perform sound-based detection. Then, three deep learning models that are fully connected neural networks, CNN and RNN, were implemented, and their results were analyzed. Finally, an ensemble based on the three configurations discussed above has been implemented. From the experimental results, it has been concluded that the proposed ensemble performs better than individual models. RNN also provides acceptable results but has less performance than the ensemble model.

Finally, green signal optimization has been done using an adaptive neuro-fuzzy inference system based on vehicle density calculation and the presence or absence of emergency vehicles. In this system, density at the current lane, adjacent lane, and intersection flow rate has been considered input parameters. The green signal to be provided on a particular phase is determined. The proposed model’s results have been compared with a fixed timer and fuzzy-based controller. It is concluded that the proposed model reduces the vehicles' overall waiting time.

There are some limitations of the research work. As for vehicle detection, an ensemble technique is proposed, which takes longer detection time compared to base estimators. Integration of modules like vehicle detection, emergency vehicle detection, and signal optimization is also lacking.

The proposed work will be extended for futuristic work to improve the system's performance. The length and width of the lane can also be considered for vehicle detection and traffic density estimation. Some mechanisms will be introduced in acoustic-based emergency vehicle detection to eliminate environmental noise. Likewise, green signal optimization can be done using reinforcement learning.

## List of Publications

[1.] Usha Mittal, and Priyanka Chawla. "Vehicle detection and traffic density estimation using ensemble of deep learning models." Multimedia Tools and Applications (2022): 1-23. (Published)

[2.] Usha Mittal, Priyanka Chawla, and Rajeev Tiwari. "EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models." Neural Computing and Applications (2022): 1-20. (Published)

[3.] Usha Mittal, Sonal Srivastava, and Priyanka Chawla. "Object detection and classification from thermal images using region based convolutional neural network." Journal of Computer Science 15, no. 7 (2019): 961-971. (Published)

[4.] Mittal Usha, and Priyanka Chawla. "Acoustic Based Emergency Vehicle Detection Using Ensemble of deep Learning Models." Procedia Computer Science 218 (2023): 227-234. (Published)

[5.] Usha Mittal, and Priyanka Chawla. "Detection of Emergency Vehicles Using Radio Frequency Identification (RFID)." In International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi Petronas, Malaysia, pp. 271-278. Singapore: Springer Nature Singapore, 2022. (Published)

[6.] Usha Mittal, and Priyanka Chawla. "Neuro–Fuzzy Based Adaptive Traffic Light Management system." In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 396-402. IEEE, 2020. (Published)

[7.] Usha Mittal, and Priyanka Chawla. "Traffic Green Signal Optimization using Adaptive Neuro-Fuzzy Inference System on Coordinated Intersections." In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1-7. IEEE, 2022. (Published)

## Bibliography


1 G. H. Kulkarni and P. G. Waingankar, “Fuzzy logic based traffic light controller,” in 2007 International Conference on Industrial and Information Systems, Aug. 2007, pp. 107–110. doi: 10.1109/ICIINFS.2007.4579157.

2 “Road Network in India: National Highways, Projects, Govt Initiatives | IBEF,” www.ibef.org, Dec. 2022. https://www.ibef.org/industry/roads-india (accessed Jan. 23, 2023).

3 S. Kul, S. Eken, and A. Sayar, “A concise review on vehicle detection and classification,” in 2017 International Conference on Engineering and Technology (ICET), Aug. 2017, pp. 1–4. doi: 10.1109/ICEngTechnol.2017.8308199.

4 S. Samadi, A. P. Rad, F. M. Kazemi, and H. Jafarian, “Performance Evaluation of Intelligent Adaptive Traffic Control Systems: A Case Study,” Journal of Transportation Technologies, vol. 02, no. 03, pp. 248–259, 2012, doi: 10.4236/jtts.2012.23027.

5 A. Boukerche, Y. Tao, and P. Sun, “Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems,” Computer Networks, vol. 182, p. 107484, Dec. 2020, doi: 10.1016/j.comnet.2020.107484.

6 V. Jain, A. Sharma, and L. Subramanian, “Road traffic congestion in the developing world,” in 2nd ACM Symposium on Computing for Development, 2012. doi: 10.1145/2160601.2160616.

7 TomTom, “Traffic congestion ranking | TomTom Traffic Index,” Tomtom.com, 2020. https://www.tomtom.com/en_gb/traffic-index/ranking/ (accessed Jul. 23, 2022).

8 Md. A. Iqbal, A. Zahin, Z. S. Islam, and M. S. Kaiser, “Neuro-Fuzzy based adaptive traffic flow control system,” in 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS), Dec. 2012, pp. 349–352. doi: 10.1109/CODIS.2012.6422210.

9 K. Kundalia, Y. Patel, and M. Shah, “Multi-label Movie Genre Detection from a Movie Poster Using Knowledge Transfer Learning,” Augmented Human Research, vol. 5, no. 1, Dec. 2019, doi: 10.1007/s41133-019-0029-y.

10 K. Jani, M. Chaudhuri, H. Patel, and M. Shah, “Machine learning in films: an approach towards automation in film censoring,” Journal of Data, Information and Management, vol. 2, no. 1, pp. 55–64, Dec. 2019, doi: 10.1007/s42488-019-00016-9.

11 K. Jha, A. Doshi, P. Patel, and M. Shah, “A comprehensive review on automation in agriculture using artificial intelligence,” Artificial Intelligence in Agriculture, vol. 2, pp. 1–12, Jun. 2019, doi: 10.1016/j.aiia.2019.05.004.

12 D. Patel, Y. Shah, N. Thakkar, K. Shah, and M. Shah, “Implementation of Artificial Intelligence Techniques for Cancer Detection,” Augmented Human Research, vol. 5, no. 1, Nov. 2019, doi: 10.1007/s41133-019-0024-3.

13 D. Patel, D. Shah, and M. Shah, “The Intertwine of Brain and Body: A Quantitative Analysis on How Big Data Influences the System of Sports,” Annals of Data Science, vol. 7, no. 1, pp. 1–16, Jan. 2020, doi: 10.1007/s40745-019-00239-y.

14 V. Kakkad, M. Patel, and M. Shah, “Biometric authentication and image encryption for image security in cloud framework,” Multiscale and Multidisciplinary Modeling, Experiments and Design, vol. 2, no. 4, pp. 233–248, May 2019, doi: 10.1007/s41939-019-00049-y.

15 M. Gandhi, J. Kamdar, and M. Shah, “Preprocessing of Non-symmetrical Images for Edge Detection,” Augmented Human Research, vol. 5, no. 1, Dec. 2019, doi: 10.1007/s41133-019-0030-5.

16 V. Gupta, R. Kumar, K. S. Reddy, and B. K. Panigrahi, “Intelligent traffic light control for congestion management for smart city development,” in 2017 IEEE Region 10 Symposium (TENSYMP), Jul. 2017, pp. 1–5. doi: 10.1109/TENCONSpring.2017.8070077.

17 C. Mcshane, “The Origins and Globalization of Traffic Control Signals,” Journal of Urban History, vol. 25, no. 3, pp. 379–404, Mar. 1999, doi: 10.1177/009614429902500304.

18 A. Saikar, M. Parulekar, A. Badve, S. Thakkar, and A. Deshmukh, “TrafficIntel: Smart traffic management for smart cities,” in 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI), Feb. 2017, pp. 46–50. doi: 10.1109/ETIICT.2017.7977008.

19 M. Y. Siyal, M. Fathy, and F. Dorry, “Neural-vision based approach for real-time road traffic applications,” Electronics Letters, vol. 33, no. 11, p. 969, 1997, doi: 10.1049/el:19970646.

20 BBC, “The man who gave us traffic lights,” www.bbc.co.uk. https://www.bbc.co.uk/nottingham/content/articles/2009/07/16/john_peake_knight_traffic_lights_feature.shtml (accessed Jul. 24, 2022).

21 G. M. Sessions, Traffic devices: historical aspects thereof. Washington: Institute of Traffic Engineers, 1971. Accessed: Jul. 24, 2022. [Online]. Available: https://openlibrary.org/books/OL5324456M/Traffic_devices_historical_aspects_thereof

22 Wikipedia Contributors, “Traffic light,” Wikipedia, May 28, 2019. https://en.wikipedia.org/wiki/Traffic_light (accessed Jul. 24, 2022).

23 C.-H. Chou and J.-C. Teng, “A fuzzy logic controller for traffic junction signals,” Information Sciences, vol. 143, no. 1–4, pp. 73–97, Jun. 2002, doi: 10.1016/s0020-0255(02)00199-8.

24 C. L. Wan and K. W. Dickinson, “Road Traffic Monitoring Using Image Processing—A Survey of Systems, Techniques and Applications,” IFAC Proceedings Volumes, vol. 23, no. 2, pp. 27–34, Sep. 1990, doi: 10.1016/s1474-6670(17)52646-4.

25 T. Chowdhury, S. Singh, and S. M. Shaby, “A Rescue System of an advanced ambulance using prioritized traffic switching,” in 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Mar. 2015, pp. 1–5. doi: 10.1109/iciiecs.2015.7193161.

26 E. Bingham, “Reinforcement learning in neurofuzzy traffic signal control,” European Journal of Operational Research, vol. 131, no. 2, pp. 232–241, Jun. 2001, doi: 10.1016/s0377-2217(00)00123-5.

27 S. Araghi, A. Khosravi, and D. Creighton, “Design of an Optimal ANFIS Traffic Signal Controller by Using Cuckoo Search for an Isolated Intersection,” in 2015 IEEE International Conference on Systems, Man, and Cybernetics, Oct. 2015, pp. 2078–2083. doi: 10.1109/SMC.2015.363.

28 C. Ozkurt and F. Camci, “AUTOMATIC TRAFFIC DENSITY ESTIMATION AND VEHICLE CLASSIFICATION FOR TRAFFIC SURVEILLANCE SYSTEMS USING NEURAL NETWORKS,” Mathematical and Computational Applications, vol. 14, no. 3, pp. 187–196, Dec. 2009, doi: https://doi.org/10.3390/mca14030187.

29 Suryanto, D.-H. Kim, H.-K. Kim, and S.-J. Ko, “Spatial color histogram based center voting method for subsequent object tracking and segmentation,” Image and Vision Computing, vol. 29, no. 12, pp. 850–860, Nov. 2011, doi: 10.1016/j.imavis.2011.09.008.

30 A. Kembhavi, D. Harwood, and L. S. Davis, “Vehicle detection using partial least squares,” IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 6, pp. 1250–1265, Jun. 2011, doi: 10.1109/TPAMI.2010.182.

31 J. Arróspide, L. Salgado, and J. Marinas, “HOG-like gradient-based descriptor for visual vehicle detection,” in 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, Jul. 2012.

32 J. Arrospide and L. Salgado, “Log-Gabor Filters for Image-Based Vehicle Verification,” IEEE Transactions on Image Processing, vol. 22, no. 6, pp. 2286–2295, Jun. 2013, doi: 10.1109/tip.2013.2249080.

33 B. Zhang, “Reliable Classification of Vehicle Types Based on Cascade Classifier Ensembles,” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 1, pp. 322–332, Mar. 2013, doi: 10.1109/tits.2012.2213814.

34 J. Arróspide and L. Salgado, “A Study of Feature Combination for Vehicle Detection Based on Image Processing,” The Scientific World Journal, vol. 196251, pp. 1–13, 2014, doi: 10.1155/2014/196251.

35 B. Qi, V. John, Z. Liu, and S. Mita, “Use of Sparse Representation for Pedestrian Detection in Thermal Images,” in the Workshop on Perception Beyond the Visible Spectrum, Jun. 2014.

36 P. Dollar, R. Appel, S. Belongie, and P. Perona, “Fast Feature Pyramids for Object Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 8, pp. 1532–1545, Aug. 2014, doi: 10.1109/tpami.2014.2300479.

37 A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-Scale Video Classification with Convolutional Neural Networks,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Jun. 2014, doi: 10.1109/cvpr.2014.223.

38 K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv.org, 2014. https://arxiv.org/abs/1409.1556.

39 V. John, Z. Zheng, S. Mita, and B. Qi, “Pedestrian Detection in Thermal Images Using Adaptive Fuzzy C-Means Clustering and Convolutional Neural Networks,” in the International Conference of Machine Vision Applications, Japan, May 2015.

40 S. Zangenehpour, L. F. Miranda-Moreno, and N. Saunier, “Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic: Methodology and application,” Transportation Research Part C: Emerging Technologies, vol. 56, pp. 161–176, Jul. 2015, doi: 10.1016/j.trc.2015.04.003.

41 K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/tpami.2015.2389824.

42 A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, and D. Ferguson, “Real-Time Pedestrian Detection with Deep Network Cascades,” in the British Machine Vision Conference, 2015, doi: 10.5244/c.29.32.

43 R. Girshick, “Fast R-CNN,” in 2015 IEEE International Conference on Computer Vision (ICCV), in 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015, pp. 1440–1448, doi: 10.1109/iccv.2015.169.

44 Z. Dong, Y. Wu, M. Pei, and Y. Jia, “Vehicle Type Classification Using a Semisupervised Convolutional Neural Network,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2247–2256, Aug. 2015, doi: 10.1109/tits.2015.2402438.

45 Q. Fan, L. Brown, and J. Smith, “A closer look at Faster R-CNN for vehicle detection,” in 2016 IEEE Intelligent Vehicles Symposium (IV), Jun. 2016, pp. 124–129, doi: 10.1109/IVS.2016.7535375.

46 T. Fu, J. Stipancic, L. Miranda-Moreno, S. Zangenehpour, and N. Saunier, “Traffic Data Collection Using Thermal Camera Under Varying Lighting and Temperature Conditions in Multimodal Environments,” in the 2016 Conference and Exhibition of the Transportation Association of Canada, 2016, Accessed: Jan. 03, 2022. [Online]. Available: https://trid.trb.org/view/1434869

47 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/cvpr.2016.91.

48 W. Liu et al., “SSD: Single Shot MultiBox Detector,” in Computer Vision – ECCV 2016, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

49 T. Kong, A. Yao, Y. Chen, and F. Sun, “HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 845–853, doi: 10.1109/CVPR.2016.98.

50 S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 2874–2883, doi: 10.1109/CVPR.2016.314.

51 J. Dai, Y. Li, K. Hi, and J. Sun, “R-FCN: object detection via region-based fully convolutional networks,” in Proceedings of the 30th International Conference on Neural Information Processing Systems, Dec. 2016, pp. 1–9.

52 K. Kang, W. Ouyang, H. Li, and X. Wang, "Object Detection from Video Tubelets with Convolutional Neural Networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 817–825, doi: 10.1109/cvpr.2016.95.

53 P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to Refine Object Segments," in Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 2016, vol. 9905, pp. 75–91, doi: 10.1007/978-3-319-46448-0_5.

54 J. Dai, K. He, and J. Sun, "Instance-Aware Semantic Segmentation via Multi-task Network Cascades," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3150–3158.

55 Y. Zhou, L. Liu, L. Shao, and M. Mellor, "DAVE: A Unified Framework for Fast Vehicle Detection and Annotation," in Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 2016, vol. 9906, pp. 278–293, doi: 10.1007/978-3-319-46475-6_18.

56 G. Yan, M. Yu, Y. Yu, and L. Fan, "Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification," Optik, vol. 127, no. 19, pp. 7941–7951, Oct. 2016, doi: 10.1016/j.ijleo.2016.05.092.

57 S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.

58 I. Ullah and H. J. Lee, "Moving Vehicle Detection and Information Extraction Based on Deep Neural Network," in Int'l Conf. IP, Comp. Vision, and Pattern Recognition | IPCV'17, Jul. 2017, pp. 102–107.

59 H. Tayara, K. G. Soo, and K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," IEEE Access, vol. 6, pp. 2220–2230, Dec. 2017, doi: 10.1109/ACCESS.2017.2782260.

60 J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware Fast R-CNN for Pedestrian Detection," IEEE Transactions on Multimedia, vol. 20, no. 4, pp. 985–996, 2017, doi: 10.1109/tmm.2017.2759508.

61 A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

62 J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 6517–6525, doi: 10.1109/cvpr.2017.690.

63 T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 2999–3007, doi: 10.1109/iccv.2017.324.

64 N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS — Improving Object Detection with One Line of Code,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 5562–5570, doi: 10.1109/iccv.2017.593.

65 J. Huang et al., “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 3296–3297, doi: 10.1109/cvpr.2017.351.

66 S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, “A model for fine-grained vehicle classification based on deep learning,” Neurocomputing, vol. 257, pp. 97–103, Sep. 2017, doi: 10.1016/j.neucom.2016.09.116.

67 X. Wang, W. Zhang, X. Wu, L. Xiao, Y. Qian, and Z. Fang, “Real-time vehicle type classification with deep convolutional neural networks,” Journal of Real-Time Image Processing, vol. 16, no. 1, pp. 5–14, Aug. 2017, doi: 10.1007/s11554-017-0712-5.

68 L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, “Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks,” Machine Vision and Applications, vol. 28, no. 7, pp. 793–802, May 2017, doi: 10.1007/s00138-017-0846-2.

69 V. A. Sindagi and V. M. Patel, “A survey of recent advances in CNN-based single image crowd counting and density estimation,” Pattern Recognition Letters, vol. 107, pp. 3–16, May 2018, doi: 10.1016/j.patrec.2017.07.007.

70 L. Suhao, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian, and P. Yu, “Vehicle type detection based on deep learning in traffic scene,” Procedia Computer Science, vol. 131, pp. 564–572, 2018, doi: 10.1016/j.procs.2018.04.281.

71 A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, “Detection and classification of vehicles for traffic video analytics,” Procedia Computer Science, vol. 144, pp. 259–268, 2018, doi: 10.1016/j.procs.2018.10.527.

72 C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, “Vehicle Detection and Classification based on Deep Neural Network for Intelligent Transportation Applications,” in 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and

Conference (APSIPA ASC), Nov. 2018, pp. 1605–1608, doi: 10.23919/APSIPA.2018.8659542.

73 Y. Han, T. Jiang, Y. Ma, and C. Xu, "Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification," Advances in Multimedia, vol. 2018, pp. 1–10, Oct. 2018, doi: 10.1155/2018/3138278.

74 M. Sheeny, A. Wallace, M. Emambakhsh, S. Wang, and B. Connor, "POL-LWIR Vehicle Detection: Convolutional Neural Networks Meet Polarised Infrared Sensors," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2018, pp. 1328–13286, doi: 10.1109/CVPRW.2018.00171.

75 Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

76 H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A Low-Shot Transfer Detector for Object Detection," in Proceedings of th AAAI Conference on Artificial Intelligence, Apr. 2018, vol. 32, no. 1, [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11716.

77 Y. Nam and Y.-C. Nam, "Vehicle classification based on images from visible light and thermal cameras," EURASIP Journal on Image and Video Processing, vol. 2018, Jan. 2018, doi: 10.1186/s13640-018-0245-2.

78 W. Liu, Z. Luo, and S. Li, "Improving deep ensemble vehicle classification by using selected adversarial samples," Knowledge-Based Systems, vol. 160, pp. 167–175, Nov. 2018, doi: 10.1016/j.knosys.2018.06.035.

79 V. Murugan and V. R. Vijaykumar, "Automatic Moving Vehicle Detection and Classification Based on Artificial Neural Fuzzy Inference System," Wireless Personal Communications, vol. 100, no. 3, pp. 745–766, Jan. 2018, doi: 10.1007/s11277-018-5347-8.

80 Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask Scoring R-CNN," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, pp. 6402–6411, doi: 10.1109/CVPR.2019.00657.

81 H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," International Journal of Computer Vision, vol. 128, no. 3, pp. 642–656, Aug. 2019, doi: 10.1007/s11263-019-01204-1.

82 K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint Triplets for Object Detection,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, pp. 6568–6577, doi: 10.1109/iccv.2019.00667.

83 G. Ghiasi, T.-Y. Lin, and Q. V. Le, “NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection,” in the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, doi: 10.1109/cvpr.2019.00720.

84 Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, pp. 9626–9635, doi: 10.1109/ICCV.2019.00972.

85 S. M. Azimi, “ShuffleDet: Real-Time Vehicle Detection Network in On-Board Embedded UAV Imagery,” in European Conference on Computer Vision ECCV 2018: Computer Vision – ECCV 2018 Workshops. Lecture Notes in Computer Science, 2019, pp. 88–99, doi: 10.1007/978-3-030-11012-3_7.

86 H. Kim, “Multiple vehicle tracking and classification system with a convolutional neural network,” Journal of Ambient Intelligence and Humanized Computing, Aug. 2019, doi: 10.1007/s12652-019-01429-5.

87 D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, “An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD,” Physics and Chemistry of the Earth, Parts, vol. 110, pp. 176–184, 2019, doi: 10.1016/J.PCE.2018.12.001.

88 W. Sun, G. Zhang, X. Zhang, X. Zhang, and N. Ge, “Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy,” Multimedia Tools and Applications, vol. 80, no. 20, pp. 30803–30816, Jul. 2020, doi: 10.1007/s11042-020-09171-3.

89 C. Ranjeeth Kumar and R. Anuradha, “Feature selection and classification methods for vehicle tracking and detection,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 4269–4279, Mar. 2020, doi: 10.1007/s12652-020-01824-3.

90 X. Wang, X. Chen, and Y. Wang, “Small vehicle classification in the wild using generative adversarial network,” Neural Computing and Applications, vol. 33, pp. 5369–5379, Sep. 2020, doi: 10.1007/s00521-020-05331-6.

91 N. Shvai, A. Hasnat, A. Meicler, and A. Nakib, “Accurate Classification for Automatic Vehicle-Type Recognition Based on Ensemble Classifiers,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1288–1297, Mar. 2020, doi: 10.1109/tits.2019.2906821.

92 S. Awang, N. M. A. N. Azmi, and Md. A. Rahman, “Vehicle Type Classification Using an Enhanced Sparse-Filtered Convolutional Neural Network With Layer-Skipping Strategy,” IEEE Access, vol. 8, pp. 14265–14277, 2020, doi: 10.1109/access.2019.2963486.

93 A. Grents, V. Varkentin, and N. Goryaev, “Determining vehicle speed based on video using convolutional neural network,” Transportation Research Procedia, vol. 50, pp. 192–200, 2020, doi: 10.1016/j.trpro.2020.10.024.

94 J. Zhu, X. Li, P. Jin, Q. Xu, Z. Sun, and X. Song, “MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance,” Sensors, vol. 21, no. 1, p. 27, Dec. 2020, doi: 10.3390/s21010027.

95 P. Jagannathan, S. Rajkumar, J. Frnda, P. B. Divakarachari, and P. Subramani, “Moving Vehicle Detection and Classification Using Gaussian Mixture Model and Ensemble Deep Learning Technique,” Wireless Communications and Mobile Computing, vol. 2021, May 2021, doi: 10.1155/2021/5590894.

96 X. Hu, Z. Wei, and W. Zhou, “A video streaming vehicle detection algorithm based on YOLOv4,” in 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Mar. 2021, pp. 2081–2086, doi: 10.1109/IAEAC50856.2021.9390613.

97 Y. Yang et al., “A fast and effective video vehicle detection method leveraging feature fusion and proposal temporal link,” Journal of Real-Time Image Processing, vol. 18, no. 4, pp. 1261–1274, May 2021, doi: 10.1007/s11554-021-01121-y.

98 S. J. S and E. R. P, “LittleYOLO-SPP: A delicate real-time vehicle detection algorithm,” Optik, vol. 225, p. 165818, Jan. 2021, doi: 10.1016/j.ijleo.2020.165818.

99 X. Wang, “Vehicle Image Detection Method Using Deep Learning in UAV Video,” Computational Intelligence and Neuroscience, vol. 2022, Feb. 2022, doi: 10.1155/2022/8202535.

100 C. Szegedy et al., “Going deeper with convolutions,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9. doi: 10.1109/cvpr.2015.7298594.

101 R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2014, pp. 580–587. doi: 10.1109/cvpr.2014.81.

102 K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in Computer Vision – ECCV 2016, 2016, pp. 630–645. doi: 10.1007/978-3-319-46493-0_38.

103 A. Newell, K. Yang, and J. Deng, “Stacked Hourglass Networks for Human Pose Estimation,” in Computer Vision – ECCV 2016, 2016, pp. 483–499. doi: 10.1007/978-3-319-46484-8_29.

104 G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 2261–2269. doi: 10.1109/cvpr.2017.243.

105 Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, “Dual Path Networks,” arXiv.org, 2017. https://arxiv.org/abs/1707.01629 (accessed Jul. 24, 2022).

106 T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 936–944. doi: 10.1109/cvpr.2017.106.

107 A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” www.arxiv-vanity.com, Apr. 17, 2017. https://www.arxiv-vanity.com/papers/1704.04861/ (accessed Jul. 24, 2022).

108 M. Maithani, “EfficientDet: Guide to State of The Art Object Detection Model,” Analytics India Magazine, Dec. 17, 2020. https://analyticsindiamag.com/efficientdet/ (accessed Jul. 24, 2022).

109 B. Fazenda, H. Atmoko, F. Gu, L. Guan, and A. Ball, “Acoustic based safety emergency vehicle detection for intelligent transport systems,” in ICCAS-SICE, Aug. 2009, pp. 4250–4255,

110 J.-J. Liaw, W.-S. Wang, H.-C. Chu, M.-S. Huang, and C.-P. Lu, “Recognition of the Ambulance Siren Sound in Taiwan by the Longest Common Subsequence,” in IEEE International Conference on Systems, Man, and Cybernetics, Oct. 2013, pp. 3825–3828, doi: 10.1109/SMC.2013.653.

111 J. Schröder, S. Goetze, V. Grützmacher, and J. Anemüller, “Automatic acoustic siren detection in traffic noise by part-based models,” in IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, May 2013, pp. 493–497, doi: 10.1109/ICASSP.2013.6637696.

112 T. Miyazaki, Y. Kitazono, and M. Shimakawa, “Ambulance Siren Detector using FFT on dsPIC,” in the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing 2013, 2013, doi: 10.12792/icisip2013.052.

113 R. Sundar, S. Hebbar, and V. Golla, “Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection,” IEEE Sensors Journal, vol. 15, no. 2, pp. 1109–1113, Feb. 2015, doi: 10.1109/jsen.2014.2360288.

114 R. A. Dobre, V. A. Niţă, A. Ciobanu, C. Negrescu, and D. Stanomir, “Low computational method for siren detection,” in 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), Oct. 2015, pp. 291–295, doi: 10.1109/SIITME.2015.7342342.

115 A. A. Saad, H. A. E. Zouka, and S. A. Al-Soufi, “Secure and Intelligent Road Traffic Management System Based on RFID Technology,” in World Symposium on Computer Applications & Research (WSCAR), Mar. 2016, pp. 41–46, doi: 10.1109/WSCAR.2016.9.

116 Md. R. Islam, N. I. Shahid, D. T. ul Karim, A. A. Mamun, and Md. K. Rhaman, “An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system,” in International Conference on Advanced Communication Technology (ICACT), Jan. 2016, pp. 802–807, doi: 10.1109/ICACT.2016.7423566.

117 S. Amir, M. S. Kamal, S. S. Khan, and K. M. A. Salam, “PLC based traffic control system with emergency vehicle detection and management,” in International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Jul. 2017, pp. 1467–1472, doi: 10.1109/ICICICT1.2017.8342786.

118 B. S. Meghana, S. Kumari, and T. P. Pushphavathi, “Comprehensive traffic management system: Real-time traffic data analysis using RFID,” in International conference of Electronics, Communication and Aerospace Technology (ICECA), Apr. 2017, vol. 2, pp. 168–171, doi: 10.1109/ICECA.2017.8212787.

119 T. Naik, R. Roopalakshmi, N. Divya Ravi, P. Jain, B. H. Sowmya, and Manichandra, “RFID-Based Smart Traffic Control Framework for Emergency Vehicles,” in Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Apr. 2018, pp. 398–401, doi: 10.1109/ICICCT.2018.8473001.

120 S. V. Bhate, P. V. Kulkarni, S. D. Lagad, M. D. Shinde, and S. Patil, “IoT based Intelligent Traffic Signal System for Emergency vehicles,” in Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Apr. 2018, pp. 788–793, doi: 10.1109/ICICCT.2018.8473210.

121 A. Dutta et al., “Intelligent Traffic Control System: Towards Smart City,” in 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct. 2019, pp. 1124–1129, doi: 10.1109/IEMCON.2019.8936188.

122 Y. Ebizuka, S. Kato, and M. Itami, “Detecting approach of emergency vehicles using siren sound processing,” in IEEE Intelligent Transportation Systems Conference (ITSC), Oct. 2019, pp. 4431–4436, doi: 10.1109/ITSC.2019.8917028.

123 B. Sara, A. Noureddine, and B. Hajar, “Ambulance Detection System,” in Advanced Intelligent Systems for Sustainable Development (AI2SD’2019) . Lecture Notes in Networks and Systems, Oct. 2019, vol. 92, pp. 19–25, doi: 10.1007/978-3-030-33103-0_3.

124 S. Goel, A. Baghel, A. Srivastava, A. Tyagi, and P. Nagrath, “Detection of Emergency Vehicles Using Modified Yolo Algorithm,” Advances in Intelligent Systems and Computing, vol. 989, pp. 671–687, Aug. 2019, doi: 10.1007/978-981-13-8618-3_69.

125 S. Roy and M. S. Rahman, “Emergency Vehicle Detection on Heavy Traffic Road from CCTV Footage Using Deep Convolutional Neural Network,” in International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox’sBazar, Bangladesh, Feb. 2019, pp. 1–6, doi: 10.1109/ECACE.2019.8679295.

126 A. Raman, S. Kaushik, K. V. S. R. Rao, and M. Moharir, “A Hybrid Framework for Expediting Emergency Vehicle Movement on Indian Roads,” in 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Mar. 2020, pp. 459–464, doi: 10.1109/ICIMIA48430.2020.9074933.

127 V. Tran and W. Tsai, “Acoustic-Based Emergency Vehicle Detection Using Convolutional Neural Networks,” IEEE Access, vol. 8, pp. 75702–75713, 2020, doi: 10.1109/ACCESS.2020.2988986.

128 S. Shirabur, S. Hunagund, and S. Murgd, “VANET Based Embedded Traffic Control System,” in International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Nov. 2020, pp. 189–192, doi: 10.1109/RTEICT49044.2020.9315602.

129 G. Karmakar, A. Chowdhury, J. Kamruzzaman, and I. Gondal, “A Smart Priority Based Traffic Control System for Emergency Vehicles,” IEEE Sensors Journal, vol. 21, no. 14, pp. 15849–15858, 2020, doi: 10.1109/jsen.2020.3023149.

130 H. V. Supreeth, S. Rao, K. S. Chethan, and U. Purushotham., “Identification of Ambulance Siren sound and Analysis of the signal using statistical method,” in

International Conference on Intelligent Engineering and Management (ICIEM), Jun. 2020, pp. 198–202, doi: 10.1109/ICIEM48762.2020.9160070.
131 B. Fatimah, A. Preethi, V. Hrushikesh, A. Singh B., and H. R. Kotion, “An automatic siren detection algorithm using Fourier Decomposition Method and MFCC,” in 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Jul. 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225414.
132 A. Baghel, A. Srivastava, A. Tyagi, S. Goel, and P. Nagrath, “Analysis of Ex-YOLO Algorithm with Other Real-Time Algorithms for Emergency Vehicle Detection,” in Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019). Lecture Notes in Networks and Systems, 2020, vol. 121, pp. 607–618, doi: 10.1007/978-981-15-3369-3_45.
133 V.-T. Tran and W.-H. Tsai, “Audio-Vision Emergency Vehicle Detection,” IEEE Sensors Journal, vol. 21, no. 24, pp. 27905–27917, Dec. 2021, doi: 10.1109/JSEN.2021.3127893.
134 M. Nakatsuyama, H. Nagahashi, and N. Nishizuka, “Fuzzy Logic Phase Controller for Traffic Junctions in the One-way Arterial Road,” IFAC Proceedings Volumes, vol. 17, no. 2, pp. 2865–2870, Jul. 1984, doi: 10.1016/s1474-6670(17)61417-4.
135 S. Chinu, “Adaptive traffic signal control using fuzzy logic,” in Proceedings of the Intelligent Vehicles `92 Symposium, pp. 98–107
136 K. Kok, Tan, M. Khalid, and R. Yusof, “INTELLIGENT TRAFFIC LIGHTS CONTROL BY FUZZY LOGIC,” Malaysian Journal of Computer Science, vol. 9, no. 2, pp. 29–35, 1996.
137 M. B. Trabia, M. S. Kaseko, and M. Ande, “A two-stage fuzzy logic controller for traffic signals,” Transportation Research Part C: Emerging Technologies, vol. 7, no. 6, pp. 353–367, Dec. 1999, doi: 10.1016/S0968-090X(99)00026-1.
138 P. Mirchandani and L. Head, “A real-time traffic signal control system: architecture, algorithms, and analysis,” Transportation Research Part C: Emerging Technologies, vol. 9, no. 6, pp. 415–432, Dec. 2001, doi: 10.1016/s0968-090x(00)00047-4.
139 C.-H. Chou and J.-C. Teng, “A fuzzy logic controller for traffic junction signals,” Information Sciences, vol. 143, no. 1–4, pp. 73–97, Jun. 2002, doi: 10.1016/s0020-0255(02)00199-8.
140 Y. Li and X. Fan, “Design of signal controllers for urban intersections based on fuzzy logic and weightings,” in 2003 IEEE International Conference on Intelligent

Transportation Systems, Oct. 2003, vol. 1, pp. 867–871 vol.1. doi: 10.1109/ITSC.2003.1252073.

141 Y.-C. Chiou and L. W. Lan, “Adaptive traffic signal control with iterative genetic fuzzy logic controller (GFLC),” in IEEE International Conference on Networking, Sensing and Control, Mar. 2004, vol. 1, pp. 287–292 Vol.1, doi: 10.1109/ICNSC.2004.1297450.

142 L. Conglin, W. Wu, and T. Yuejin, “Traffic variable estimation and traffic signal control based on soft computation,” in The 7th International IEEE Conference on Intelligent Transportation Systems, Oct. 2004, pp. 1045–1050. doi: 10.1109/ITSC.2004.1399051.

143 Y. S. Murat and E. Gedizlioglu, “A fuzzy logic multi-phased signal control model for isolated junctions,” Transportation Research Part C: Emerging Technologies, vol. 13, no. 1, pp. 19–36, Feb. 2005, doi: 10.1016/j.trc.2004.12.004.

144 R. Zeng, G. Li, and L. Lin, “Adaptive Traffic Signals Control by Using Fuzzy Logic,” presented at the Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), Sep. 2007. doi: 10.1109/icicic.2007.118.

145 B. M. Nair and J. Cai, “A fuzzy Logic Controller for Isolated Signalized Intersection with Traffic Abnormality Considered,” in 2007 IEEE Intelligent Vehicles Symposium, Jun. 2007, pp. 1229–1233. doi: 10.1109/IVS.2007.4290286.

146 C. T. Wannige and D. Sonnadara, “Traffic Signal Control Based on Adaptive Neuro-Fuzzy Inference,” in 2008 4th International Conference on Information and Automation for Sustainability, 2008, pp. 301–306. doi: 10.1109/ICIAFS.2008.4783976.

147 X. Zheng and L. Chu, “Optimal Parameter Settings for Adaptive Traffic-Actuated Signal Control,” in 11th International IEEE Conference on Intelligent Transportation Systems, Oct. 2008, pp. 105–110. doi: 10.1109/ITSC.2008.4732676.

148 C. T. Wannige and D. U. J. Sonnadara, “Adaptive neuro-fuzzy traffic signal control for multiple junctions,” in 2009 International Conference on Industrial and Information Systems (ICIIS), Dec. 2009, pp. 262–267. doi: 10.1109/ICIINFS.2009.5429853.

149 M. Franceschinis, L. Gioanola, M. Messere, R. Tomasi, M. A. Spirito, and P. Civera, “Wireless Sensor Networks for Intelligent Transportation Systems,” in VTC Spring 2009 - IEEE 69th Vehicular Technology Conference, Apr. 2009, pp. 1–5. doi: 10.1109/VETECS.2009.5073915.

150 A. C, M. Kumar, and P. Kumar, “City traffic congestion control in Indian scenario using wireless sensors network,” in 2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN), Dec. 2009, pp. 1–6. doi: 10.1109/WCSN.2009.5434809.

151 B. P. Gokulan and D. Srinivasan, “Distributed Geometric Fuzzy Multiagent Urban Traffic Signal Control,” IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pp. 714–727, Sep. 2010, doi: 10.1109/tits.2010.2050688.

152 E. Azimirad, N. Pariz, and M. B. N. Sistani, “A Novel Fuzzy Model and Control of Single Intersection at Urban Traffic Network,” IEEE Systems Journal, vol. 4, no. 1, pp. 107–111, Mar. 2010, doi: 10.1109/JSYST.2010.2043159

153 L. Wu, X. Zhang, and Z. Shi, “An Intelligent Fuzzy Control for Crossroads Traffic Light,” in Second WRI Global Congress on Intelligent Systems, Dec. 2010, vol. 3, pp. 28–32. doi: 10.1109/GCIS.2010.59.

154 K. Yousef, J. Al-Karak, and A. Shatnawi, “Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks,” JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, vol. 26, pp. 753–768, 2010.

155 B. Zhou, J. Cao, X. Zeng, and H. Wu, “Adaptive Traffic Light Control in Wireless Sensor Network-Based Intelligent Transportation System,” in 2010 IEEE 72nd Vehicular Technology Conference - Fall, Sep. 2010, pp. 1–5. doi: 10.1109/vetecf.2010.5594435.

156 M. Saqib and C. Lee, “Traffic control system using wireless sensor network,” in The 12th International Conference on Advanced Communication Technology (ICACT), Feb. 2010, vol. 1, pp. 352–357.

157 P. Choudekar, S. Banerjee, and M. K. Muju, “Implementation of image processing in real time traffic light control,” in 3rd International Conference on Electronics Computer Technology, Apr. 2011, vol. 2, pp. 94–98, doi: 10.1109/ICECTECH.2011.5941662.

158 A. C. Soh, R. Z. A. Rahman, L. G. Rhung, and H. Md. Sarkan, “Traffic signal control based on Adaptive Neural-Fuzzy Inference System applied to intersection,” in IEEE Conference on Open Systems, Sep. 2011, pp. 231–236. doi: 10.1109/ICOS.2011.6079251.

159 A. N. H. Zaied and W. Al Othman, “Development of a fuzzy logic traffic system for isolated signalized intersections in the State of Kuwait,” Expert Systems with Applications, vol. 38, no. 8, pp. 9434–9441, Aug. 2011, doi: 10.1016/j.eswa.2011.01.130.

160 P. G. Balaji and D. Srinivasan, “Type-2 fuzzy logic based urban traffic management,” Engineering Applications of Artificial Intelligence, vol. 24, no. 1, pp. 12–22, Feb. 2011, doi: 10.1016/j.engappai.2010.08.007.

161 B. Zhou, J. Cao, X. Zeng, and H. Wu, “Adaptive Traffic Light Control in Wireless Sensor Network-Based Intelligent Transportation System,” in 2010 IEEE 72nd Vehicular Technology Conference - Fall, Sep. 2010, pp. 1–5. doi: 10.1109/vetecf.2010.5594435.

162 A. R. Zade and D. R. Dandekar, “Simulation of Adaptive Traffic Signal Controller in MATLAB Simulink Based On Fuzzy Inference System,” in IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012), Oct. 2012, vol. ncipet, no. 2, pp. 9–13.

163 S. Faye, C. Chaudet, and I. Demeure, “A distributed algorithm for multiple intersections adaptive traffic lights control using a wireless sensor networks,” in Proceedings of the first workshop on Urban networking, 2012, pp. 13–18.

164 P. T. V. Bhuvaneswari, G. V. A. Raj, R. Balaji, and S. Kanagasabai, “Adaptive Traffic Signal Flow Control Using Wireless Sensor Networks,” in 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Nov. 2012, pp. 85–89. doi: 10.1109/CICN.2012.38.

165 R. Bharadwaj, J. Deepak, M. Baranitharan, and V. V. Vaidehi, “Efficient dynamic traffic control system using wireless sensor networks,” in 2013 International Conference on Recent Trends in Information Technology (ICRTIT), Jul. 2013, pp. 668–673. doi: 10.1109/ICRTIT.2013.6844280.

166 J. R. Srivastava and T. S. B. Sudarshan, “Intelligent traffic management with wireless sensor networks,” in 2013 ACS International Conference on Computer Systems and Applications (AICCSA), May 2013, pp. 1–4. doi: 10.1109/AICCSA.2013.6616429.

167 R. Hussian, S. Sharma, and V. K. Sharma, “WSN Applications: Automated Intelligent Traffic Control System Using Sensors,” International Journal of Soft Computing & Engineering, vol. 3, no. 3, 2013.

168 F. Ahmad, A. Basit, H. Ahmad, S. A. Mahmud, G. M. Khan, and F. Z. Yousaf, “Feasibility of deploying wireless sensor based road side solutions for Intelligent Transportation Systems,” in 2013 International Conference on Connected Vehicles and Expo (ICCVE), Dec. 2013, pp. 320–326. doi: 10.1109/ICCVE.2013.6799814.

169 J. Zhou, C. L. P. Chen, L. Chen, and W. Zhao, “A User-Customizable Urban Traffic Information Collection Method Based on Wireless Sensor Networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 3, pp. 1119–1128, Sep. 2013, doi: 10.1109/TITS.2013.2252902.

170 R. Bodenheimer, A. Brauer, D. Eckhoff, and R. German, “Enabling GLOSA for adaptive traffic lights,” in IEEE Vehicular Networking Conference (VNC), Dec. 2014, pp. 167–174, doi: 10.1109/VNC.2014.7013336.

171 Y. Bi, D. Srinivasan, X. Lu, Z. Sun, and W. Zeng, “Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization,” Expert Systems with

Applications, vol. 41, no. 16, pp. 7338–7349, Nov. 2014, doi: 10.1016/j.eswa.2014.06.022.

172 Y. Lin, "The Design and Simulation of Intelligent Traffic Signal Control System Based on Fuzzy Logic," in Foundations of Intelligent Systems. Advances in Intelligent Systems and Computing, 2014, vol. 277.

173 K.-H. Chao and P.-Y. Chen, "An Intelligent Traffic Flow Control System Based on Radio Frequency Identification and Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. 10, no. 5, p. 694545, Jan. 2014, doi: 10.1155/2014/694545.

174 S. P. Biswas, P. Roy, N. Patra, A. Mukherjee, and N. Dey, "Intelligent Traffic Monitoring System," in Proceedings of the Second International Conference on Computer and Communication Technologies. Advances in Intelligent Systems and Computing, Sep. 2015, pp. 535–545, doi: 10.1007/978-81-322-2523-2_52.

175 M. Collotta, L. Lo Bello, and G. Pau, "A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers," Expert Systems with Applications, vol. 42, no. 13, pp. 5403–5415, Aug. 2015, doi: 10.1016/j.eswa.2015.02.011.

176 Y. M. Jagadeesh, G. M. Suba, S. Karthik, and K. Yokesh, "Smart autonomous traffic light switching by traffic density measurement through sensors," in International Conference on Computers, Communications, and Systems (ICCCS), Nov. 2015, pp. 123–126, doi: 10.1109/CCOMS.2015.7562885.

177 S. M. Odeh, A. M. Mora, M. N. Moreno, and J. J. Merelo, "A Hybrid Fuzzy Genetic Algorithm for an Adaptive Traffic Signal System," Advances in Fuzzy Systems, vol. 2015, p. e378156, Sep. 2015, doi: 10.1155/2015/378156.

178 P. S. Chakraborty, P. R. Sinha, and A. Tiwari, "Real Time Optimized Traffic Management Algorithm for Intelligent Transportation Systems," in 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Feb. 2015, pp. 744–749. doi: 10.1109/CICT.2015.39.

179 N. V. Hung, L. C. Tran, N. H. Dung, T. M. Hoang, and N. T. Dzung, "A traffic monitoring system for a mixed traffic flow via road estimation and analysis," in IEEE Sixth International Conference on Communications and Electronics (ICCE), Jul. 2016, pp. 375–378, doi: 10.1109/CCE.2016.7562665.

180 K. Nellore and G. Hancke, "A Survey on Urban Traffic Management System Using Wireless Sensor Networks," Sensors, vol. 16, no. 2, p. 157, Jan. 2016, doi: 10.3390/s16020157.

181 A. Dubey, M. Lakhani, S. Dave, and J. J. Patoliya, "Internet of Things based adaptive traffic management system as a part of Intelligent Transportation System (ITS)," in 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp), Dec. 2017, pp. 1–6, doi: 10.1109/icsoftcomp.2017.8280081.

182 L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, "Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management," Sensors (Basel, Switzerland), vol. 18, no. 2, Feb. 2018, doi: 10.3390/s18020435.

183 D. Hartanti, R. N. Aziza, and P. C. Siswipraptini, "Optimization of smart traffic lights to prevent traffic congestion using fuzzy logic," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 17, no. 1, pp. 320–327, Feb. 2019, doi: 10.12928/telkomnika.v17i1.10129.

184 A. Pughat, A. Pritam, D. Sharma, and S. Gupta, "Adaptive Traffic Light Controller for Vehicular Ad-Hoc Networks," in 2019 International Conference on Signal Processing and Communication (ICSC), Mar. 2019, pp. 338–344. doi: 10.1109/ICSC45622.2019.8938220.

185 M. Balta and ÖzçelikÍ., "A 3-stage fuzzy-decision tree model for traffic signal optimization in urban city via a SDN based VANET architecture," Future Generation Computer Systems, vol. 104, pp. 142–158, Mar. 2020, doi: 10.1016/j.future.2019.10.020.

186 M. E. M. Ali, A. Durdu, S. A. Celtek, and A. Yilmaz, "An Adaptive Method for Traffic Signal Control Based on Fuzzy Logic With Webster and Modified Webster Formula Using SUMO Traffic Simulator," IEEE Access, vol. 9, pp. 102985–102997, 2021, doi: 10.1109/ACCESS.2021.3094270.

187 A. Chabchoub, A. Hamouda, S. Al-Ahmadi, and A. Cherif, "Intelligent Traffic Light Controller using Fuzzy Logic and Image Processing," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, 2021, doi: 10.14569/ijacsa.2021.0120450.

188 X. C. Vuong, R.-F. Mou, T. T. Vu, and H. Van Nguyen, "An Adaptive Method for an Isolated Intersection Under Mixed Traffic Conditions in Hanoi Based on ANFIS Using VISSIM-MATLAB," IEEE Access, vol. 9, pp. 166328–166338, 2021, doi: 10.1109/ACCESS.2021.3135418.

189 J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective Search for Object Recognition,” International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, Apr. 2013, doi: 10.1007/s11263-013-0620-5.

190 E. Bas, A. M. Tekalp, and F. S. Salman, “Automatic Vehicle Counting from Video for Traffic Flow Analysis,” in 2007 IEEE Intelligent Vehicles Symposium, Jun. 2007, pp. 392–397. doi: 10.1109/IVS.2007.4290146.

191 S.-L. . Chang, L.-S. . Chen, Y.-C. . Chung, and S.-W. . Chen, “Automatic License Plate Recognition,” IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 1, pp. 42–53, Mar. 2004, doi: 10.1109/tits.2004.825086.

192 J. Douret and R. Benosman, “A multi-cameras 3D volumetric method for outdoor scenes: a road traffic monitoring application,” in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Aug. 2004, vol. 3, pp. 334–337 Vol.3. doi: 10.1109/ICPR.2004.1334535.

193 J. Fang, Y. Zhou, Y. Yu, and S. Du, “Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 7, pp. 1782–1792, Jul. 2017, doi: 10.1109/tits.2016.2620495.

194 M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person re-identification by symmetry-driven accumulation of local features,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 2010, pp. 2360–2367. doi: 10.1109/CVPR.2010.5539926.

195 P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in 2008 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587597.

196 O. More, “LeNet-5 : Covolution Neural Network Architecture,” Analytics Vidhya, Aug. 21, 2020. https://medium.com/analytics-vidhya/lenet-5-covolution-neural-network-architecture-23b93f75fe01 (accessed Jul. 24, 2022).

197 D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

198 X. Liu, W. Liu, T. Mei, and H. Ma, “A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance,” in Computer Vision – ECCV 2016, 2016, pp. 869–884. doi: 10.1007/978-3-319-46475-6_53.

199 D. de Oliveira and M. Wehrmeister, “Using Deep Learning and Low-Cost RGB and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV,” Sensors, vol. 18, no. 7, p. 2244, Jul. 2018, doi: 10.3390/s18072244.

200 team, “EPCTM Radio Frequency Identity Protocols Generation-2 UHF RFID,” RFID Card, Nov. 11, 2020. https://www.rfidcard.com/epc-radio-frequency-identity-protocols-generation-2-uhf-rfid/ (accessed Jul. 24, 2022).

201 T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, “Supervised model training for overlapping sound events based on unsupervised source separation,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 8677–8681. doi: 10.1109/ICASSP.2013.6639360.

202 A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, “Acoustic event detection in real life recordings,” in 2010 18th European Signal Processing Conference, Aug. 2010, pp. 1267–1271. Accessed: Jul. 24, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/7096611

203 D. Gerhard, “Audio Signal Classification: History and Current Techniques Audio Signal Classification: History and Current Techniques,” 2003. Accessed: Jul. 24, 2022. [Online]. Available: https://www.cs.uregina.ca/Research/Techreports/2003-07.pdf

204 T. Heittola, E. Çakır, and T. Virtanen, “The Machine Learning Approach for Analysis of Sound Scenes and Events,” Computational Analysis of Sound Scenes and Events, pp. 13–40, Sep. 2017, doi: 10.1007/978-3-319-63450-0_2.

205 S. S. Stevens, “A Scale for the Measurement of the Psychological Magnitude Pitch,” Acoustical Society of America Journal, vol. 8, no. 3, p. 185, 1937, doi: 10.1121/1.1915893.

206 S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357–366, Aug. 1980, doi: 10.1109/tassp.1980.1163420.

207 “ISO 7731:2003 Ergonomics — Danger signals for public and work areas — Auditory danger signals,” ISO. https://www.iso.org/standard/33590.html (accessed Jul. 24, 2022).

208 “AudioSet,” Google.com, 2017. https://research.google.com/audioset/ (accessed Jul. 24, 2022).

209 B. McFee, C. Raffel, Dawen Liang, and Oriol Nieto, “librosa: Audio and Music Signal Analysis in Python,” in Python in Science Conference, 2015, pp. 1–7. [Online]. Available:

https://www.researchgate.net/publication/328777063_librosa_Audio_and_Music_Signal_Analysis_in_Python

210 D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv.org, 2014. https://arxiv.org/abs/1412.6980 (accessed Jul. 24, 2022).

211 “Cyrus Mistry’s death brings back the focus on road crashes in India,” Quartz, Sep. 05, 2022. https://qz.com/cyrus-mistry-death-returns-focus-on-india-road-crashes-1849496593#:~:text=However%2C%20it%20has%20brought%20back (accessed Jan. 24, 2023).

212 209.J.J. Liaw, W.S. Wang, H.C. Chu, M.S. Huang and C.P. Lu, “Recognition of the ambulance siren sound in Taiwan by the Longest Common Subsequence,” in IEEE Int. Conf. on Systems, Man, and Cybernetics, 2013

213 L. Marchegiani, I. Posner, “Leveraging the urban soundscape: Auditory perception for smart vehicles,” in IEEE Int. Conf. on Robotics and Automation (ICRA), Singapore, 2017, pp. 6547-6554.

214 D. Carmel, A. Yeshurun and Y. Moshe, “Detection of alarm sounds in noisy environments,” in 25th European Signal Processing Conference (EUSIPCO), Kos, 2017, pp. 1839-1843

215 L. Marchegiani, and P. Newman, “Listening for Sirens: Locating and Classifying Acoustic Alarms in City Scenes,” ArXiv, abs/1810.04989, 2018

216 Nasir, S. Z. (2012, October 4). XBee Arduino Interfacing - The Engineering Projects. Retrieved February 25, 2023, from www.theengineeringprojects.com website: https://www.theengineeringprojects.com/2012/10/xbee-arduino-interfacing.html

217 amazon web services - Handle an S3 “miss” with a lambda using serverless framework. Retrieved February 25, 2023, from Stack Overflow website: https://stackoverflow.com/questions/70780069/handle-an-s3-miss-with-a-lambda-using-serverless-framework

218 Mittal, U., & Chawla, P. (2023). Acoustic Based Emergency Vehicle Detection Using Ensemble of deep Learning Models. Procedia Computer Science, 218, 227-234.

219 Mittal, U., & Chawla, P. (2022, November). Detection of Emergency Vehicles Using Radio Frequency Identification (RFID). In International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi Petronas, Malaysia (pp. 271-278). Singapore: Springer Nature Singapore.

220 Mittal, U., & Chawla, P. (2022, October). Traffic Green Signal Optimization using Adaptive Neuro-Fuzzy Inference System on Coordinated Intersections. In 2022 10th

International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-7). IEEE.

221 Mittal, U., & Chawla, P. (2022). Vehicle detection and traffic density estimation using ensemble of deep learning models. Multimedia Tools and Applications, 1-23.