Smart Traffic Crossing and Advance Parking Management System using IoT

M.Tech Thesis

Submitted in partial fulfillment of
the requirements for the award of the degree
of
Master of Technology
in
Department of Software Engineering
submitted by

Upendra Singh (23/SWE/11)

under the guidance of

Dr. Sanjay Patidar
Assistant Professor
Department of Software Engineering

DEPTT. OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY, DELHI
June 2025

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I am very thankful to **Dr Sanjay patidar** (Assistant Professor, Department of Software Engineering) and all the Department of Software Engineering faculty members at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing me with the laboratories, infrastructure, testing facilities, and environment, which allowed me to work without obstructions.

I would also like to express my appreciation for the support provided to me by the lab assistants, seniors, and our peer group, who aided me with all the knowledge they had regarding various topics.

Upendra Singh

(23/SWE/11)

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I hereby declare that the M.Tech Thesis entitled **Smart Traffic Crossing and Advance Parking Management System using IoT**, which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master Of Technology (Software Engineering) is a bonafide report of M.Tech Thesis carried out by me. The material contained in the thesis has not been submitted to any university or institution for the award of any degree.

Date: 19 June 2025

Place: New Delhi

Upendra Singh

23/SWE/11

This is to certify that the student has incorporated all the corrections suggested by the examiner in the thesis and the statement made by the candidate is correct to the best of our knowledge. .

Signature of Supervisor **Signature of External Examiner**

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that M.Tech Thesis entitled **Smart Traffic Crossing and Advance Parking Management System using IoT** which is submitted by Upendra Singh, Roll No - 23/SWE/11, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree Master Of Technology (Software Engineering) is a record of the candidate work carried out by him under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: 19 June 2025

Place: New Delhi

Dr. Sanjay Patidar
Assistant Professor
Department of Software Engineering
Delhi Technological University

ABSTRACT

IoT and AI are in the greatest demand these days and are likely to have the dominant future. Smart cities are being demanded more and more, and IoT is one of the major facilitators towards making this vision a reality. IoT is being heavily developed so that efficient urban infrastructure can be built. Cities are plagued with several issues, such as inadequate parking area, risk of transport, and constant traffic jams. The tempo of contemporary life requires easy and efficient transport, which the present infrastructure cannot supply. Passengers are thus compelled to suffer from extended delays caused by traffic congestion even when traveling solo, which equates to excessive fuel usage and time wastage. Parking has also emerged as an issue of utmost concern for the majority of automobile owners. In response to these problems, we introduce an IoT solution. The thesis outlines an algorithm that will decongest traffic and improve the parking system. This entails a license number plate (LNP) recognition algorithm based on grayscale image processing and morphological operations for optimal identification. We also incorporate cutting-edge IoT sensor technology and AI, in the shape of a fuzzy logic system, to regulate traffic. Cars are regulated in time slots to facilitate smooth traffic. Our system also includes a cloud-based smart parking system that identifies empty spaces through IoT modules. We can now integrate AI-driven methods to track over-speed as traffic speed, road accidents, and automated emergency reporting on traffic. Data protection and security initiatives further render our IoT-based platform a deserving vehicle for intelligent traffic control and parking in modern urban environments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Generic Approach

Today, the internet has become part of the daily life of everyone, and the device that receives the internet is referred to as the Internet of Things (IoT). IoT facilitates devices in device-to-device communication, connecting the physical and digital worlds by sharing and transferring information without any interruption.

It was when electronic devices or modules started to recognize and talk to one another that they came up with the term IoT. Life became simple living our day-to-day life with the advent of IoT since we are able to monitor, control, and automate quite a lot of things through internet-enabled devices. The devices include household devices like air conditioners, refrigerators, microwaves, and air purifiers—most of which are now transforming to become smart systems with embedded sensing and smart capabilities, like robots [1].

IoT produces large quantities of data, and this is a normal practice to save and process it in the cloud. In the future, IoT and cloud computing will work together to provide even more effective systems. By using the cloud together with IoT, individuals will remotely monitor and control devices, resulting in wiser and more responsive systems. Moreover, adding or deleting devices (modules) will also make it more scalable and personalized, and therefore more sophisticated and responsive solutions will be provided. [2]

1.1.1 Concept of IoT-based smart parking system(SPSI)

By taking services of SPSI, we are able to book parking lot from your home or other places where you have internet access, from there user can save time and don't miss taking parking due to noise. So, we save petrol/diesel/CNG etc., indirectly saving money.

The major goal of the system is to provide efficient parking services and convenient parking lots for drivers, and be an operating service platform too. It acts as a communication between providers and service and serves the unique requirements of drivers. Driver congestion will be lowered, and the overall traffic control will be boosted by the system. The control signals are designed in a particular way so that drivers can easily spot and pick available parking spaces. [3].

A better license plate recognition algorithm is implemented to modify the parking record efficiently. Additionally, an approach has been suggested to find the shortest entry and exit routes in massive parking lots to improve the driver experience as well as decrease the waiting time.

1.1.2 Traffic Management System

Fuzzy logic with IoT, is utilized to control traffic effectively. Fuzzy logic can track the motion of the vehicle on the complete road so that dynamic time control can be handled with less traffic congestion as shown in Figure 1.1. Traffic lights automatically change according to the actual traffic condition—when there's no vehicle on one side, the duration for that traffic side reduces and is transferred to other sides having more traffic. [4]

This smart measure radically enhances traffic control, reducing hold-ups as well as fuel conservation, reducing travel time, and generally enhancing road performance.

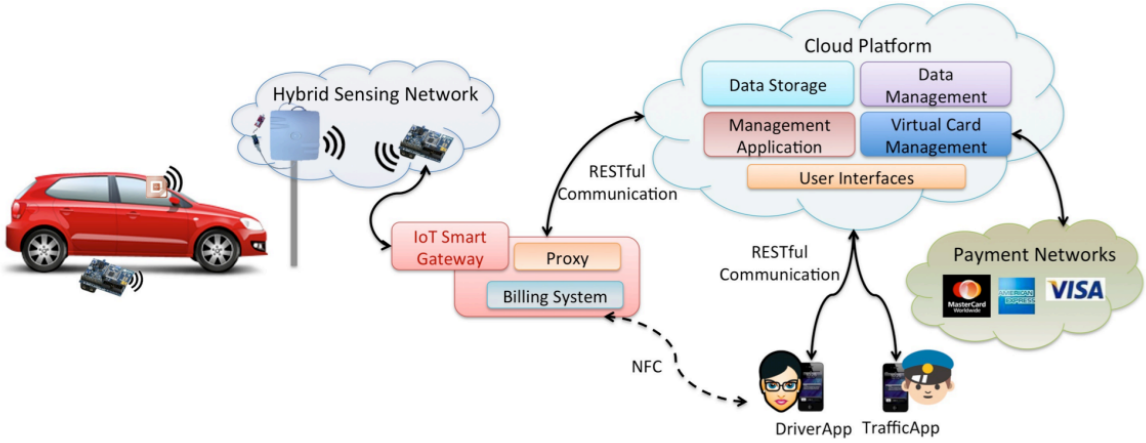


Figure 1.1: Smart Parking System

1.2 Motivation

Today, individuals want things to be convenient and easy, and that is being made possible by IoT. As digitalization increases, conventional locations are being transformed—houses are being converted into smart homes, and cities are becoming smart cities. The internet is a major contributor to making this achievable because connectivity is the key to making things smarter and more effective.

When IoT is combined with Artificial Intelligence (AI), the systems of each day become intelligent, and they can act on their own and effectively. Big data technologies manage the massive amount of information created by the devices of IoT, whereas cloud computing maintains this infrastructure by providing a higher level of accessibility and usability. [1, 4]

Today, various projects are being conducted for building smart cities and smart houses. Our mission is to do our part to this revolution by developing an intelligent system for traffic and parking. Traffic jams and parking become crazy issues nowadays. For instance, ineffective management of parking spaces always leads to redundant traffic and fuel wastage. Likewise, traffic management needs to deal with safety issues and time wastage.

IoT is a solution to all these issues as it makes traffic and parking systems smarter and more efficient than ever. Our thesis is inspired by all these practical problems, and the rest of the sections discuss the work done by us to justify and enhance these systems. [5]

- Algorithm for detection of number plates of vehicle.
- For the creation of intelligent parking system in IoT and cloud database.
- For making traffic easy IoT and Fuzzy logic is used.

1.3 Previous Work

1.3.1 Advanced parking system

There are numerous works dominated in recent times on smart parking system. In our project IoT used with wireless sensor network for effective management of parking system and effectively. We can sense parking data through the use of wireless sensor network and can employ into our parking system. Another developed algorithm i.e. LNP also used for monitoring of parking lot using vehicle detection. Now we can save parking space and parking time in parking system with this new algorithm. Some of the earlier similar research studies are as follows:

Cyber-Physical System (CPS)

Cyber-Physical Systems (CPS) have been a popular area of research, offering effective solutions to complex problems, as seen in studies like. CPS can manage and coordinate various operations by controlling and interacting with connected systems. It provides insight into how physical components communicate and engage with the real world.

In today's fast-paced environment, CPS faces many challenges, which are discussed in. One of the key areas of CPS research is transportation , along with security, environmental monitoring, and industrial applications. [5]

According to earlier surveys in the U.S., transportation accounted for nearly 30 percent of total energy consumption in 2008—a significant rise of about 90 percent over 35 years . If efforts are made to reduce energy use in transport, considerable energy can be saved. A primary goal of CPS is to lower transport-related energy consumption, even enabling solutions without drivers. As highlighted in , one of the most effective ways to reduce energy use in transportation also considers environmental impacts.

CPS focuses on real-world physical systems, specifically on monitoring, controlling, and interacting with them . The major challenges faced by CPS include:

- CPS object framework.
- Organizational reliability and robustness.
- Hybrid system management.
- Computational capabilities.
- Online integration.
- Knowledge acquisition and learning.
- Paradigm- and sensor-based interaction.
- System verification and validation.

Cyber-Physical Systems or Online Materialistic Systems (OMS) are vital in ensuring secure and reliable navigation. For example, air transportation also relies on such networks. In cases of adverse weather or airspace congestion, CPS can offer adaptive routing through a flexible gateway and maintain constant connectivity. As discussed in , researchers have highlighted the potential of mobile networks to evolve using transportation-focused CPS technologies.

VANET - Based smart parking Vehicle Ad-Hoc Network, or VANET, is another successful solution to evolving parking systems. In VANET, customers are able to make reservations on parking spaces rapidly and effectively using an interactive system that offers instant feedback. Wireless Sensor Networks (WSN) were previously used for making reservations, taking accounts, and precise space assignment for vehicles. Although WSN worked, it operates in a different way compared to VANET. [6]

Using VANET, Rongxing Lu et al. proposed the Smart PARKing (SPARK) scheme, which accurately identifies available parking spots and allows real-time vehicle tracking. This not only improves convenience but also enhances security by enabling continuous monitoring. The method for locating new parking lots is further discussed in.

Parking lot Detection Jake Reisdorff suggested an effective method to identify empty parking spaces in. The method includes a site where pictures of cars are uploaded. With image processing algorithms and the uploaded content, empty as well as full parking spaces can be identified. [7] The method begins with a high-definition camera taking the pictures, which are sent to a web server. From these snapshots, the system can identify available parking spaces and assign them to new clients. The system does this with the aim of reserving the parking spaces more quickly because the detection is highly fast and effective.

IoT based smart parking Khanna and Anand suggested an application-based parking space booking scheme. The solution is limited to fewer spaces and is unstable and non-user-friendly. The app facilitates booking and payment for parking spaces with debit/-credit cards, e-wallets, or cash. Although it is helpful for parking booking in small areas, it does not handle matters like rapid identification of free parking spaces, security, or planning entry and exit routes in large parking spaces. citeabinaya2017,chianyung2022cross

1.3.2 LNP detection related work

Automatic object detection is a significant challenge today, and numerous studies are being conducted on this topic. Several techniques for license plate detection include methods such as neural networks, color features, edge detection, morphological operations, and image transformations. [8]

The License Number Plate (LNP) is unique for every vehicle, and when combined with IoT , it offers an effective method for vehicle recognition. In this approach, the camera focuses on areas with high energy and frequency. The captured image can either be of the entire vehicle or just the license plate, which helps save storage space in urban city surveillance systems.

In vehicle license plate recognition systems , the license plate image is cropped from the full vehicle image, which is a key advantage of this detection method. Another technique involves vertical edge detection , where the two vertical edges of the image are

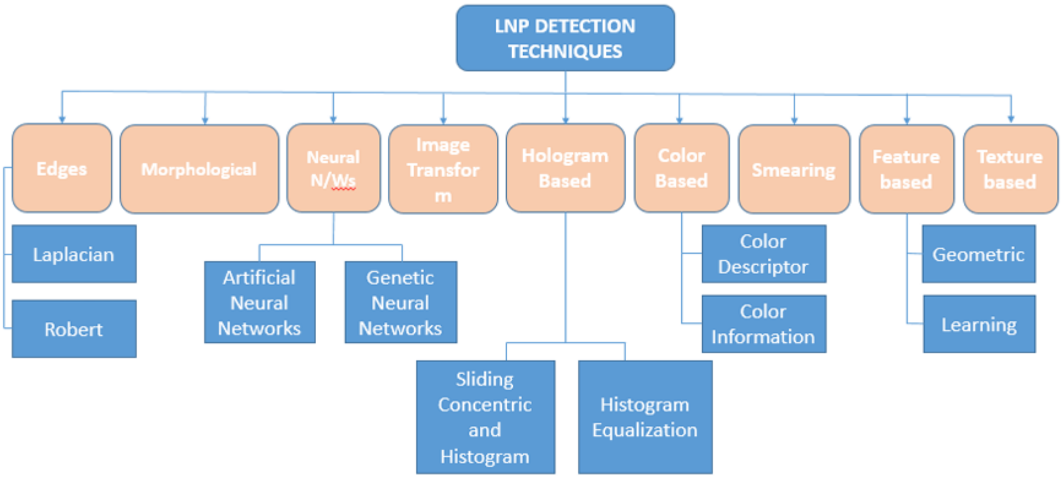


Figure 1.2: Diffrent LNP detection technique

identified first, followed by detection of the four corners, as shown in Figure 1.2. This method is highly effective for LNP detection .

The morphological-based LNP method is useful for capturing high-contrast images, enhancing accuracy. Additionally, a color-based method for Vehicle License Plate Recognition (VLPR) detects the license plate by focusing on a specific color from various image colors. [9]

One of the most effective methods for LNP detection is based on neural networks . This process involves detecting the plate region, segmenting characters, and identifying each character.

1.3.3 Corresponding strategy for smart management of traffic system

To reduce traffic congestion, researchers have proposed several approaches, and ongoing studies continue to explore new solutions. One such method is video-based analysis, which focuses on traffic parameters to identify the causes of congestion .

Vehicle detection can be categorized into two techniques: stereo vision and monocular vision. Both methods rely on motion and appearance detection . Stereo vision uses motion detection, recognizing movement by capturing 3D images. On the other hand, monocular vision relies on appearance detection, primarily using 2D images to identify vehicles.

Machine learning also plays a crucial role in this process, helping to accurately recognize images, track object movement, and localize the license number plate (LNP).

1.4 Problem Statement

We had debated earlier research based on parking and traffic systems for providing smartness to cities. The proposed here parking system algorithm is very efficient based on the conversion of normal images into greyscale, along with detecting distortion, noise filtering, and performing morphological operations. This approach offers an extremely accurate solution for identifying vacant parking spaces. Further, tracking of the cars inside the parking lot is also feasible due to all the information being stored on the cloud via the license plate recognition algorithm. Billing and other tasks are thus simplified. With the assistance of IoT and Fuzzy Logic integration, we can measure the traffic volume on both sides of the road and manage traffic lights accordingly. Not only does this save time, but it also decreases transportation energy consumption within the city.

1.5 Scope of the Thesis

We implement a cloud-based system in our thesis where we keep all the parking records in a database as a large database that keeps all the parking spaces of the city. Drivers can reserve parking slots from any rural or urban area with the help of sensors and IoT. It proves to be of great use to the drivers in searching for available parking slots.

License plate detection system, augmented by a high-resolution camera, aid in the allocation of parking space properly. The shortest path algorithm is also utilized to navigate cars from the entry point to the parking space and then to the exit point.

For traffic management, we use IoT and AI (Fuzzy Logic) to provide a realistic solution to actual traffic issues. In the current era of IoT and AI, the provided solution is focused on building a smart city by improving traffic and parking management. Through computerized algorithms of vehicle number plate recognition (converting normal images into greyscale) and integration with AI, the system improves efficiency, saves time, fuel, and resources and simplifies life for humans.

Chapter 2

LITERATURE REVIEW

2.1 IoT-Cloud Integration

By making a few advancements, we can transform our city into a smart city. Currently, parking and traffic systems are major issues, and our goal is to improve or provide better solutions for them.

As the population continues to grow, so does the number of vehicles, which inevitably leads to problems with traffic and parking systems. While the government can allocate parking spaces, our solution focuses on using that space more efficiently. With an IoT-based application, we can reduce time, energy, and costs. Additionally, AI and IoT can help manage traffic load, energy consumption, and time more effectively. [10]

Our parking system requires sensors to detect available parking spaces and record occupied spots. All data regarding vacant and filled spaces will be stored in a cloud database. With this information, we can make reservations and track availability through sensors.

Since IoT generates vast amounts of data, storing this "big data" posed a challenge for IT companies. Cloud computing has solved this problem, making data storage easier. IoT enables devices to interact through the internet, and cloud computing provides the infrastructure to store the data. Some key benefits of combining IoT and cloud computing are:

- Handling large volumes of data.
- Ensuring security and privacy.
- Providing remote processing power.
- Enabling inter-device communication.
- Lowering entry barriers for hosting providers.

2.2 Fuzzy Logic

"Fuzzy" is vague or indistinct, and fuzzy logic is a reasoning approach that mimics human decision-making. It is a decision-making approach that considers all possibilities. Similar to mathematical algebra and Boolean logic, fuzzy logic considers everything. [11] Unlike binary logic, fuzzy systems operate with inexact inputs (e.g., "high traffic density" or "low queue length") to optimize signal timings. Implemented fuzzy rules in a bias towards lanes with higher vehicle queues, decreasing the average waiting time by 30%. Membership functions for input parameters such as traffic flow (F) and queue length (W) are shown with the help of Eq. (2.1):

$$\mu_{\text{Medium}}(F) = \begin{cases} \frac{F-4}{8} & \text{if } 4 \leq F < 12, \\ \frac{16-F}{4} & \text{if } 12 \leq F \leq 16, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

2.3 Wireless Senser Network (WSN)

In Wireless Sensor Networks (WSN)(Fig 2.1), sensors are used to collect data, which is transmitted to various endpoints. This data is gathered and sent to a central node, known as the destination, using wireless connections and other communication methods. The Figure 2.1 below illustrates several applications of WSN, including transportation, industrial automation, agriculture, and security. [11]

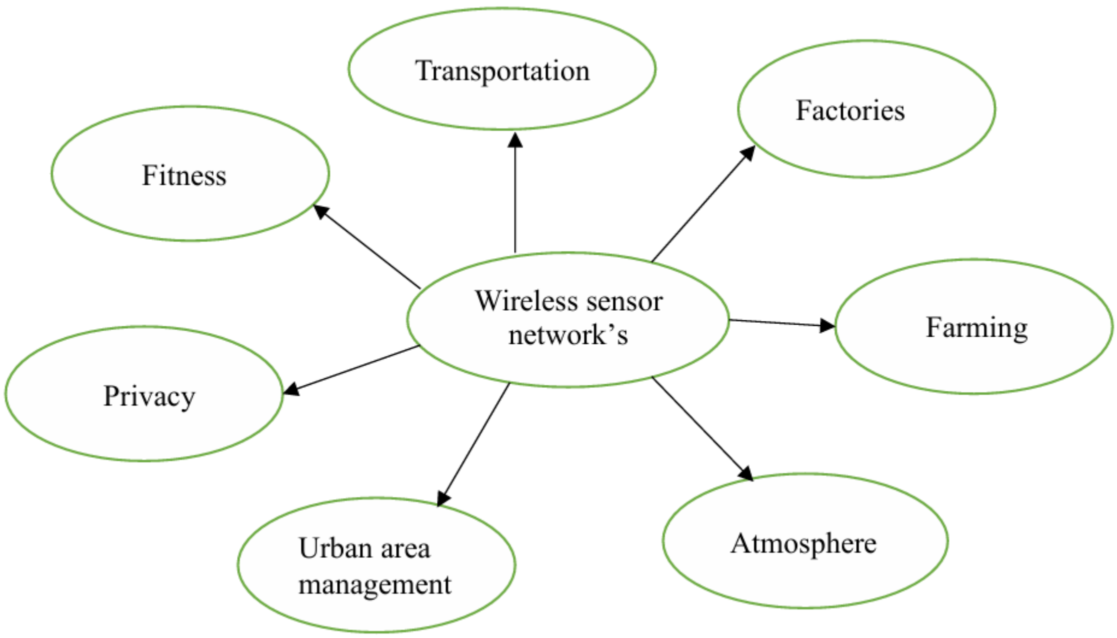


Figure 2.1: Wireless Sensor Network Application

2.4 Exit Algorithm in big parking

We have proposed a solution for efficient exit from large parking areas. In cases where there is a single entry point but multiple exit routes, we addressed the challenge by implementing Dijkstra's algorithm. Each parking lane is equipped with an LED light, and once the algorithm determines the shortest path, the corresponding light will blink automatically. Drivers can then follow the blinking light to exit quickly and efficiently. [12]

2.5 Morphologic

Morphological operations are amongst the image processing methods, which process shape and transformation of an image. Since morphological operations process structuring elements over an input image, they therefore provide an output image of equal size. We have utilized morphological operations within this thesis in the same way, whose output is a shape transformation version of the input image, which is operated upon under a shape comparison and scale with respect to surrounding elements. The two strongest of the two techniques employed are dilation (addition) and erosion (deletion). Dilation fills object boundary pixels with the highest pixel value of neighboring pixels, while erosion removes object boundary pixels based on the lowest pixel value, depending on the neighbor input pixels [13]

Chapter 3

Proposed Work

3.1 Automatic object identification by IoT-Improvement

By incorporating our approach into the existing image-based algorithm , the following improvements were observed:

Distortion Minimization:

Image distortion, often caused by the camera lens, needs to be corrected for accurate vehicle number plate detection. Since number plates are typically black, capturing this portion can result in higher distortion. This happens due to blank spaces between the black digits, which are often lost due to lens effects .The algorithm applies radial distortion correction to restore geometric accuracy. [8, 9]

Noise Minimization:

To reduce noise and smooth image details, the Gaussian smoothing method is applied using the Gaussian function. This technique helps blur the image and minimize unwanted visual noise. The filtered image is calculated using the Eq. (3.1):

$$F(p) = \frac{1}{H_p} \sum_{q \in \Omega} \mathcal{G}_\alpha(\|p - q\|) \cdot \mathcal{G}_\beta(\|J(p) - J(q)\|) \cdot J(q) \tag{3.1}$$

Where:

- $F(p)$ denotes the output intensity at pixel location p after bilateral filtering.
- $J(q)$ is the intensity value of the input image J at neighboring pixel q.
- Ω represents the local window or neighborhood centered at pixel p.
- $\mathcal{G}_\alpha(\|p - q\|)$ is the spatial Gaussian kernel, which decreases the weight of pixels farther from p; the parameter α controls spatial spread.
- $\mathcal{G}_\beta(\|J(p) - J(q)\|)$ is the range (intensity) Gaussian kernel, assigning lower weights to pixels whose intensities differ significantly from $J(p)$; the parameter β governs sensitivity to intensity differences.
- H_p is a normalization factor defined as in Eq. (3.2):

$$H_p = \sum_{q\in\Omega} \mathcal{G}_\alpha(\|p-q\|)\cdot\mathcal{G}_\beta(\|J(p)-J(q)\|), \tag{3.2}$$

[14]
which ensures that the filtered output $F(p)$ remains within the valid range of intensity values.

Grey scale convergence:

The formula used for grayscale conversion is:

$$Z = \frac{3.06R + 5.85G + 1.09B}{10}$$

where Z denotes the value of grayscale pixels, R denotes the red color, G denotes the green color, and B denotes the blue color for every pixel. [3]

Morphologic operation:

Dilation and erosion are fundamental processes in morphological image processing. Dilation Operation: This operation enlarges the boundaries of objects in an image, causing nearby features to merge or connect.

Below are Eq. (3.3) and (3.4) for operation dilation and erosion:

$$Z(i,j) = \max_{(u,v)\in K} [A(i-u, j-v) + K(u,v) \ : \ (i-u, j-v) \in A] \tag{3.3}$$

In this transformation, known as dilation, the pixel at position (i,j) in the output image Z is derived as the maximum sum between the input image A and the structuring element K, shifted across all valid overlaps. [11]

$$Z(i,j) = \min_{(u,v)\in K} [A(i-u, j-v) - K(u,v) \ : \ (i-u, j-v) \in A] \tag{3.4}$$

The erosion operation determines the value at (i,j) in the output Z as the smallest result of subtracting the structuring element K from the input A, over every overlapping position where the condition holds.

3.2 Algorithm for detecting traffic loads.

We have created a unique algorithm incorporating fuzzy logic and IoT-based motion sensing effectively to detect and manage traffic congestion. The algorithm considers the analysis of real-time traffic density from multiple directions. With dynamically adjusted traffic signal time depending on traffic load, it optimizes traffic flow and minimizes congestion effectively. The combination of fuzzy logic and motion sensors enables smart decision-making, and that's why the method is extremely effective in adaptive traffic light control systems as shown in Figure 3.1 and 3.2.

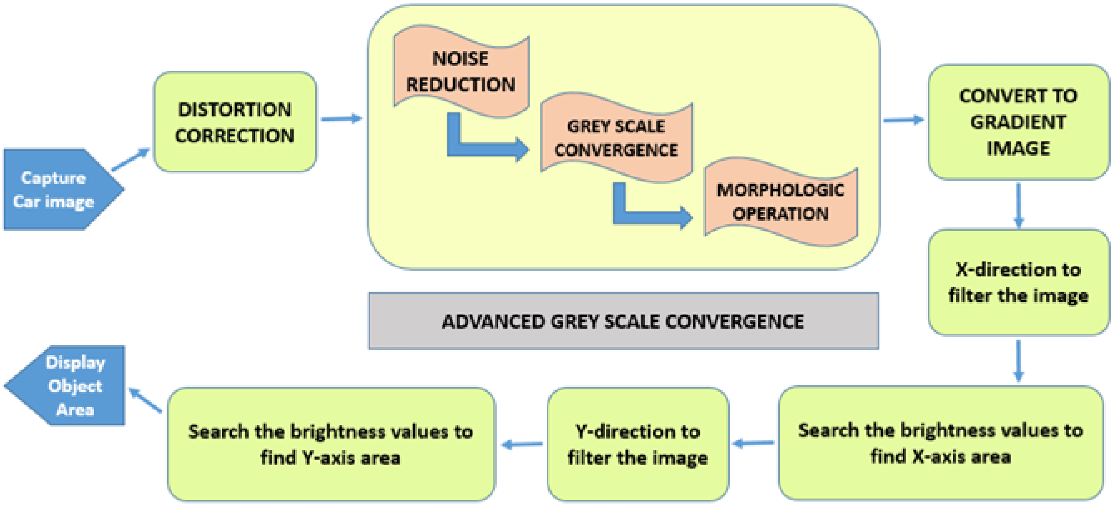


Figure 3.1: Automated detection of an object using IoT

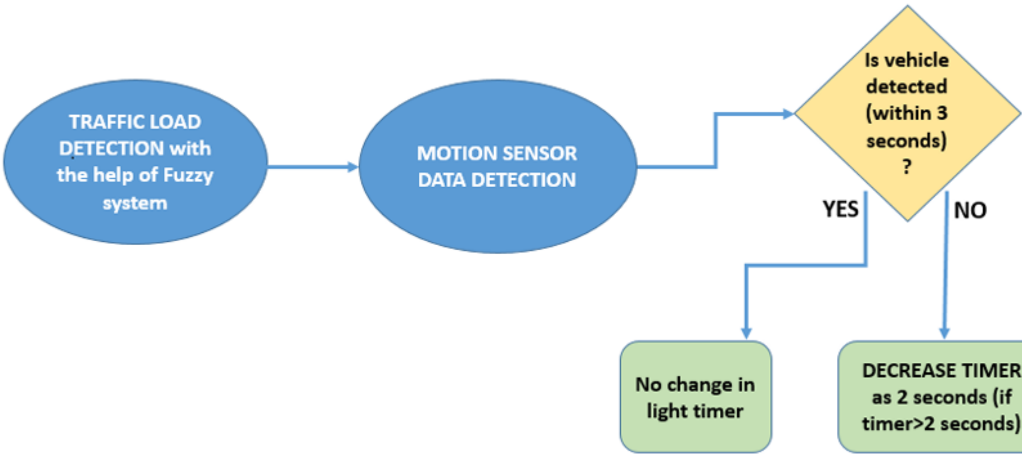


Figure 3.2: Advanced detection of load in traffic

3.3 Detection of object using IOT

In this piece, we suggested a system that makes it easier to find available parking spaces through a shortest path algorithm, which will take drivers directly to the closest vacant space without hassle. For even simpler application, the system also gives clear instructions on how to direct in picking the most effective detour around the parking lot. In addition, a license plate recognition algorithm has also been included to enable easy identification and tracing of cars for a safe and automated parking system. As shown in Figure 3.3

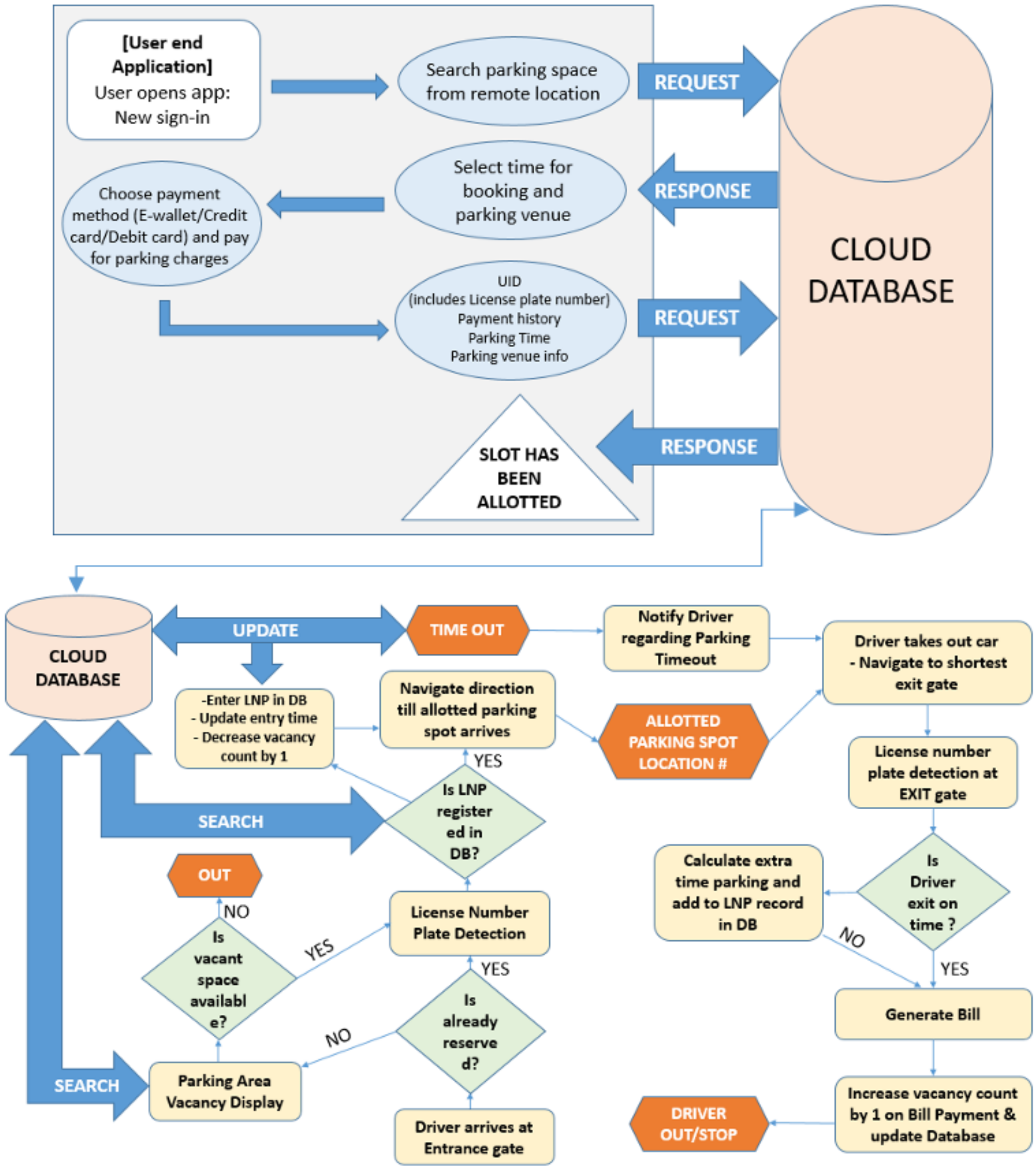


Figure 3.3: Advanced detection of load in traffic [13]

Chapter 4

Methodology

4.1 Improvement in License Plate Recognition Accuracy

In this section of the thesis, we present an enhanced version of the automatic license number plate detection method originally proposed by [13]. The previous approach was comparatively less efficient, particularly in terms of cloud storage usage. Our proposed solution addresses this limitation by capturing only the license plate image, rather than the entire vehicle, thereby reducing memory consumption in the cloud.

To convert the image to a gradient format, we applied the following four methods:

4.1.1 Distortion Adjustment

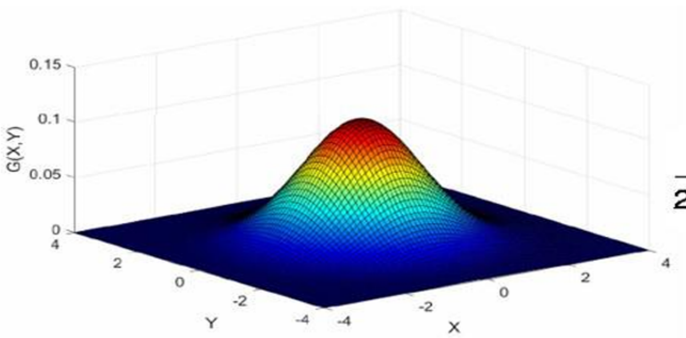
It is well known that images often suffer from distortion (as shown in Figure 4.1) due to the characteristics of camera lenses. To ensure accurate identification of license plate numbers, correcting this distortion should be a priority. In our case, significant distortion was observed when capturing images of license plates, primarily due to the type of lens used. Wide-angle lenses, in particular, tend to introduce greater distortion, which can negatively impact recognition accuracy. [14]

Figure 4.1: Example of distortion with different lenses

When an image is captured, it often appears curved or warped due to the inherent properties of the camera lens. To address this, radial distortion correction is applied to enhance the clarity and accuracy of the image. In our thesis, we specifically focused on correcting radial distortion by adjusting the coordinate ratios and calculating the intersection points of straight lines within the image. We employed an inverse distortion technique, allowing us to minimize further distortion by comparing the original distorted image with the corrected, distortion-free version. [15]

4.1.2 Denoising

Noise in an image can originate from various sources, such as the scanner circuitry, image sensor, or digital camera. It represents unwanted variations in the signal, similar to the interference picked up by an AM radio. Detecting and reducing noise in license number plate images is particularly important. To address this, we applied the Gaussian filtering technique to effectively minimize noise. Shown in Figure 4.2. [3].



$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 4.2: Gaussian noise reduction method
[3]

Gaussian give the formula and through the application of that formula image blur was was reduced. Using this method image pixel transformation we utilized

Eq. (4.1) shows the guassian method:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{4.1}$$

[16]

Here, x and y represent the spatial coordinates in the horizontal and vertical directions, respectively. The term σ denotes the standard deviation, which controls the spread of the Gaussian distribution applied for smoothing.

Average weighted values are entered in figure 4.2 which are the values of the neighboring pixels. If the distance between the original pixel and the neighboring pixel is greater then the weight will be low provide will be and if the distance is becoming smaller between original pixel and the neighboring pixel then assign weight will be greater. Gaussian methods reduced random noise with this method.

4.1.3 Operation of Morphologic

It is simply helpful in dividing the every elements and combining the scattered elements. For Dilation of Objects: Characters become disjointed on dilation of objects. Eq. (4.1) and (4.2) for erosion process and dilation are as follows:

$$S = (X \oplus Y)[k,l] = \max\{X[k-n, l-m] + Y[n,m] \mid (k-n, l-m) \subset X,\ [n,m] \subset Y\} \tag{4.2}$$

Operation of Erosion: In this operation objects are reduced by value in size as shown in Eq. (4.3)

$$S = (X \ominus Y)[k,l] = \min\{X[k-n, j-m] - Y[n,m] \mid (k-n, j-m) \subset X,\ [n,m] \subset Y\} \tag{4.3}$$

Technology Background	Environment	Light's Condition	Picture Quality	Color's Model	Accuracy
Based on color feature	More dependent on proper location of number plate	Illumination should not be skewed	Good in day & night images but poor in non-uniform lighting	Binary images, RGB > HIS	80.00%
Based on Geometry	Uniform	Illumination should be uniform	More contrast; Both binary and digital images	Greyscale	82.00%
Histogram	Uniform	Better in varied weather	Best for poor illumination & different lighting conditions; Greyscale 640×480 pixels; digital and binary image	HIS	82.50%
Based on Edge	For high quality results, sharp edges required	NA	More sensitive to noise and light intensity; 352×288 images, 30f/sec video	Greyscale	91.00%
Neural Network	Good for uniform background; bad for non-uniform background	Good for uniform illumination	Additional light in day and reflective material at night; Digital image: 320×240, Optimized: 1600×1200, aspect ratio 4:3	RGB to Greyscale	94.00%
Morphology	Better for multiple backgrounds	Good performance in almost every weather	Good for low-light conditions; Static image	Greyscale	96.00%

Table 4.1: Comparison of Technologies Based on Various Image Processing Conditions

As shown in Table 4.1, various image processing techniques are compared based on environmental conditions, lighting, picture quality, color models, and accuracy.

4.2 Optimized Algorithm for Traffic Load Analysis

In this traffic congestion detection system, AI and IoT have been implemented with a complex algorithm. The traffic light timing is dynamically controlled with this algorithm to minimize the jam. For instance, take a four-way intersection: whenever one side receives a green signal, the cars on the other three sides must wait for 20 seconds for their turn. But in the absence of any car on the route cleared, the light goes yellow for 3 seconds. In this interval, the system examines the load of the other three sides and provides green light to the most-loaded side. This method can reduce waiting time up to 17 seconds. Fuzzy logic is utilized in order to measure the traffic load of both sides of the intersection and allow for more precise and robust load verification as shown in Figure 4.3. In fuzzy, absolute true is 1 and absolute false is 0; intermediate values such as 0.9 or 0.8 express degrees of truth, and values such as 0.1 or 0.2 express degrees of falsehood. The fuzzy membership function is employed to determine when the green light should be extended or shortened. This is done efficiently so that traffic load on both sides of the roads is minimized by optimizing the flow of vehicles during peak hours. Sensors were also placed at zebra crossings so that proper detection can be made.

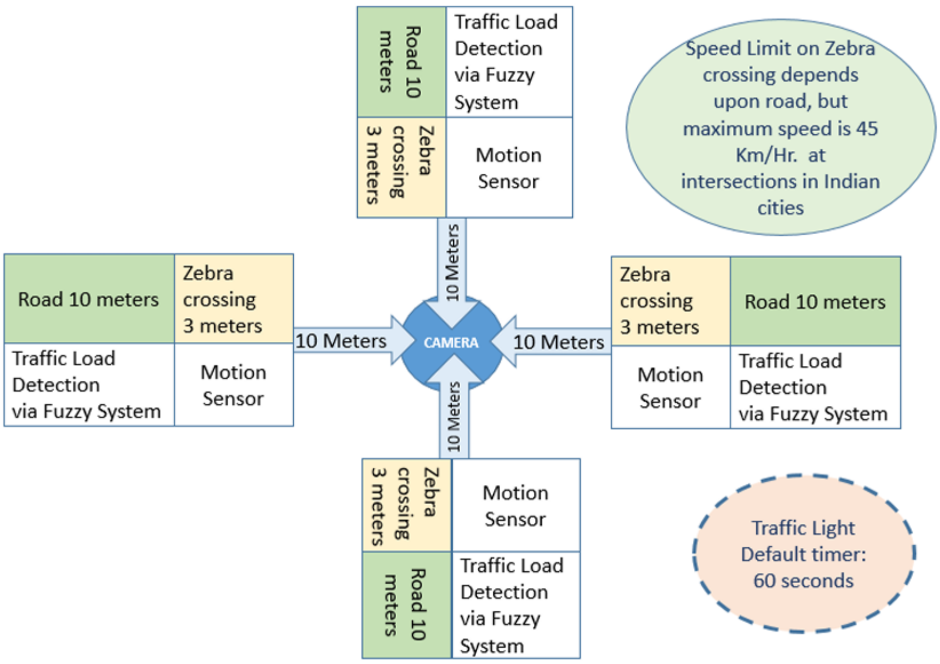


Figure 4.3: IoT and Fuzzy logic for traffic load reduction

For instance, as mentioned below diagram 4.3: We obtained the highest possible speed across the crossing of 35 km/h. At this speed, it would take approximately 0.31 seconds to cross over a 3-meter wide zebra crossing. At the lowest possible speed of 4 km/h, it would take approximately 2.88 seconds to cross over the same zebra crossing.

In our system under consideration, we have given a waiting time of 4 seconds. That is, if there is no motion detected by the sensor, up to 3 seconds waiting time will be given to the traffic light. We have also given an additional 5 seconds of signal time, which will be reallocated to the direction that has the maximum traffic load out of the other directions.

4.3 Smart parking system using IoT

We have implemented License number plate detection technique in all entry points and exits and upgraded parking system with improved tracking system of the vehicles as depicted by Figure 4.4.

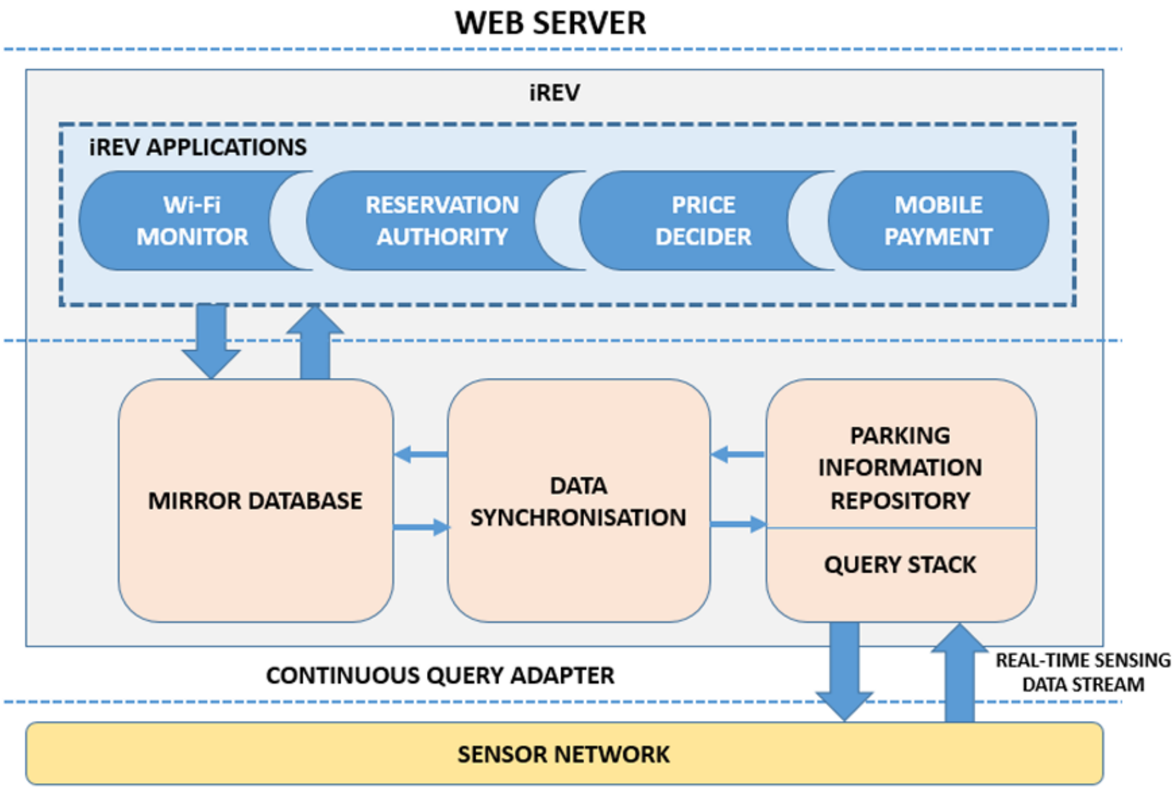


Figure 4.4: Smart parking system architecture
[17]

The parking system introduced in this work integrates several components for efficient vehicle management. Firstly, the license plate recognition module, as described in earlier sections, is used to uniquely identify incoming vehicles.

For vehicle tracking within the parking facility, we implement Dijkstra's shortest path algorithm to determine the most efficient route from entry to an available parking space, and subsequently from the parked position to the nearest exit. [18]

To facilitate intuitive navigation, LED indicators are installed along both sides of the internal roadways. These lights dynamically guide drivers along the optimal path, ensuring smooth entry and exit from the parking lot.

Chapter 5

Results and Experiments

5.1 Tools used

The traffic and parking smart system was simulated based on user experience observations. We employed several patterns of user data to simulate various real-world situations. The patterns were then built into the simulation in various ways, enabling us to examine and test the performance of the system under various scenarios of traffic and parking activities..

In our project, implementation of the shortest path algorithm was done using Visual Studio and Java (JDK), whereas MySQL was utilized for database operations like retrieving and storing vehicle and reservation data. Microsoft Excel was also utilized to gather and manage sensor-based data because it is easy and simple to represent data. Additionally, MATLAB was utilized to simulate the smart parking system and identify traffic behavior under different conditions. All the tools chosen are easy to use and provide high interactivity, therefore appropriate for development as well as simulation.

5.2 Analysis using data set

5.2.1 Using dataset for traffic management

Roadside and zebra crossing used car detectors. It is of very good use. at a point where 4 roads meet. Traffic load detection is shwon in Figure 5.1 and 5.2.

By using detectors, we are able to estimate our time interval of crossing. Let V be the fuzzy variable and N be the vehicle number that will be in queue and W as extra waiting time for extra vehicles as shown in Table 5.1 and 5.2.

Membership Function	**Phase**	**Few**	**Small**	**Medium**	**Many**
V	Green	-5 to 5	0 to 7	5 to 13	7 to 11
N	Red	-5 to 5	0 to 7	5 to 13	7 to 11

Table 5.1: Membership function of the input data at various stages

Now we can determine the vale of X with the help of above table.

Duration of green light in our fuzzy controller was increased by up to 5 seconds depending on traffic.

Fuzzy Rule Base:

Membership Function	Zero	Short	Medium	Long
X	0 to 5	5 to 10	10 to 15	15 to 20

Table 5.2: Membership function for output data

After the minimum green time (5 seconds), the following fuzzy conditions are evaluated:

```
if (V == small) and (N in [many, few, medium, small]):
    X = zero
elif (V == many) and (N in [few, many]):
    X = short
elif (V == many) and (N in [small, medium]):
    X = zero
elif (V == medium) and (N in [many, small]):
    X = medium
elif (V == medium) and (N in [few, many]):
    X = short
elif (V == few) and (N == many):
    X = long
elif (V == few) and (N in [medium, small]):
    X = medium
elif (V == small) and (N == few):
    X = short
```

Listing 5.1: Fuzzy Rule Logic

Initial Traffic Signal Timing Adjustment:
Begin by applying the first timing decision, which increases the green signal by 5 seconds. The fuzzy inference rules for decision variable T are evaluated as follows:

```
# Start with a 5-second green light extension

if (V == medium) and (N in [many, few, small, medium]):
    X = zero
elif (V == few) and (N in [many, few]):
    X = zero
elif (V == few) and (N in [small, medium]):
    X = zero
elif (V == many) and (N in [few, small]):
    X = short
elif (V == many) and (N in [many, medium]):
    X = zero
elif (V == small) and (N == many):
    X = medium
elif (V == small) and (N in [medium, few]):
    X = short
elif (V == medium) and (N == small):
    X = zero
```

Listing 5.2: Fuzzy Decision Rules for Initial Green Time Adjustment

As seen in the second expansion of the green light, there is no need for a further extension, and an IoT-based motion sensor can be used to handle the signal more efficiently.

For example : 35 km/h is the maximum speed at the junction, and at this speed it would take about 0.31 seconds to cross for 3 meters length of the zebra crossing. In an even extreme scenario, where it's at a very slow speed of 4 km/h, it would take about .

We used a waiting time of 3 seconds in our thesis. If there is no movement sensed by the motion sensor within this interval, then the system will automatically toggle the green signal to the next suitable direction. Our main goal for our solution is to allow as many cars as possible to cross the intersection in a specified amount of time.

The weight, W(X), is determined based on the number of vehicles queuing while as shown in Table 5.3.

Label	**Range**
Zero (Z)	-5 to 5
Low (L)	0 to 7
Medium (M)	5 to 14
High (H)	9 to 18

Table 5.3: Fuzzy membership labels for variable $W(X)$

This is used to give priority to the signal at times of maximum demand. For instance, priority 4 can be given for 'right of way' as shown in Table 5.4.

M(Index)	**Direction**
$W(1)$	East
$W(2)$	West
$W(3)$	South
$W(4)$	North

Table 5.4: Directional labels corresponding to index values of M

Fuzzy Rule Set for Determining the Next Green Phase:

```
# Decision rules for next green signal based on fuzzy input levels

if W(2) == 'low':
    Next_Phase = 2
elif W(2) == 'medium' and W(3) == 'high' and W(1) == 'high':
    Next_Phase = 2
elif W(2) == 'high' and W(3) == 'high' and W(1) == 'high':
    Next_Phase = 2
elif W(2) != 'low' and W(3) == 'low' and W(1) in ['high', 'medium', 'low']:
    Next_Phase = 3
elif W(2) == 'high' and W(3) == 'medium' and W(1) != 'low':
    Next_Phase = 3
elif W(2) != 'low' and W(3) != 'low' and W(1) == 'low':
    Next_Phase = 1
elif W(2) == 'high' and W(3) == 'high' and W(1) == 'medium':
    Next_Phase = 'E'  # Possibly East or Exit phase
```

Listing 5.3: Traffic Signal Phase Selection Logic

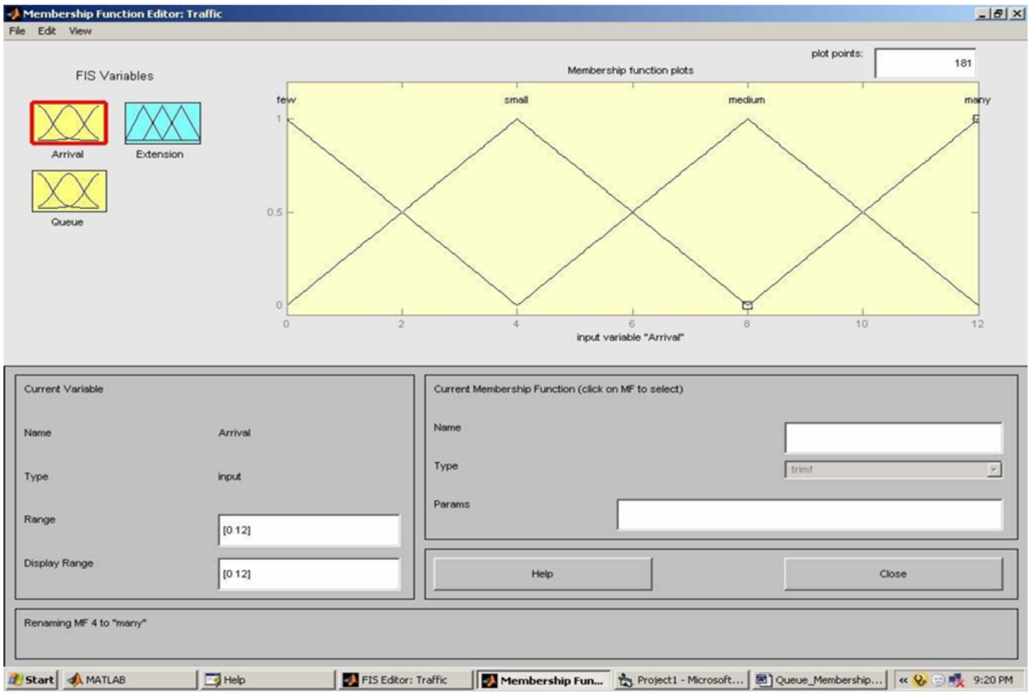


Figure 5.1: IoT and Fuzzy logic for traffic load reduction

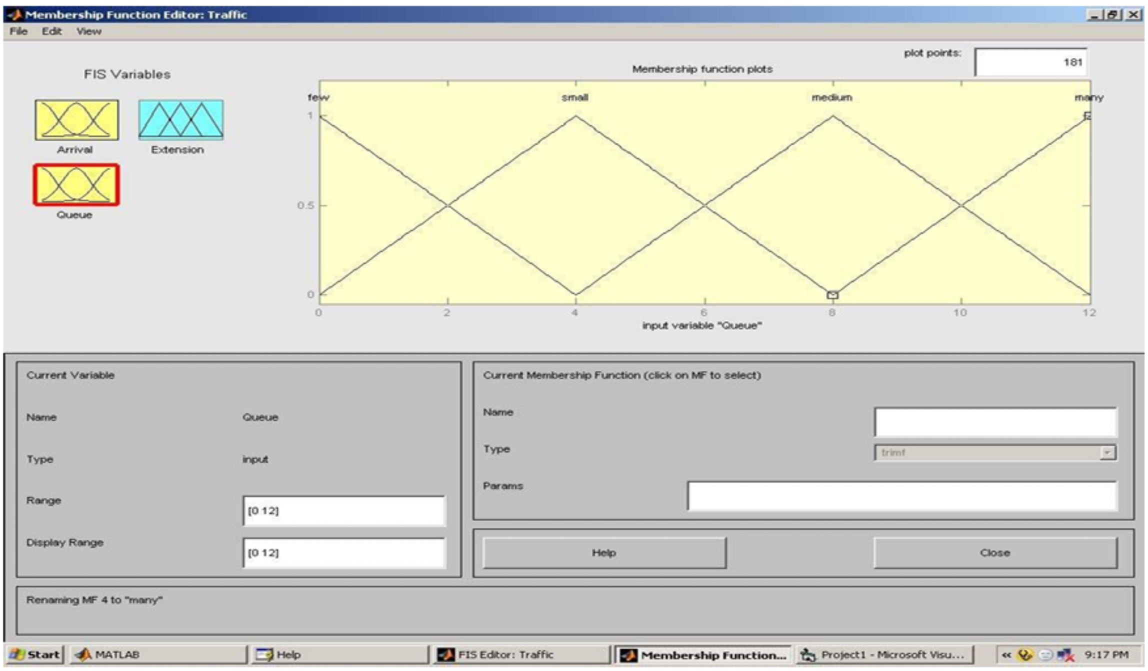


Figure 5.2: Fuzzy logic on queue variable

5.2.2 Incorporation of Dataset in the Design of an IoT Smart Parking System

1. **Nearest available parking system**

   The suggested intelligent parking system is meant to operate in an effective manner as per many prime parameters. One of the prime characteristics is to identify the parking lots that are close to the user's location.

   Reachability by Walking

   Walking distance is the distance from the driver's initial location to his or her destination parking lot. It is a measurement that is of utmost concern when considering user satisfaction since individuals tend to like parking close to their destination location.

   Find the traffic load

   Reserving parking spaces in advance has been integrated in our system, with regard to numerous constraints including traffic congestion. Our proposed solution has been designed as efficient and up-to-date as possible, and performance results have been satisfactory. The proposed parking solution has been simulated, and relevant data is provided in the following section.

   Searching blindly

   Our suggested solution makes sure that reserved parking spaces are notified to the driver beforehand, avoiding them being kept waiting at the last minute. Drivers can advance-book available parking space in advance; otherwise, they might be forced to start looking again, with the added threat that spaces will be filled.

   Shared parking information (spi)

   Since it is usual for people to inform others of vacant parking, our solution capitalizes on this aspect by providing a real-time way of verifying vacant parking at any moment. This gives everyone an opportunity to conserve precious time when roads are congested.

   Buffered shared parking information

   Being as aware as we are, peak-hour parking need is much higher as shown in Figure 5.3. In addressing this problem—named the Smart Parking Infrastructure (SPI) problem—the solution at hand reserves some of the parking spots as a buffer during peak hours. While the system still returns "full," these buffer spots deal with congestion and lower overall traffic load efficiently.

   We made random observations of the traffic during peak hours and found traffic to be heavier overall during the morning period of 7:00 AM to 10:00 AM and the

Date	Road	6–10 AM	10–2 PM	2–5 PM	5–9 PM	9–6 AM
16-June-2023	X	Medium	High	Low	High	Low
17-June-2023	X	Low	Medium	High	Low	Medium
18-June-2023	X	High	Low	Medium	High	Low
19-June-2023	X	Medium	Medium	Low	Low	High
20-June-2023	Y	Low	Low	Medium	Low	Low
21-June-2023	Z	Medium	Low	Low	Low	Low
22-June-2023	Y	High	Medium	Low	Low	High
23-June-2023	Y	Low	High	Medium	High	Low
24-June-2023	Y	Medium	Low	Low	High	Medium
25-June-2023	X	Low	Medium	Medium	Low	High
26-June-2023	X	High	Low	Medium	Medium	Low
27-June-2023	X	Medium	Low	Low	High	Low
28-June-2023	X	Low	Medium	Low	Low	Medium
29-June-2023	Y	Medium	High	Low	High	Low
30-June-2023	Y	High	Medium	Low	Low	Medium

Table 5.5: Data of traffic on two diffrent roads for some day

evening period of 4:00 PM to 6:00 PM. Traffic may increase due to festivals or games also. The BSPI functions very well under heavy loads in such situations as shown in Table 5.5.

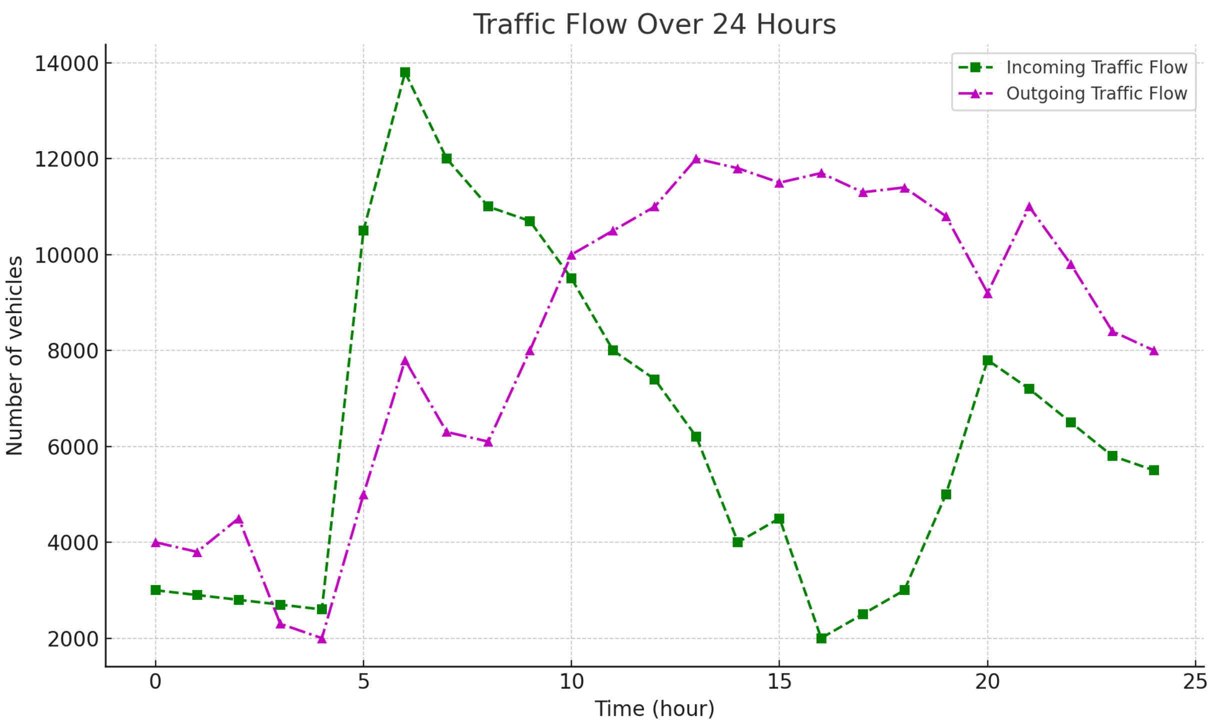


Figure 5.3: Traffic on date 16th June

2. **For indoor parking** We developed a simulation program in C++ where the detected License Number Plate (LNP) is used as input. Upon detection, the program checks the database to determine whether a parking slot (e.g., Slot X) has already

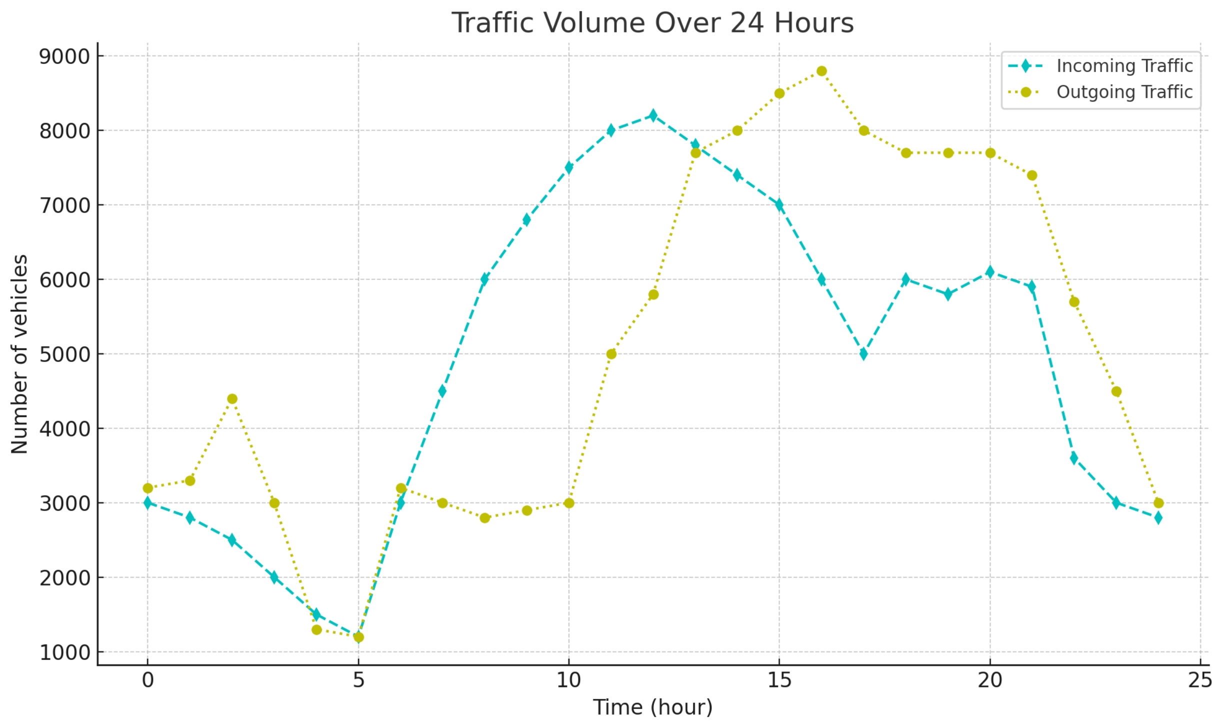


Figure 5.4: Traffic on date 30th June

been reserved. If so, it provides directions to reach that specific slot. In the case of a new vehicle entry, the system allocates a newly available slot (e.g., Slot Y) and provides the corresponding directions. The Figure 5.4 shows the traffic data for the month of June.

On the exit, the system drives the vehicle via the shortest route to the exit gate, rescans the License Number Plate (LNP) and produces a bill based on total parking time.

5.3 Result

5.3.1 Result for traffic management

A traffic intersection simulation was conducted and applied in the MATLAB environment, as depicted below.

From the graph Figure 5.6 and finding Figure 5.5, the system suggested was more efficient than the traditional fixed-time traffic system. It significantly assisted in cutting down the waiting time of cars, thus being an economy of time system compared to the standard fixed system.

5.3.2 Result for proposed parking system

1. **For parking in city** Traffic Parking looking for parking using proposed method:

   As I have shown in the Figure 5.7 it defines the comparision between diffrent searching technique.And this was created wit the help of Table 5.6

Figure 5.5: Simmulation of traffic system

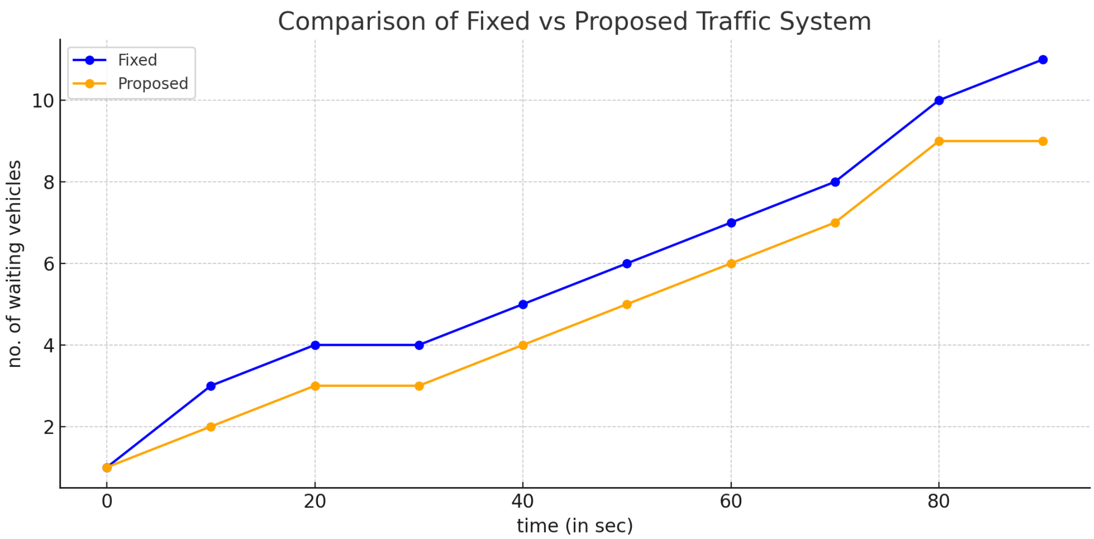


Figure 5.6: Compression between fixed and proposed system

Table 5.6: Comparison of Search Strategies Based on Travel Distance (in miles)

Method	6–10 AM	10 AM–5 PM	5–9 PM
Conventional Search	Least Efficient	Moderate	Least Efficient
PIS Approach	Least Efficient	Moderate	Moderate
BPIS Strategy	Moderate	Moderate	Moderate
Proposed Solution	Most Efficient	Most Efficient	Most Efficient

The findings showed that the system targeted worked best even at times of heavy demand. This method proved to be better than the PIS (Parking Information System) and BPIS (Backup Parking Information System).

2. **Parking to Exit** We used C++ for the simulation, and once the shortest path to the provided parking lot was defined, the following algorithm was used to find the

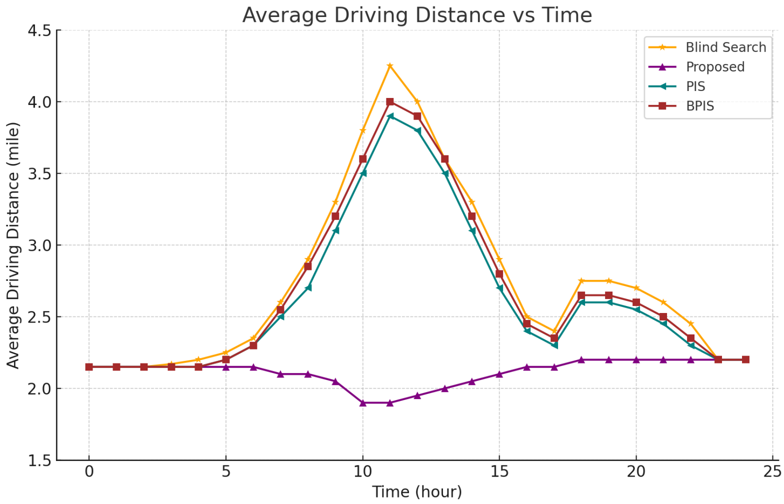


Figure 5.7: Compression between diffrent searching technique

shortest path to the exit.

Parking Lot Dimensions and Structure

The parking facility is organized in a grid format with a total of $K \times L = 10 \times 10$ units.

Legend for Area Representation

- **A** : Denotes a road segment used for vehicle movement.
- **B** : Represents a designated parking spot.
- **-1** : Indicates the vehicle entry point, located at coordinate (1, 1).
- **-2** : Marks the exit points, positioned at coordinates (10, 1) and (10, 10).

Layout Description

This grid-based configuration facilitates smooth entry, efficient parking, and streamlined exit operations. The entry point is strategically placed at the top-left corner of the grid, while the exit points are positioned at the bottom corners to minimize congestion and optimize traffic flow within the parking area.

Parking Grid Representation

A	B	B	B	B	B	B	B	B	A
A	A	A	A	A	A	A	A	A	A
A	B	B	B	B	B	B	B	B	A
A	B	B	B	B	B	B	B	B	A
A	A	A	A	A	A	A	A	A	A
A	B	B	B	B	B	B	B	B	A
A	A	A	A	A	A	A	A	A	A
A	B	B	B	B	B	B	B	B	A

Trial Records

Total Number of Trials: **5**

Trial Number	**LNP Number**
1	45878
1	23564
1	4781
2	23564
1	7894

Output

As a result of simulating the parking system, the navigation routes of individual vehicles within the designated parking grid were effectively identified. Furthermore, the duration between vehicle entry and exit was computed in seconds. This time interval served as the basis for calculating the respective parking charges.

Operation	Way Length	License Plate	Duration (m)
PARK	5	4781	29
EXIT	9	23564	52
PARK	4	23564	21
PARK	4	7894	35

Table 5.7: Summary of Vehicle Parking Operations Including Entry/Exit Type, License Plate Numbers, and Time Duration Spent in the Parking System

Pseudocode: Smart Parking Navigation System

Initialize Parking Grid

- Define grid of size $N \times M$
- Mark:

- ''A'' for Road
- ''B'' for Parking Slot
- Entry at $(1, 1)$
- Exits at $(N, 1)$ and (N, M)

Vehicle Parking Operation

- Receive `PARK` request with license number
- Search for first available vacant slot
- Use Dijkstra's Algorithm to find shortest path from Entry to Slot
- Record:
  - Time taken to reach
  - Path traversed
- Mark slot as `occupied`

Vehicle Exit Operation

- Receive `EXIT` request with license number
- Retrieve current parked slot
- Use Dijkstra's Algorithm to find shortest path to nearest exit
- Record:
  - Time taken to exit
  - Path traversed
- Calculate duration and generate parking bill
- Mark slot as `vacant`

Dijkstra's Algorithm for Pathfinding

- Start from source node (entry or parked slot)
- Use priority queue to explore minimum cost path
- Track parents to reconstruct the route
- Stop once exit or destination is reached

System Output

- Display parking map with highlighted path
- Summary:
  - Time in/out
  - Slot coordinates
  - Distance traveled
  - Parking fee

Chapter 6

Conclusion and Future Work

In today's era of fast-paced lives, the term "smart city" has become very prominent and is of utmost relevance. With the demand for smarter and more effective systems, research on a large scale is being conducted to make day-to-day systems smarter through the use of automation and connectivity. Technologies such as Artificial Intelligence (AI), Cloud Computing, and the Internet of Things (IoT) are revolutionizing traditional infrastructure into smart systems with improved performance, increased convenience, and better resource usage.

n order to make our cities completely smart destinations, it is crucial that important issues such as traffic congestion and parking inefficiency are addressed. This study primarily focuses on optimizing traffic and parking systems using the Internet of Things (IoT), Artificial Intelligence (AI), and Cloud Computing. The solution proposed here shows a novel and effective means, especially when it comes to parking, where the users can book vacant parking spaces remotely with the help of a web portal or a mobile phone application. The method not only makes one more time-effective but also saves fuel, thereby helping to make urban living greener. Additionally, the license plate recognition method used is very resilient and flexible, therefore making it perform just as well under changing environmental conditions.

In traffic management, the suggested method—integration of Artificial Intelligence using fuzzy logic with IoT sensor networks—is providing insight into more precise real-time traffic information. With this method, there is better decision-making at intersections, and waiting time greatly minimized, maximizing traffic flow efficiency overall.

Future Work

As all the data of customers are stored on the cloud in a secure manner, it has the potential to use this data in various connected parking systems. It also increases the comfort of users and system compatibility. The future development of the system can be like this:

Security: For enabling secure communication between the user and the service provider, symmetric encryption methods can be used for securing user data. In addition, more security algorithms can be incorporated to hinder further access by unauthorized users.

Privacy: For enabling protection of the user identity, a virtual system of identifier can be proposed. Through such an application, personal details are kept anonymous but still enable system operation and accountability.

Smart License Plate Integration: License plate numbers can be enriched with sensor-embedded plates that enable real-time user identification during emergency situations.

Sensing integration with IoT technologies can add system response and safety across environments.

Bibliography

[1] M. N. Peter and P. Rani, "Advanced traffic management system using a fuzzy logic controller to differentiate parking vehicle and moving vehicle," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 5, pp. 488–493, 2020, uploaded file: 104.pdf.

[2] M. N. Kamil and G. Kosala, "Zebra cross violation detection with yolov9: A novel approach for traffic regulation in indonesia," *Edumatic: Jurnal Pendidikan Informatika*, vol. 9, no. 1, pp. 41–50, 2025, uploaded file: 29244-183576-PB.pdf.

[3] L. Mainetti, L. Patrono, M. L. Stefanizzi, and R. Vergallo, "A smart parking system based on iot protocols and emerging enabling technologies," in *IEEE International Conference on Innovation and Technology*, 2015, uploaded file: mainetti2015.pdf.

[4] G. Abinaya, R. Banumathi, V. Seshasri, and A. N. Kumar, "Rectify the blurred license plate image from fast moving vehicles using morphological process," in *2017 International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, 2017, uploaded file: abinaya2017.pdf.

[5] M. A. Al-qanees, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, "An improved yolo-based road traffic monitoring system," *Computing*, vol. 103, no. 2, pp. 211–230, 2021.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[7] Chianyung, "Cross zebra violation detection system on motorcycle vehicles using the yolov4 algorithm," *E-Proceeding of Engineering*, vol. 10, no. 1, pp. 815–822, 2022.

[8] C. M. Mwangi, S. M. Kang'ethe, and G. N. Nyakoe, "Design and simulation of a fuzzy logic traffic signal controller for a signalized intersection," *International Journal of Engineering Research*, Year not specified, uploaded file: DESIGN AND SIMULATION OF A FUZZY LOGIC TRAFFIC SIGNAL CONTROLLER FOR A SIGNALIZED INTERSECTION.pdf.

[9] C. Dewi, R. C. Chen, Y. C. Zhuang, and H. J. Christanto, "Yolov5 series algorithm for road marking sign identification," *Big Data and Cognitive Computing*, vol. 6, no. 4, pp. 1–16, 2022.

[10] M. Glučina, N. Žigulić, D. Frank, I. Lorencin, and D. Matika, "Comparative analysis of yolov9 and yolov10 algorithms for urban safety improvement," in *IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2024, pp. 185–190.

[11] W. Y. Hsu and W. Y. Lin, "Adaptive fusion of multi-scale yolo for pedestrian detection," *IEEE Access*, vol. 9, pp. 110 063–110 073, 2021.

[12] L. Hu and Q. Ni, "Iot-driven automated object detection algorithm for urban surveillance systems in smart cities," *IEEE Internet of Things Journal*, 2017, uploaded file: hu2017.pdf.

[13] A. Imran, M. S. Hulikal, and H. A. Gardi, "Real time american sign language detection using yolo-v9," *arXiv preprint arXiv:2407.17950*, 2024.

[14] G. Lavanya and S. D. Pande, "Enhancing real-time object detection with yolo algorithm," *EAI Endorsed Transactions on Internet of Things*, vol. 10, pp. 1–9, 2024.

[15] J. Leng and Y. Liu, "An enhanced ssd with feature fusion and visual reasoning for object detection," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6549–6558, 2019.

[16] M. G. Naftali, J. S. Sulistyawan, and K. Julian, "Comparison of object detection algorithms for street-level objects," *arXiv preprint arXiv:2208.11315*, 2022.

[17] Y. Jia, S. Cui, X. Chen, and D. Qu, "Modeling of safe braking distance considering pedestrian psychology and vehicle characteristics and the design of an active safety warning system for pedestrian crossings," *Sensors*, vol. 25, no. 1100, 2025, uploaded file: sensors-25-01100.pdf.

[18] C. Ugwu and D. Bale, "An application of fuzzy logic model in solving road traffic congestion," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 2, pp. 2960–2969, 2014.

[19] D. Nkurunziza, R. Tafahomi, and I. A. Faraja, "Pedestrian safety: Drivers' stopping behavior at crosswalks," *Sustainability*, vol. 15, no. 16, pp. 1–17, 2023.

[20] S. Johnson, M. F. A, and H. K. S, "Iot based rear-end collision avoidance system in highways," *International Journal of Advanced Computer Research*, vol. 9, no. 45, 2019, uploaded file: out.pdf.

[21] P. Kanakaraja, L. S. P. S. Nadipalli, S. V. A. Kumar *et al.*, "An implementation of advanced iot in the car parking system," *Materials Today: Proceedings*, 2020, uploaded file: 10.1016@j.matpr.2020.09.042.pdf.

[22] D. Work, A. Bayen, and Q. Jacobson, "Automotive cyber physical systems in the context of human mobility," *National Workshop on High-Confidence Automotive Cyber-Physical Systems*, 2008, uploaded file: WorkAutoCPS2008.pdf.

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Smart Traffic Crossing and Advance Parking Management System using IoT

Total Pages ____________ Name of the Scholar UPendra Singh

Supervisor (s)

(1) Dr. Sanjay Patidar

(2) ____________

(3) ____________

Department Software engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin Similarity Index: 15, Total Word Count: 9571

Date: 30, 05, 2025

Upendra Singh

Candidate's Signature

Signature of Supervisor(s)

Smart_traffic_crossing_&_Parking_system.pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98171739

Submission Date

May 28, 2025, 4:47 PM GMT+5:30

Download Date

May 28, 2025, 4:49 PM GMT+5:30

File Name

Upendra_M.Tech_Thesis_23SWE11.pdf

File Size

2.6 MB

43 Pages

9,571 Words

50,294 Characters

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

90 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

14% Internet sources
3% Publications
4% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Replaced Characters
9 suspect characters on 4 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Smart_traffic_crossing_&_Parking_system.pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98171739

Submission Date

May 28, 2025, 4:47 PM GMT+5:30

Download Date

May 28, 2025, 4:49 PM GMT+5:30

File Name

Upendra_M.Tech_Thesis_23SWE11.pdf

File Size

2.6 MB

43 Pages

9,571 Words

50,294 Characters

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

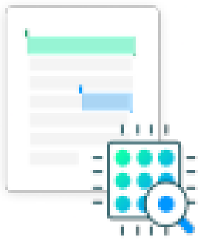
Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.