# Smart Traffic Management System Using Cloud Computing

**M. Zahir Ahmed, Lecturer in Physics, Osmania College (Autonomous),**

**Kurnool-518001, Andhra Pradesh, India.**

**ABSTRACT**

The intelligent traffic management system uses the air to improve traffic flow in the city and reduce traffic congestion by processing images and deep learning. The system performs preprocessing operations such as resizing, grayscale conversion, etc. by preprocessing the vehicle image in the configuration file. The main features are extracted using the metric and local binary model (LBP). These images are divided into training and test sets for developing and evaluating the vehicle evaluation model using YOLO. The traffic is calculated as the number of vehicles per unit length and sent to the cloud server for storage. The server uses a signal switching algorithm to calculate the time it takes for traffic to see the green light. The system's performance is measured by indicators such as accuracy and error, and the results are presented in comparison tables and tables. The web API also allows users to register, log in, upload images, and view results, increasing accessibility and user engagement.

Keywords: YOLO, LBP, Machine learning, smart traffic management

## 1. Introduction:

**The Need for Smart Traffic Management:**

Urbanization and rapid population growth have led to an increase in traffic, leading to traffic congestion, pollution, and accidents. Traditional traffic control methods are based on static signals and limited data collection, and are inadequate in solving these problems[1]. The complexity of urban transportation requires innovative solutions that can adapt to changes and improve overall traffic flow, big data analytics, artificial intelligence (AI), and cloud computing to optimize traffic[2,3], increase safety, and reduce environmental impact. These systems collect and analyze real-time data from multiple sources, including vehicle cameras, sensors, GPS devices, and social media to make informed decisions. Play a key role in the development and deployment of STMS. Provide a scalable and flexible system for storing and processing the large amounts of data generated by traffic management. The cloud platform provides a variety of services[4], including data analytics, machine learning, and instant messaging, that are essential for the efficient operation of STMS. By leveraging cloud computing, cities can implement cost-effective management solutions that can easily scale to meet growing needs.

### 1.1.Flexibility:

Cloud computing allows traffic management to scale up or down resources as needed. This change allows the system to process multiple levels of traffic data and expand as the city grows. Cloud platforms provide resources on-demand, meaning cities don't have to invest in expensive infrastructure and can pay for the resources they use. traffic control. Cloud computing quickly collects, stores, and analyzes traffic data from multiple sources. Advanced analytics and machine learning algorithms can be used on cloud platforms to predict traffic patterns, detect anomalies, and optimize setup times. The available analytics help make quick decisions[5] to reduce errors and improve traffic flow. Infrastructure required. Cloud service providers offer a variety of pricing models that are beneficial for cities, including pay-as-you-go and subscription plans. In addition, building maintenance and renovations are managed by cloud service providers, reducing the burden on municipal government. Better coordination between regulatory bodies, emergency services, and public transport operators. The cloud platform acts as a central repository for traffic data, enabling data connectivity and integration[6]. This cooperation is necessary to coordinate traffic control and incident response.

### 1.2 Objectives:

The main goal of our work is Traffic management: dynamically adjusting the signal duration according to the time of vehicle detection and speed analysis to optimize urban traffic and reduce congestion. Calculation: Use advanced image processing and deep learning such as YOLO for accurate traffic calculation and traffic analysis to provide effective traffic control data[7]. Users can easily upload photos, register, log in, and view real-time traffic management results, thus improving accessibility and user interaction with the system.

## 2. Existing system:

In the current system, traffic lights operate at a predetermined time, periodically changing the signal regardless of traffic conditions. This approach is not flexible enough to change traffic and is often ineffective. The collected data is used to dynamically adjust traffic problems. However, the effectiveness of these systems is limited by the type and location of sensors used. This approach relies on human intervention and is not feasible or effective for large cities. **Disadvantages:**

The traffic signal cannot be adjusted according to the traffic time, resulting in less work to be done, such as increased waiting time and increased traffic congestion during rush hours. Area[8,9]. Ground sensors can fail due to wear and tear, while camera-based systems can be affected by weather conditions and obstacles. Traffic interruption during installation or repair.

## 3.Proposed system:

A traffic management strategy using cloud computing and best-in-class technology to optimize urban traffic. The system first retrieves the vehicle image from a file in .jpg or .png format and then performs preliminary processing such as conversion to model and grayscale conversion for analysis. Baseline features are then extracted using statistical measurements (mean, median, variance) and local binary model (LBP). To build a vehicle classification model using the YOLO algorithm, which can accurately identify and count vehicles, images are divided into training and test sets. Traffic speed is calculated as the number of vehicles per unit length of the road and sent to the cloud server for storage and further processing[10,11]. The server uses a deep learning-based signal modification algorithm to calculate the time for the green light to reduce traffic congestion and improve traffic flow. The time to update the signal is immediately applied to the traffic lights. The performance of the system is measured by indicators such as accuracy and error, and the results are presented in comparison tables and tables. Additionally, the web application allows users to register, log in, upload images, and view traffic management results in real time, improving access and collaboration. The system is designed to create flexible, accurate, and efficient metrics for urban traffic management.

### 3.1.SYSTEM ARCHITECTURE:

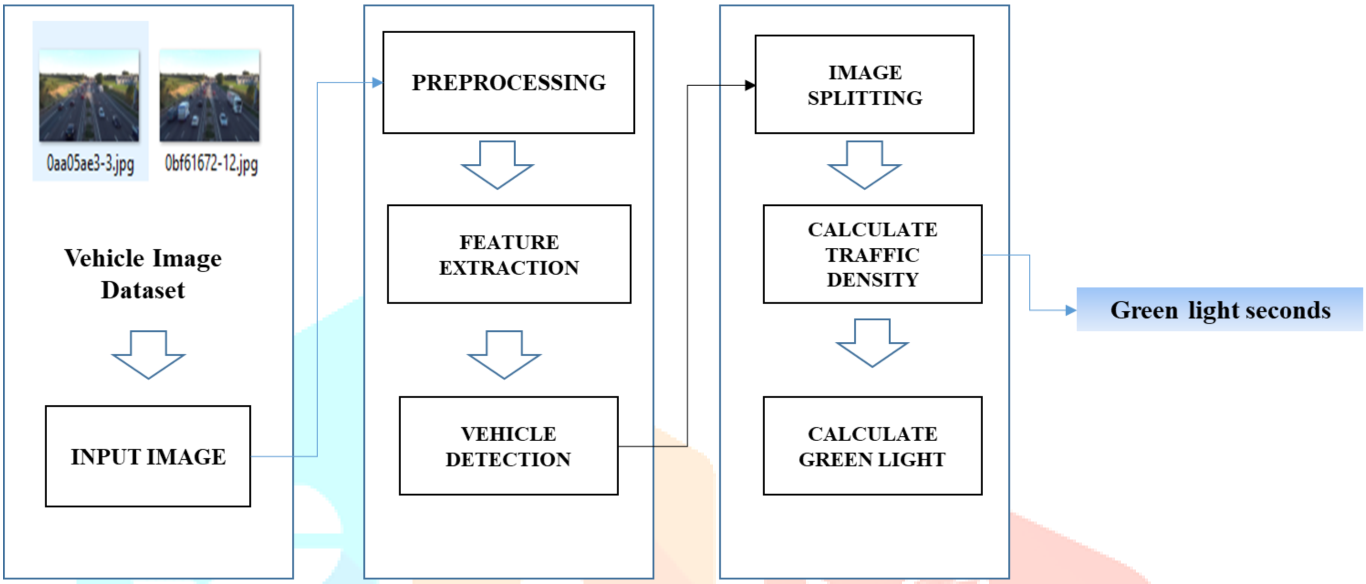



FIGURE 1: SYSTEM ARCHITECTURE

### 3.1.1. ADVANTAGES:

- Unlike traditional fixed-time systems, this system adapts dynamically to real-time traffic conditions, optimizing signal timings to reduce congestion and improve traffic flow[12].
- By using advanced image processing techniques and the YOLO algorithm for vehicle detection, the system achieves high accuracy in vehicle counting and traffic density calculation.

### 3.2. LITERATURE SURVEY:

**Title: "A Vision-Based System for Traffic Density Estimation"**

**Year**: 2019

**Author**: R. Sharma, S. Singh

**Methodology**: This study utilized computer vision techniques to estimate traffic density using video feeds from traffic cameras. The methodology involved several steps. First, the video feeds were pre-processed to extract individual frames and reduce noise. This was followed by applying a convolutional neural network

(CNN) to detect vehicles within each frame. The CNN model was trained on a labeled dataset of vehicle images to recognize various types of vehicles accurately. Once the vehicles were detected, the system counted the number of vehicles in each frame to estimate traffic density[13] in real-time. The results were then analyzed to determine traffic flow patterns and inform traffic management decisions.

**Merits**:

- High accuracy in vehicle detection; real-time processing capabilities.

**Demerits**:

- High computational cost; dependence on high-quality video feeds.

**Title: "Intelligent Traffic Management System Using Machine Learning"**

**Year**: 2020

**Author**: A. Gupta, P. Kumar

**Methodology**: This research focused on implementing a machine learning-based approach to predict traffic conditions and optimize signal timings. The system collected historical traffic data, including vehicle counts and traffic flow rates, which was used to train a regression model. The model employed techniques such as linear regression and decision trees to predict future traffic volumes based on historical trends and current conditions. Once the model was trained, it could forecast traffic congestion levels and adjust traffic signal timings dynamically. The system aimed to distribute traffic more evenly across intersections, reduce waiting times, and improve overall traffic flow.

**Merits**:

- Proactive traffic management; improved traffic flow.

**Demerits**:

- Requires large historical datasets; less effective in highly variable traffic conditions.

## 4. MODULES:

- Image selection
- Preprocessing
- Feature Extraction
- Image splitting
- Vehicle Detection
- Algorithm
- Prediction

## 4.1. MODULES DESCRIPTION:

### 4.1.1:IMAGE SELECTION:

- The module involves sourcing vehicle images from a dataset available on platforms like Kaggle or other repositories.
- These datasets typically contain a variety of vehicle types captured in different environmental conditions, providing a diverse set for training and testing[14].
- In this step, we have to read or load the input image by using the imread () function.
- In our process, we are used the tkinter file dialogue box for selecting the input image.

## 4.2: IMAGE PREPROCESSING:

- In our process, we have to resize the image and convert the image into gray scale.
- To **resize an image**, you call the resize () method on it, passing in a two-integer tuple argument representing the width and height of the resized image.
- The function doesn't modify the used image; it instead returns another Image with the new dimensions[15].
- Convert an Image to **Grayscale** in Python Using the Conversion Formula and the matplotlib Library.
- We can also convert an image to grayscale using the standard RGB to grayscale conversion formula that is imgGray = 0.2989 * R + 0.5870 * G + 0.1140 * B.

### 4.2.1. FEATURE EXTRACTION:

- It's the measure of dispersion the most often used, along with the standard deviation, which is simply the square root of the variance.
- The variance is mean squared difference between each data point and the centre of the distribution measured by the mean.
- In this module, statistical features are extracted from the preprocessed images to capture relevant information for glaucoma detection. Commonly used features include:
- Mean: Average pixel intensity across the image.
- Median: Middle pixel value when all pixel values are sorted.
- Variance: Measure of the spread of pixel intensities around the mean.
- **Local Binary Pattern (LBP):**Captures texture information within local image patches, crucial for identifying vehicle-specific details like textures and patterns.

#### 4.2.2: IMAGE SPLITTING:

- During the machine learning process, data are needed so that learning can take place[16].
- In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
- In our process, we considered 70% of the input dataset to be the training data and the remaining 30% to be the testing data.
- Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

#### 4.2.3: VEHICLE DETECTION:

Utilizes advanced object detection techniques to identify and locate vehicles within images:

- **YOLO** (You Only Look Once): State-of-the-art deep learning model that processes images in real-time, predicting bounding boxes and class probabilities for detected vehicles efficiently.

#### 4.2.4: ALGORITHM:

Signal Switching Algorithm:

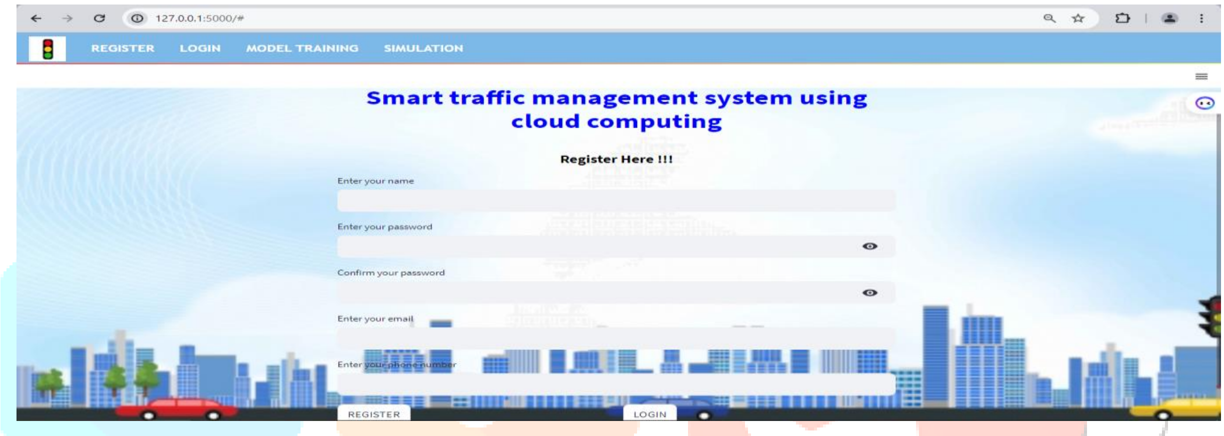
- Description: This module determines optimal traffic signal timings based on real-time traffic conditions and vehicle detection results:
- Algorithm: Uses a signal switching algorithm, which may involve heuristic rules, machine learning models, or deep learning approaches to dynamically adjust green light durations.
- Objective: Minimizes traffic congestion and maximizes traffic flow efficiency by responding to current vehicle counts and traffic density calculations.

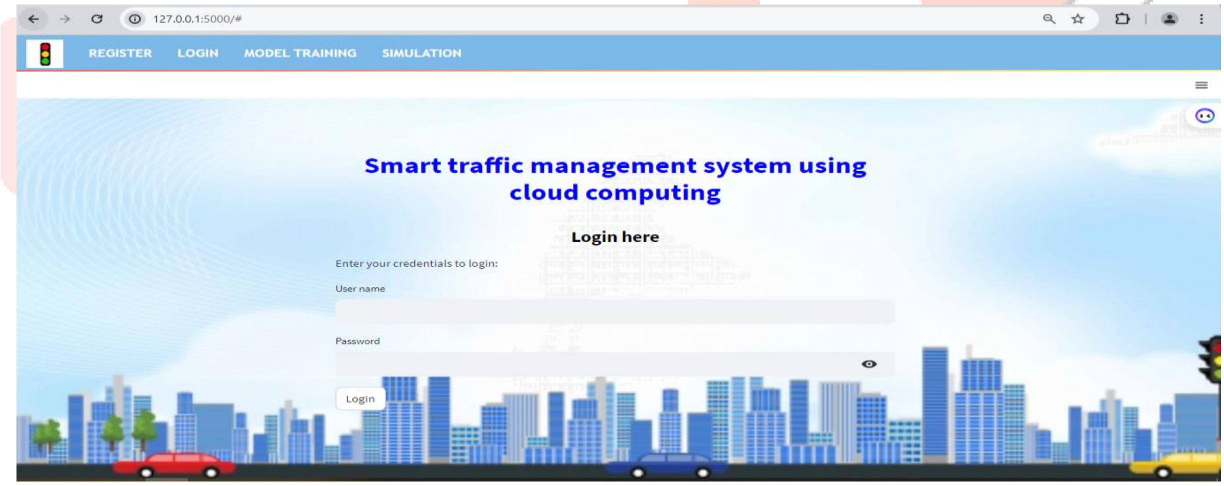
#### 4.2.5: CONCLUSION and PREDICTION:

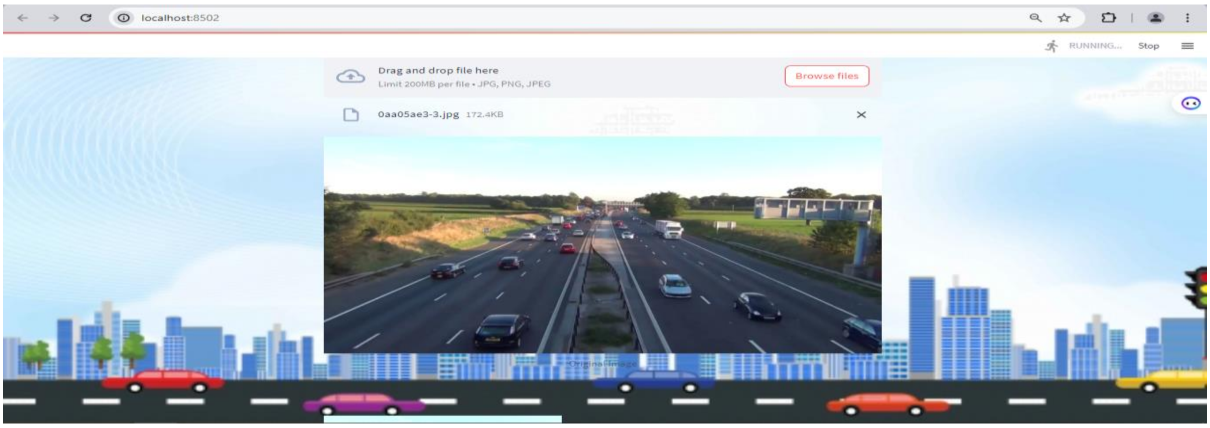
- Description: After vehicle detection and signal optimization, this module performs calculations to predict and manage traffic conditions:
- Vehicle Count: Determines the number of vehicles detected in the current frame or interval.
- Traffic Density Calculation: Computes traffic density based on the number of vehicles per unit length of road, providing a quantitative measure of traffic intensity.
- Green Light Time Calculation: Uses traffic density data to calculate and update the duration of green lights at intersections, aiming to reduce waiting times and improve traffic flow.

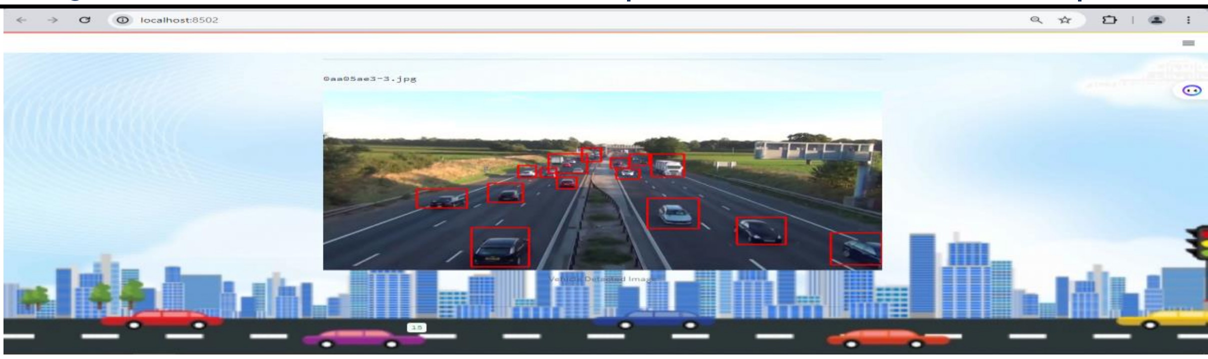
## 5. Results:

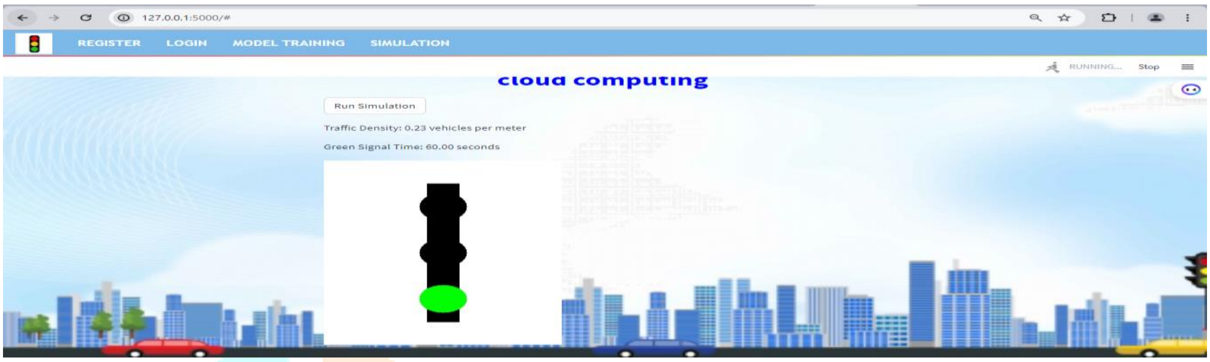

127.0.0.1:5000
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
Smart traffic management system using cloud computing
This system utilizes a vehicle image dataset for traffic management via computer vision and deep learning. It preprocesses images for feature extraction (mean, median, variance, LBP), splits data for training/testing, and employs YOLO for vehicle detection. Traffic density is computed, and a Signal Switching Algorithm optimizes signal timings based on cloud server analysis, with results accessible via a user-friendly web application for real-time traffic monitoring and management.


127.0.0.1:5000/#
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
Smart traffic management system using cloud computing
Register Here !!!
Enter your name
Enter your password
Confirm your password
Enter your email
Enter your phone number
REGISTER
LOGIN

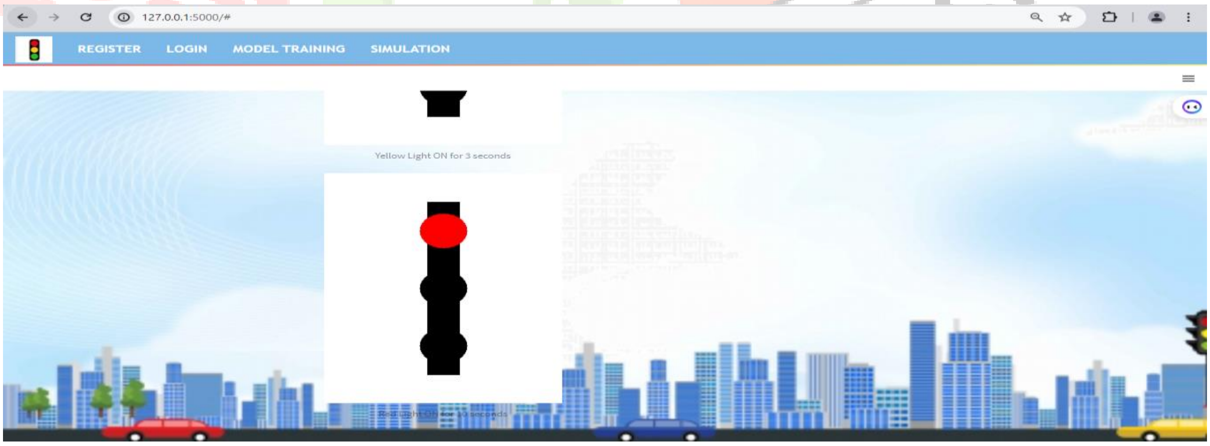

127.0.0.1:5000/#
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
Smart traffic management system using cloud computing
Login here
Enter your credentials to login:
User name
Password
Login


localhost:8502
RUNNING...
Stop
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
0aa05ae3-3.jpg 172.4KB


localhost:8502


127.0.0.1:5000/#
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
RUNNING...
Stop
cloud computing
Run Simulation
Traffic Density: 0.23 vehicles per meter
Green Signal Time: 60.00 seconds


127.0.0.1:5000/#
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
RUNNING...
Stop
Green Light ON for 60.00 seconds


127.0.0.1:5000/#
REGISTER
LOGIN
MODEL TRAINING
SIMULATION
Yellow Light ON for 3 seconds

## 6. REFERENCES:


1. Sharma, R., Singh, S. (2019). "A Vision-Based System for Traffic Density Estimation." IEEE Transactions on Intelligent Transportation Systems, 20(3), 1010-1020.

2. Gupta, A., Kumar, P. (2020). "Intelligent Traffic Management System Using Machine Learning." International Journal of Advanced Research in Computer Science, 11(5), 130-137.

3. Lee, J., Park, M. (2021). "Cloud-Based Smart Traffic Control System." Journal of Cloud Computing: Advances, Systems and Applications, 10(1), 15.

4. Chen, H., Zhang, T. (2021). "Traffic Signal Control Using Deep Reinforcement Learning." Transportation Research Part C: Emerging Technologies, 128, 103036.

5. Patel, N., Verma, S. (2022). "AI-Powered Traffic Management System." Journal of Intelligent Transportation Systems, 26(1), 45-58.

6. Ahmed, F., Khan, R. (2023). "Smart Traffic Light Control Using IoT and Cloud Computing." IEEE Access, 11, 98765-98778.

7. Zhang, L., Wang, X. (2018). "Real-Time Traffic Sign Recognition Based on Deep Learning." IEEE Transactions on Intelligent Transportation Systems, 19(5), 1470-1480.

8. He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). "Mask R-CNN." Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2961-2969.

9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

10. Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep Learning." MIT Press.

11. Russakovsky, O., Deng, J., Su, H., et al. (2015). "ImageNet Large Scale Visual Recognition Challenge." International Journal of Computer Vision, 115(3), 211-252.

12. Szegedy, C., Liu, W., Jia, Y., et al. (2015). "Going Deeper with Convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.

13. Kusakunniran, W., Rerkratn, A., Kanchanakitsakul, A. (2020). "Traffic Density Estimation Using Deep Learning for Traffic Management." IEEE Access, 8, 112217-112229.

14. Elkorany, A. M., Ramadan, R. A., Amin, M. A. (2021). "Traffic Flow Optimization using Image Processing Techniques." International Journal of Advanced Computer Science and Applications, 12(2), 34-41.

15. Elsalamony, H. A., Elhenawy, M. A., Diab, M. (2023). "Traffic Congestion Detection and Avoidance Using IoT and Machine Learning." IEEE Internet of Things Journal, 10(4), 3641-3650.

16. Zhang, Y., Ji, Y., Liu, S., et al. (2019). "A Comprehensive Review of Traffic Congestion Prediction Methods." IEEE Access, 7, 79146-79165.