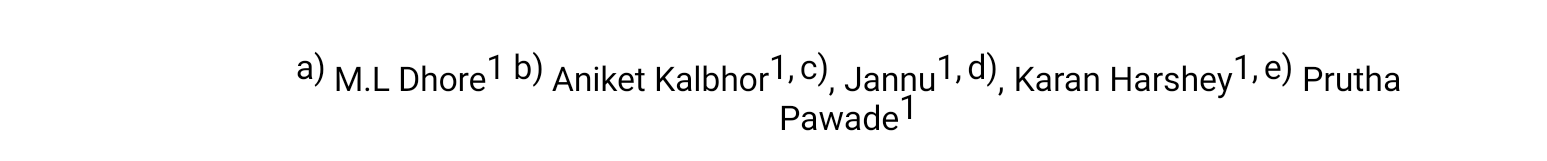
**Self-Adaptive Traffic Management System**



1*Department of Computer**Engineering, Vishwakarma Institute of Technology, Pune 411037*

*Author mails*

1. *manikaro.dhore@vit.edu*
2. *aniket.kalbhor22@vit.edu*
3. [*undefined.jannu22@vit.edu*](mailto:undefined.jannu22@vit.edu)
4. [*karan.harshey22@vit.edu*](mailto:karan.harshey22@vit.edu)
5. [*prutha.pawade23@vit.edu*](mailto:prutha.pawade23@vit.edu)

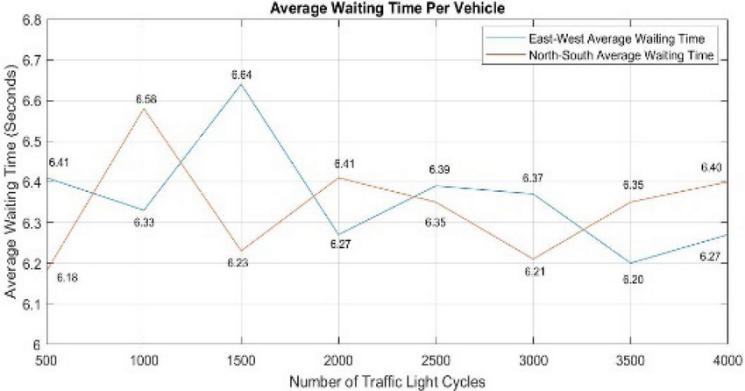
***Abstract:*** *Overpopulation has become a major global issue, contributing to an increase in the number of vehicles on the road, especially in case of high population density countries like India. Due to this, a significant share of time is wasted waiting on the roads. Adding to this, the current solutions of traffic management are highly inefficient as they’re not based on real time traffic conditions. Moreover, there’s no proper mechanism to handle the traffic in case an ambulance needs to pass quickly. This paper presents an adaptive traffic management solution utilizing the YOLO (You Only Look Once) algorithm. The system employs a Raspberry Pi 5 and a webcam to capture real-time traffic images. For this implementation, we focus on a four-lane crossing, where a camera mounted on a servo motor rotates to capture images from all four lanes. The system calculates the traffic density of all four lanes, and gives priority accordingly.* *If an ambulance is detected in a lane, that lane is given immediate priority to facilitate the ambulance's passage.*

***Keywords- YOLO, Embedded Image processing, Machine learning, Raspberry Pi 5, servo motor***

1. **Introduction**

Rapidly increasing urbanization has led to significant traffic congestion and safety concerns. Traditional traffic management systems are not very efficient, because of their lack of real time, adaptive operation. This is the exact problem that we aim to solve with our project. Our project consists of YOLO algorithm combined with a robust hardware setup. The microcontroller used are Raspberry Pi 5

and Arduino Uno. For image capturing purpose, we’ve used a high-quality web cam. The webcam is mounted on a servo motor to provide rotation so that it can capture image from all four lanes.



The camera captures images from all four lanes. These images are then run through a custom object detection and classification machine learning model trained by our group. The model then determines the number of vehicles in each lane, thereby assigning highest priority to the lane with least number of vehicles.

Once priorities are assigned, each lane is given a certain time for with it will get a green signal, depending on the traffic density.

In case the system detects an ambulance in any of the lanes, that lane will immediately get highest priority, thereby giving it green signal instantly so that the ambulance can pass through.  
  
This system thus provides a much more efficient solution to traffic management as opposed to traditional systems. Not only does it save time and ensure a higher degree of order at traffic signals, but it also has the potential to save lives because of its ability to instantly prioritize passing of an ambulance. Moreover, it can be easily integrated with the existing traffic management mechanisms, thereby making it fast and convenient to deploy.

1. **Literature Survey**

| Sr | Title | Methods Used |
| --- | --- | --- |
| 1 | Traffic Management Through Image Processing And Fuzzy Logic | Image processing, face recognition, MATLAB, fuzzy logic, morphological filtering, fuzzy interface system |
| 2 | Street Smart: An Intelligent Traffic Management System | Fuzzy logic, YOLO, trapezoidal membership function, background algorithm |
| 3 | Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information | Canny edge detection, Gaussian filter, image preprocessing, RGB to gray conversion, image matching, hardware implementation |
| 4 | The Traffic Congestion Investigating System by Image Processing from CCTV Camera | Contour-based method, median filtering, background subtraction, dilation and erosion operations |
| 5 | Smart Traffic Light Scheduling in Smart City Using Image and Video Processing | IoT, image processing, Raspberry Pi, OpenCV |
| 6 | A Vehicle Detection Technique for Traffic Management using Image Processing | RGB to HSV conversion, day and night image differentiation, foreground and background subtraction, image binarization, object counting methodology |
| 7 | Smart Traffic Light Using Raspberry Pi and Digital Image Processing | Prototyping, digital image processing, Raspberry Pi 4, OpenCV, TensorFlow |

*Table 1: Methods used in different papers*

The following paragraphs in this section address numerous concepts or studies that researchers have carried out and put into practice.

The literature survey outlines a series of research studies focused on the development and implementation of traffic management systems that leverage image processing, fuzzy logic, and IoT technologies. These systems are designed to optimize traffic flow and minimize delays by adapting dynamically to real-time traffic conditions. For instance, image processing combined with fuzzy logic has been employed to manage traffic signals more efficiently by adjusting signal timings based on real-time traffic demands. Additionally, future advancements in the integration of face recognition for criminal tracking purposes have been suggested [1].

Addressing the challenge of traffic congestion, particularly its impact on emergency vehicle response times, researchers have proposed systems that utilize fuzzy logic to manage lane density. These systems aim to enhance traffic control during uneven traffic distribution, with the YOLO algorithm providing real-time vehicle detection capabilities to improve overall system performance [2]. Similarly, another study proposes the use of Canny edge detection for measuring vehicle density and demonstrates that real-time data can significantly enhance traffic light management systems. Hardware implementation validates the effectiveness of this technique, with the novel approach of comparing detected edges further increasing accuracy [3].

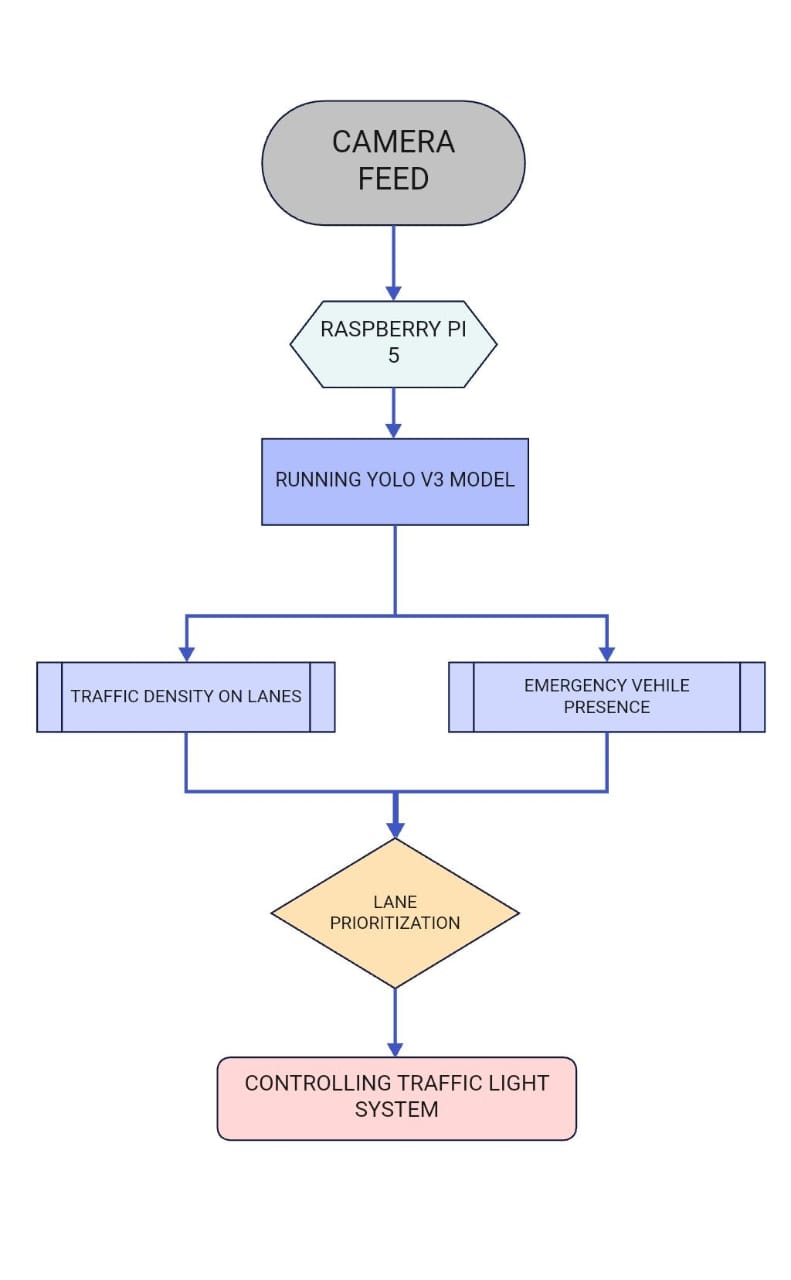
Further studies explore the use of image processing from CCTV camera feeds to classify traffic conditions into categories such as Flow, Heavy, or Jammed. These systems rely on background subtraction techniques and vehicle detection algorithms to compute traffic density, with the resulting data being fed into databases for transportation planning [4]. The integration of IoT and video processing techniques offers a novel framework for traffic light control, wherein a Raspberry Pi and OpenCV tools manage real-time traffic data. This system dynamically schedules traffic light timings based on vehicle density and quantity, ensuring efficient traffic flow [5][6].

Innovative methods for distinguishing day and night images by converting RGB to HSV and applying different object counting techniques have achieved high levels of accuracy, with studies reporting an average detection accuracy of 95% for both daytime and nighttime scenarios [8]. Another system uses the Raspberry Pi 4 as a controller, with TensorFlow and OpenCV handling real-time image processing. This system dynamically adjusts traffic light timings based on vehicle counts, demonstrating adaptability in real-time traffic conditions [9]. For emergency vehicle management, convolutional neural network (CNN)-based systems on Raspberry Pi platforms have been employed to rapidly assess whether to clear traffic for emergency vehicles [10].

Prototypes using MATLAB for image processing count vehicles captured by webcams to optimize traffic flow during peak hours [11]. Significant advancements in vehicle detection have been achieved through the application of deep learning models, particularly CNN architectures like YOLOv3, SSD, and Faster R-CNN. These models, when integrated with optical flow tracking and trajectory analysis, offer superior performance in terms of precision, recall, and F1 scores for traffic management systems [12]. Research also highlights the incorporation of the Kalman filter alongside YOLO for improved vehicle detection and motion analysis, leading to more effective traffic flow management [13].

Finally, the YOLOv7 model has shown promising results in enhancing object detection in traffic management applications. Experiments conducted using the bdd100k dataset have demonstrated notable improvements in both accuracy and efficiency. This model's adaptability to real-world conditions indicates significant potential for further optimization and refinement in future studies [14]. A dynamic traffic management system utilizing real-time image processing and the YOLO algorithm has been proposed to optimize signal timings, reduce vehicle waiting times, and improve overall traffic efficiency in urban areas [15].

1. **Methodology**



1. ***Image Capture:***

The process begins with capturing images across all four lanes of our crossing. The camera, which is mounted on a servo motor (controlled by Arduino Uno, to facilitate rotation) will take images from all four lanes.

****

*Figure 3 Servo Motor*

There is a set time interval at which the camera will rotate by 90 degrees to change lanes. This ensures that all lanes are successfully covered.



These captured images are then used for calculating traffic density across all four lanes. This allows us to determine the order of change of traffic signal, and the individual time slices for each lane.

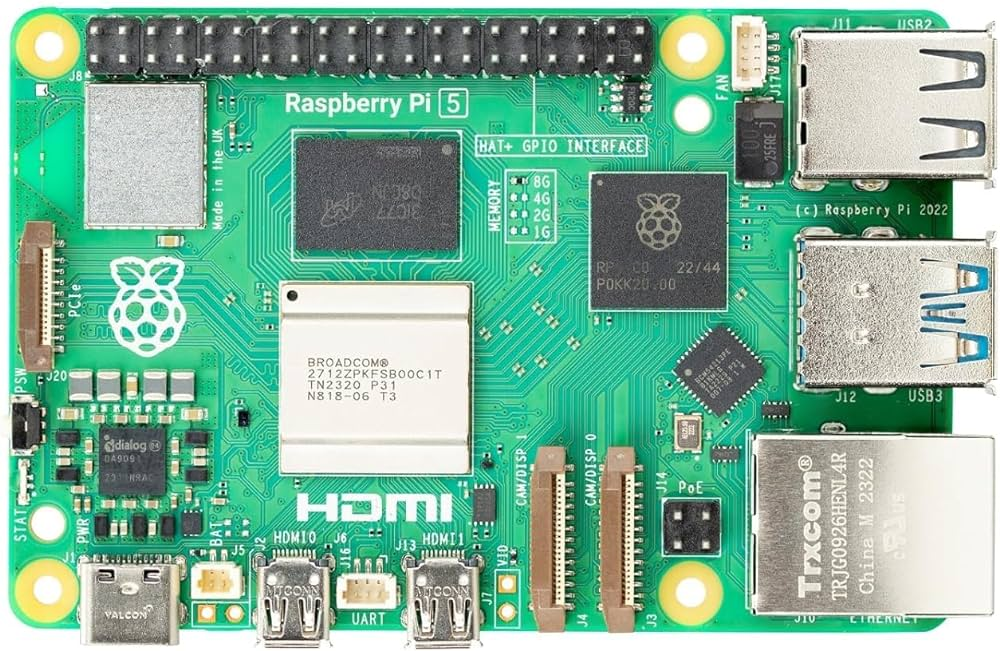
Now, the images are sent to Raspberry Pi 5, so that they can be further processed using YOLOv5 algorithm and the mechanism can move further.

1. ***Image Processing:***

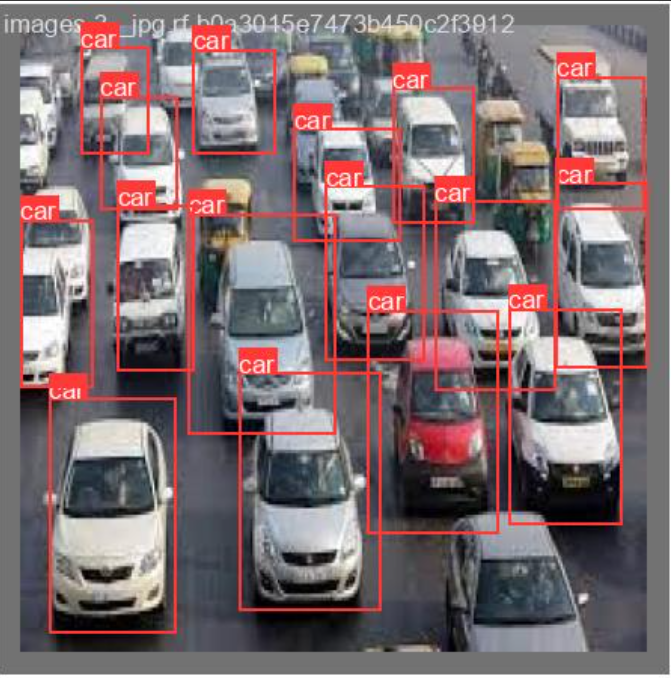
The next step in the methodology is image processing. Image processing refers to manipulation of images according to the way we want so that meaningful and necessary conclusions can be made from them.

In our case, the processing is done by YOLOv5 model. This model is responsible for detecting vehicles in the images, and then returning the count of vehicles. This is the count which is used for determining the traffic density, priority and time slice for each lane so that traffic can be managed efficiently. (Figure 5)

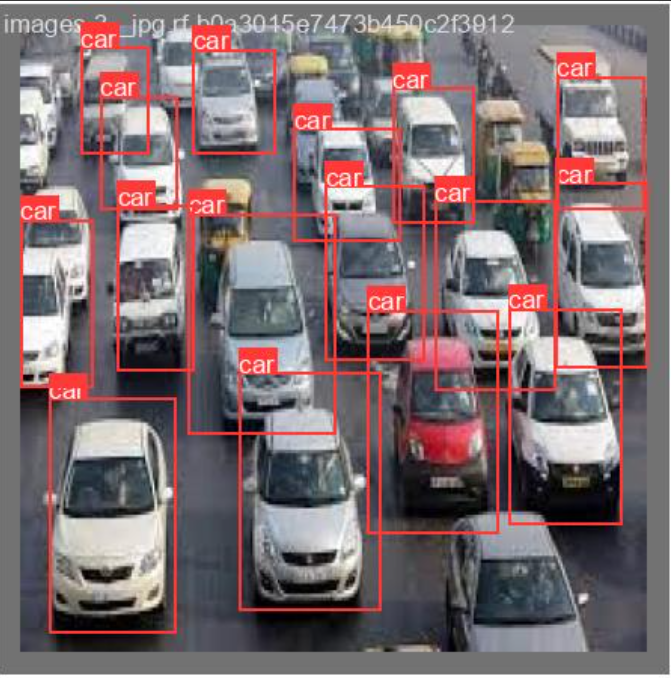
Thus, our count of vehicles is generated here, and this count is fed to an algorithm which further calculates the densities, priorities and time slices. (Figure 6)

In case of detection of an emergency vehicle (Figure 7) in a lane, that lane is immediately given the highest priority, and provided with a green signal so that the ambulance can pass through as quickly as possible. To do this, we had to train our model such as to make sure that it didn’t miss an ambulance in a whole slew of vehicles on the road.

This step is completed on the Raspberry Pi 5, which locally runs the model to detect vehicles, and then feed this count to further algorithm.



*Figure 5 Processed Image 1*



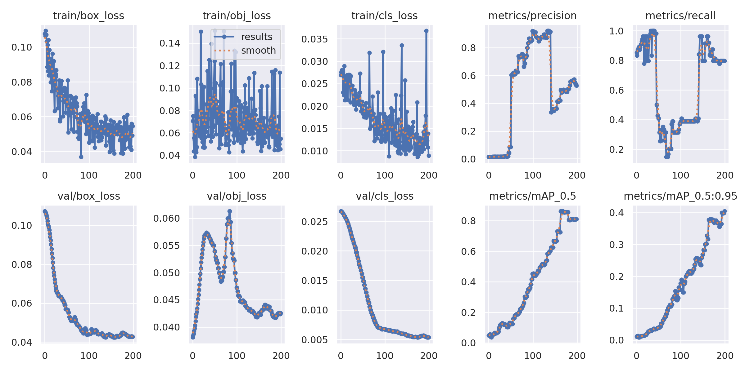
*Figure 6 Processed Image 2*

| **Metric** | **Initial Value** | **Final Value** | **Description** |
| --- | --- | --- | --- |
| **val/box\_loss** | **0.1** | **0.04** | **Consistent performance between training and validation sets** |
| **val/obj\_loss** | **0.055** | **0.045** | **Improved Object detection on the validation set** |
| **val/cls\_loss** | **0.025** | **<0.01** | |  | | --- |  | **Reflects improved classification accuracy on unseen data** | | --- | |



*Figure 7 Processed Image 3*

| **Metric** | **Initial Value** | **Final Value** | **Description** |
| --- | --- | --- | --- |
| **Metrics/precision** | **0** | **~0.9** | **High accuracy of positive predictions.** |
| **Metrics/recall** | **0** | **~0.9** | **High proportion of true positives correctly identified.** |
| **Metrics/mAP\_0.5** | **0** | **~0.9** | |  | | --- |  | **Significant overall performance improvement.** | | --- | |
| **Metrics/mAP\_0.5:0.95** | **0** | **~0.4** | **Robust performance across varying levels of precision and recall.** |



| **Metric** | **Initial Value** | **Final Value** | **Description** |
| --- | --- | --- | --- |
| **train/box\_loss** | **0.1** | **0.04** | **Indicates improvement in bounding box predictions.** |
| **train/obj\_loss** | **0.14** | **0.08** | **Reflects the model's ability to distinguish objects.** |
| **train/cls\_loss** | **0.035** | **<0.015** | |  | | --- |  | **Shows enhanced accuracy in class predictions.** | | --- | |

1. ***Controlling traffic lights:***

Now, after most of the processing part is done, it’s the actual output that comes. When the results of priorities arrive, the Raspberry Pi 5 instructs the Arduino Uno about the order in which to manipulate the traffic lights. The order if defined by using SJF (Shortest Job First) Non-Pre-emptive algorithm with time slicing.

The Arduino accepts this data from Raspberry Pi 5, and finishes manipulating the traffic lights in the instructed order.A diagram of a circuit board

Description automatically generated

Thus, this is where once cycle ends.

1. ***Continuing the loop:***

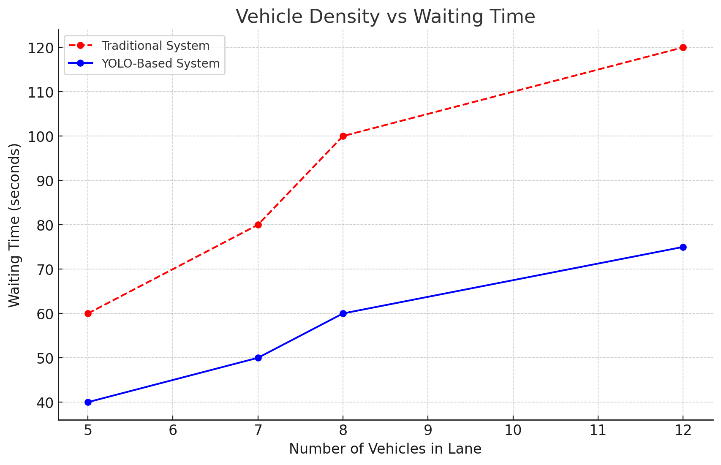
This will continue to run in an infinite loop without a stopping condition, since it has to resume the next round of image capture, processing and rotation as soon as it’s done with the first one.

1. **CONCLUSION**

We successfully implemented a system which is self-adaptive, and manages traffic in real time as opposed to traditional solutions which work on pre-set time slices.

After multiple rounds of testing, we got our model to a respectable level of accuracy, where we can confidently rely on it to produce positive results most of the times.

This project has massive applications in real life, due to the fact that it’s really easy to be fitted on to existing traffic management infrastructure. It’s portable, easy to program, energy efficient and saves significant amount of time for people on the road. Moreover, it’s a more thoughtful approach when it comes to emergency vehicles such as ambulances, police vehicles, firetrucks, etc. which need to reach their respective locations without any delays.



| **Lane** | **Traditional System (seconds)** | **YOLO-Based System (seconds)** | **Time Reduction (%)** |
| --- | --- | --- | --- |
| **1** | 120 | 80 | 33% |
| **2** | 150 | 90 | 40% |
| **3** | 130 | 85 | 35% |
| **4** | 160 | 100 | 37.5% |

*Table 2: Average Time Spent in Queue*

**ACKNOWLEDGEMENT:**

We are exceptionally grateful to our esteemed director, Professor (Dr.) R.M. Jalnekar, for giving us the opportunity to finish this project as a requirement for the EDI (Engineering Design and Innovation) Course. We are grateful to Prof. (Dr.) Sandip Shinde, Head of the Computer Engineering Department, for giving us the tools and assistance we needed. We also want to thank our project guide, Prof. M.L. Dhore, for all his help and support throughout the project, including her insightful advice, vision, and constant encouragement.

**REFERENCES:**

[1] Joel Joseph Joy, Manali Bhat, Namrata Verma, Milind Jani1, “Traffic Management Through Image Processing and Fuzzy Logic”.2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). 14-15 June 2018.

[2] Adwait Hegade, Ojas Patil, Soham Belurgikar,Anand, Anand, Anand Godbole,” An Intelligent Traffic Management System”. 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom). 15-17 March 2023.

[3] Taqi Tahmid, Ekal’s Hossain, “Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information”, 2017 3rd International Conference on Electrical Information and Communication Technology (EICT). 07-09 December 2017.

[4] Busarin Emanthankul, Mahasak Ketcham, Narumol Chumuang,” The Traffic Congestion Investigating System by Image Processing from CCTV Camera”. 2017 International Conference on Digital Arts, Media and Technology (ICDAMT). 01-04 March 2017.

[5] Meisam Razavi, Mehdi Hamidkhani, Rasool Sadeghi,” Smart Traffic Light Scheduling in Smart City Using Image and Video Processing”, 2019 3rd International Conference on Internet of Things and Applications (IoT), 17-18 April 2019.

[6] Basla Siripatan, Kittipong Nopchanasuphap, Somporn Chuai-Aree,” Intelligent Traffic Light System Using Image Processing”, 2021 2nd SEA-STEM International Conference (SEA-STEM), 24-25 November 2021.

[7] E. Paunova-Hubenova, E. Trichkova-Kashamova,” Algorithm for traffic management with priority for emergency vehicles”, 2022 XXXI International Scientific Conference Electronics (ET), 13-15 September 2022.

[8] Partha Narayan Chowdhury, Tonmoy Chandra Ray, Jia Uddin,”A Vehicle Detection Technique for Traffic Management using Image Processing”, 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), 08-09 February 2018.

[9] Helfy Susilawati, Shelmy Luthfi Pathur Rahman, Ade Rukmana, lik Muhamad Malik Matin, Sarbini, Nanang Ismail,” Smart Traffic Light Using Raspberry Pi and Digital Image Processing”, 2023 9th International Conference on Wireless and Telematics (ICWT)” 06-07 July 2023.

[10] S. Deepajothi, D. Palanival Rajan, P. Karthikeyan, S. Velliangiri,” Intelligent Traffic Management for Emergency Vehicles using Convolutional Neural Network”, 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 19-20 March 2021.

[11] Uthara E. Prakash, Athira Thankappan2, Vishnupriya K. T, Arun A. Balakrishnan,” Density Based Traffic Control System Using Image Processing”, International conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR2k18), November 2018.

[12] P. . Swathi, D. S. . Tejaswi, M. A. Khan, M. . Saishree, V. B. . Rachapudi, and D. K. . Anguraj, “Real-Time Vehicle Detection for Traffic Monitoring: A Deep Learning Approach”, Data and Metadata, vol. 3, p. 295, Jan. 2024

[13] C. Chen, "Intelligent Traffic Monitoring and Management Based on Computer Vision Technology," 2024 Second International Conference on Data Science and Information System (ICDSIS), Hassan, India, 2024, pp. 1-5

[14] Xuebin Hong, Jubin Huang, Weiwei Zhao, Huiwen Zou, Zhe Lin, Yuecheng Chen, "Object detection for traffic management based on YOLO," Proc. SPIE 13018, International Conference on Smart Transportation and City Engineering (STCE 2023)

[15] K. S. Rekha, S. S. M, T. M. D and S. R. L. B, "Dynamic Traffic Management System," *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Uttar Pradesh, India, 2023, pp. 1-5

| Sr | Title | Methods Used | Limitations |
| --- | --- | --- | --- |
| 1 | Traffic Management Through Image Processing And Fuzzy Logic | * Image processing for traffic management and surveillance. * Face recognition for tracking absconded criminals. * MATLAB for algorithm development and computation. * Fuzzy logic for traffic signal time management. * Morphological filtering for image processing. * Fuzzy interface system for traffic intensity analysis. | * Camera installation height affects traffic intensity calculations. * Viewing angle impacts effectiveness of traffic management. * Environmental conditions hinder camera visibility during fog. * Current traffic signal equipment is expensive and inaccurate. |
| 2 | Street Smart: An Intelligent Traffic Management System | * Fuzzy Logic based on lane density for traffic management * YOLO image processing for real-time object detection. * Trapezoidal membership function for fuzzification and defuzzification * Background algorithm for vehicle counting in each lane. | * Existing systems are inefficient and costly. * Emergency services are often overlooked in traffic management. * Current technology may fail in dimly lit conditions. * Traffic police struggle to manage congestion during rush hours. * Low accuracy in emergency vehicle detection due to dataset limitations. |
| 3 | Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information | * Canny edge detection for vehicle shape isolation. * Gaussian filter for image smoothing and noise reduction. * Image preprocessing including resizing and RGB to gray conversion, and Image matching by comparing detected edges for vehicular density. * Time allocation based on traffic density similarity. * Hardware implementation to verify results | * Current traffic control methods are outdated and ineffective. * Infra-red sensors and induction loops have significant demerits. * Vehicle counting may yield false results due to spacing issues. * Pixel counting can mistakenly include non-vehicle objects * Time allocation based solely on density may disadvantage low-traffic lanes. |
| 4 | The Traffic Congestion Investigating System by Image Processing from CCTV Camera | * Contour-based method for edge detection. * Median filtering to reduce noise * Background subtraction for object detection. * Dilation and erosion operations for image processing. | * The system cannot operate in rainy conditions. * Visibility issues affect traffic detection accuracy. |
| 5 | Smart Traffic Light Scheduling in Smart City Using Image and Video Processing | * The first model is based on vehicle density. * The second model is based on the number of vehicles * The system uses IoT and image processing techniques using Raspberry-Pi and OpenCV tools | * Previous systems require costly maintenance and constant analysis * Vulnerable to hard external conditions. * Limited by VANET hardware installation in vehicles. |
| 6 | A Vehicle Detection Technique for Traffic Management using Image Processing | * RGB images are converted to HSV images. * Day and night images are differentiated. * Foreground and background image subtraction is applied. * Image binarization is performed to filter pixels. * Object counting methodology is used for vehicle detection | * Detection accuracy decreases for vehicles at far distances. * Reflections of headlights can miscount vehicle presence. * Limitations in detecting vehicles during congested conditions. |
| 7 | Smart Traffic Light Using Raspberry Pi and Digital Image Processing | * Prototyping method for system development. * Digital image processing for object classification. * Raspberry Pi 4 microcontroller. * OpenCV for real-time computer vision applications. * TensorFlow for object detection model training. | * The detection process has a significant delay. * The calculation of vehicle numbers is often inaccurate. * The system's output may not align with website data. * Image processing methods are less effective for moving cars. |