IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134209

Smart Traffic Control System

Dr. Vasanthamma.H[1] , Shreya Navali[2] , Shreya SS[3] , Kritika[4] , Sharon Lilly[5]

Professor, Department of Computer Science and Engineering, Proudhadevaraya Institute of Technology, Hospet , Karnataka, India[1]

Students, Department of Computer Science and Engineering, Proudhadevaraya Institute of Technology, Hospet, Karnataka, India.[2-5]

Abstract: Our system uses camera feedback at the intersection using image processing and artificial intelligence to calculate traffic speeds. By optimizing the control of traffic lights according to the speed of the vehicle, we aim to reduce congestion, improve traffic flow and reduce pollution. In addition, our project eliminates ambulance delays by using an ambulance tracking system at intersections. The system detects the arrival of ambulances and adjusts traffic lights to speed up their passage, shortening response time. Geocoding facilitates operational efficiency by converting location data into location information for ambulance navigation.

I. INTRODUCTION

Urban areas like Mumbai and Bengaluru are facing serious traffic problems due to the increasing number of vehicles and irregular traffic controls. Traditional methods such as manual management, time management and electronic equipment have limitations. Manual control requires a lot of work and is therefore not suitable for general use. Fixed timer lighting does not change with traffic time, resulting in less work. Electronic equipment has faced the problem of accuracy and payment due to price limitation and limited use. The system uses CCTV cameras at intersections to capture real-time images to calculate traffic speeds. Using YOLO (You Only Look Alone), the system can detect and classify vehicles (cars, bikes, buses/cars, rickshaws) to accurately predict the time of the green light. Unlike static systems, the controller is proposed to dynamically adjust the signal timing according to traffic conditions. This reform improves green light timing, speeds up traffic and reduces delays, congestion and waiting times. The result is better traffic management that not only improves overall travel but also helps reduce fuel consumption and pollution. Mumbai and Bangalore solutions. Leveraging real-time data and advanced image processing technology, the system represents a significant improvement on traditional management methods and supports more stable and efficient transport travel across urban infrastructure. The vehicle search method should take less time than other in-depth methods. The device is designed to track vehicles through video surveillance. Unlike other neural network methods that detect specific objects, deep convolutional algorithms learn to increase the accuracy of detecting an ambulance. Therefore, the time taken is less than other methods and YOLOv3 is an exception to this time. It uses the latest technology to improve traffic flow, increase safety and reduce environmental impact. Intelligent vehicle management systems collect data from a variety of sources, including vehicle cameras, road-embedded sensors, vehicles' GPS signals, and mobile applications. This data is constantly analyzed to provide the latest traffic information. Advanced algorithms process incoming data to predict traffic patterns, identify hotspots and recommend routes for traffic. Machine learning algorithms can also adjust and evolve over time based on new lessons and information. This will help reduce delays, reduce downtime, and improve overall performance.

Intersections are the main source of traffic in the city. Intelligent traffic management uses technologies such as signal control, priority lanes for buses and emergency vehicles, and optimal intersection design to improve traffic and safety. Vehicles equipped with smart sensors and communication technology can interact with vehicle management to receive instant updates and recommendations. This enables features like cruise control, collision avoidance, and intersection control. attempt. A data-driven approach can lead to more efficient and effective transportation solutions. Smart traffic management systems help improve road safety and reduce carbon emissions by reducing congestion, reducing stop-and-go traffic and optimizing traffic. They also encourage the adoption of alternative modes of transportation such as public transport, cycling and walking. One of the most difficult problems: traffic.

II. LITERATURE SURVEY

Khushi, "Smart Control of Traffic Light System using Image Processing,"2019 [2] proposes a solution using video processing. The video from the live feed is processed before being sent to the servers where a C++ based algorithm is used to generate the results. Hard code and Dynamic coded methodologies are compared, in which the dynamic algorithm showed an improvement of 35% .

DOI: 10.17148/IJARCCE.2024.134209

In this method, VANETS are used to get information and location of every vehicle, which in turn is passed on to the nearest Intelligent Traffic light with the help of installed GPS Further, these ITLs will update the statistics and sent it to nearby vehicles. In case of accidents, the information would be sent to drivers to choose an alternate route to avoid congestion. However, this technique is not feasible as its deployment is quite expensive.

A. Vogel, I. Oremović, R. Šimić and E. Ivanjko, "Improving Traffic Light Control by Means of Fuzzy Logic,"2019 [3] proposes an Arduino-UNO based system that aims to reduce traffic congestion and waiting time. This system acquires images through the camera and then processes the image in MATLAB, where the image is converted to a threshold image by removing saturation and hues, and traffic density is calculated. Arduino and MATLAB are connected using USB and simulation packages, which are preinstalled. Depending on traffic count and traffic density, the Arduino sets the duration of green light for each lane. In this method, infrared sensor-based microcontrollers are used, which capture the unique ID of every car using transmitter and receiver. In case of an emergency situation, vehicle's radio frequency tags can be used to identify them and let other vehicles move. This method detects red light violations. However, this technique is not flexible due to the fact that infrared sensors need to be in sight.

A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2021 proposes a fuzzy logic-controlled traffic light that can be adapt to the current traffic situations. This system makes use of two fuzzy controllers with 3 inputs and one output for primary and secondary driveways. A simulation was done using VISSIM and MATLAB and for low traffic density, it improved traffic conditions. In this method, fuzzy logic technique is used in which two fuzzy logic controllers are used – one is to optimize the signal and the other controller is used to extend the green phase of a road in an intersection.

The sensors used to collect input data are video cameras that are placed at incoming and outgoing lines. The controller then utilizes the information collected through these sensors to make optimal decisions and minimize the goal function.
Renjith Soman "Traffic Light Control and Violation Detection Using Image Processing" 2020 [5] proposes a smart traffic light system using ANN and fuzzy controller. This system makes use of images captured from cameras installed at traffic site. The image is first converted to a grayscale image before further normalization. Then, segmentation is performed using sliding window technique to count the cars irrespective of size and ANN is run through the segmented image, the output of which is used in fuzzy controller to set timers for red and green light using crisp output. Results had an average error of 2% with execution time of 1.5 seconds. In this method, fuzzy logic is used, and the system takes in the number of vehicles and the average speed of traffic flow in each direction as the input parameters. The number of vehicles and the average speed of traffic flow can be determined using sensors placed on the road.

III. PRUPOSE OF THE PROJECT

A. Problem Statement

To reduce the traffic congestion based on density that poses a critical challenge in urban areas, causing delays, stress, and air pollution at particular junction and to implement the solution such that the ambulance doesn't get held in the traffic light giving it the maximum priority.

B. Existing System

1) In the current system, the use of RFID and Bluetooth has been used in vehicle detection for a long time.
Advantages: This app uses RFID and Arduino for police tracking; The advantage is that manual procedures are digitized using Android, so people making mistakes in crime control will be reduced, especially in Indonesia. : The number of tools required for vehicle inspection has increased. As there is more product, the price also increases. It takes time to establish the connection due to slow response time. The range is smaller, so the tip should be placed on each diameter of the device for greater accuracy.

2) Traffic control: As the name suggests, manpower is needed to control traffic. Traffic police were dispatched to the area to control traffic. Traffic police carry signs, lights and whistles to control traffic. It is believed that this strategy can be implemented as part of the management system. Since the traffic police are weak, it is not easy for us to control traffic all over the city.

3) Conventional lighting with static time: controlled by timer. Load a fixed value into the timer. The light will automatically turn red and green according to the timer value. They have a clear red, yellow and green signal pattern that helps control decisions and prevent accidents or unsafe conditions on the road. Disadvantages: Static traffic management uses a signal with a timer at each level. The signal is fixed and cannot be adjusted based on real-time conditions on the road.

DOI: 10.17148/IJARCCE.2024.134209

C. Proposed System

We propose a system designed to process images captured by CCTV cameras at intersections in real time. The main task is to modify the traffic calculation using the best performance tools and detection algorithms. Using input from these cameras, the system can detect traffic at busy intersections.

This approach allows us to better understand the current traffic situation, facilitate effective traffic management strategies and improve mobility in the city. The system also includes a place at the intersection of ambulances with the help of officers to control the ambulance from traffic, to be in front to give way to the ambulance, and to turn the red light to green until the ambulance passes the red light. light.

D. System Requirements

1. System Hardware: INTEL i3/i5:The Core i3, i5 Processor from Intel has a base clock speed of 3.7 GHz,3.60GHz respectively and comes with features such as Intel Hyper-Threading technology (Note: processor should be i3 or more than i3). 2. HARD DISK
2. Hard Disk requirement should be 500GB or more, Hard Disk helps in processing the project faster and hard disk is the primary computer storage device and is used to store files on the computer. 3. RAM
3. RAM (random access memory) temporarily stores the computer's operating system, application programs and current data so that the processor can reach them quickly. Ram is a faster memory and volatile in nature i.e., when the power is switched off, the data in this memory is lost. 4. CAMERA
4. Applicable to signal control system, traffic information service system, road traffic surveillance, etc. Integrated with millimeter wave radar and 4 MP low illumination camera. Multiple targets track detection and visualization in bi-directional 4 lanes with vertical coverage of 200 m. 1. OPERATING SYSTEM
5. An operating system is a piece of software that serves as a bridge between a computer's user and its hardware. It is a piece of software that handles computer hardware and lets the user to run programs quickly and easily. 2. SOFTWARE TOOL: OPEN CV PYTHON
6. The term "open-source computer vision" refers to software that is available for free. It was designed to provide an uniform architecture for computer vision operations and financial product system behaviour. Image processing, facial recognition, video recording, searching, and object disclosure are some of the topics it concentrates on. 3. CODING LANGUAGE: PYTHON
7. Summary of the report Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, in combination with dynamic typing and dynamic binding, make it ideal for Rapid Application Development and scripting.

E. Methodology

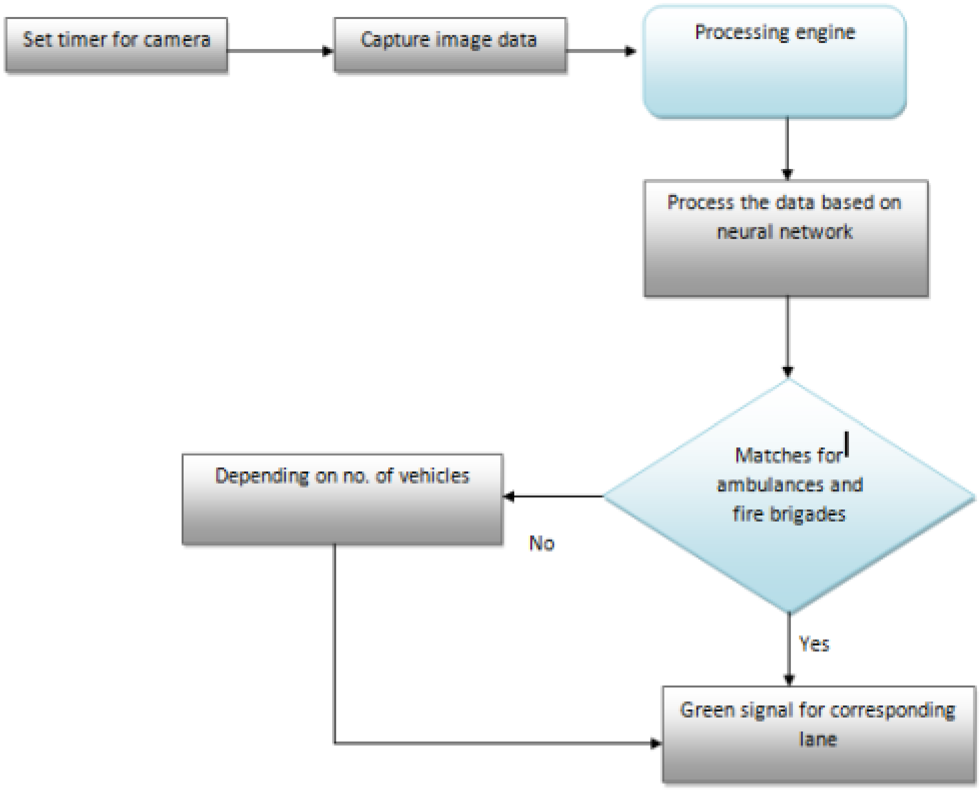
Our main goal when developing the "Smart Traffic Control System" is to produce original solutions that will reduce traffic congestion. We want the design to be around the main points and to ensure that all the pieces fit together well to create good results for the client.

This isn't about getting what you want; We want to go even further to be both creative and practical. The diagram below shows the system architecture - The architecture of our smart traffic management system and main ambulance system includes various methods to solve the urgent problem of ambulance and Ambulance is slow. First of all, the system integrates technology by using camera data at the intersection in real time to calculate the traffic speed with images and skills. Control Algorithm.

The design of our intelligent driving control system and the main ambulance in the ambulance contains many ways to solve the speed problem of truck patients and ambulances are slow. First of all, the system integrates technology by using camera data at the intersection in real time to calculate the traffic speed with images and skills.

The algorithm that controls the train set aims to reduce congestion and optimize travel time by changing dynamically according to the calculation of vehicle speed. The design also includes an ambulance tracking system that is activated in advance at intersections. The system uses YOLO technology for rapid ambulance detection, and when an ambulance is detected the system activates the green ambulance light during a critical 15-second window.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134209



In our system architecture, we will first set the camera's clock. The camera then captures everything along the line. After that it sends all the inputs to the machine and works as a neural network. It is one of the learning algorithms in machine learning and has three layers. The first is the input method that stores the data. The second is a hidden layer that divides the image into regions and predicts bounding boxes and probabilities that are weighted by the predicted probability for each region. The third is the output process, which contains all the data. If the captured data matches that of an ambulance or fire department, the traffic light will automatically display green for the corresponding line; otherwise it depends on the number.

F. Implementation

1. Video frame capture (capture())
The Capture() function plays an important role in extracting frames from recorded video files to facilitate the analysis and processing completed in the car searching and counting machine. This component is implemented in OpenCV's VideoCapture module to properly remove frames, enable navigation by time frame (frame_time_ms) or frame selection. Video data input

The Capture() function accepts input video data specified by the user or configured in the system. These video files often represent vehicle conditions or conditions that require vehicle inspection and calculation. Frame capture process OpenCV's VideoCapture module is used to open and read video files (cap = cv2.VideoCapture(video)). By setting the video capture function (cap.set (cv2.CAP_PROP_POS_MSEC, Frame_time_ms)), frames can be rotated according to the set time (frame_time_ms) to ensure the correct time during the standard capture process. Capture the time delay by making a good sampling of the frames for subsequent analysis. Time-based frame selection

The frame_time_ms parameter determines the time sampling strategy used during the frame. Randomly select frames or samples at a fixed time (frame_time_ms) to provide different processes and representatives for analysis. Frame Storage and Memory Management

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134209

Extracted frames are temporarily stored in memory or buffered for later processing. The Capture() function effectively manages memory resources to accommodate different video file sizes and durations, ensuring good performance during frame extraction. Integration with vehicle line detection

• Captured frames are used as input data for subsequent vehicle detection and system line calculations. The existence of multiple frameworks allows analysis and analysis of vehicle detection algorithms in different situations and situations.

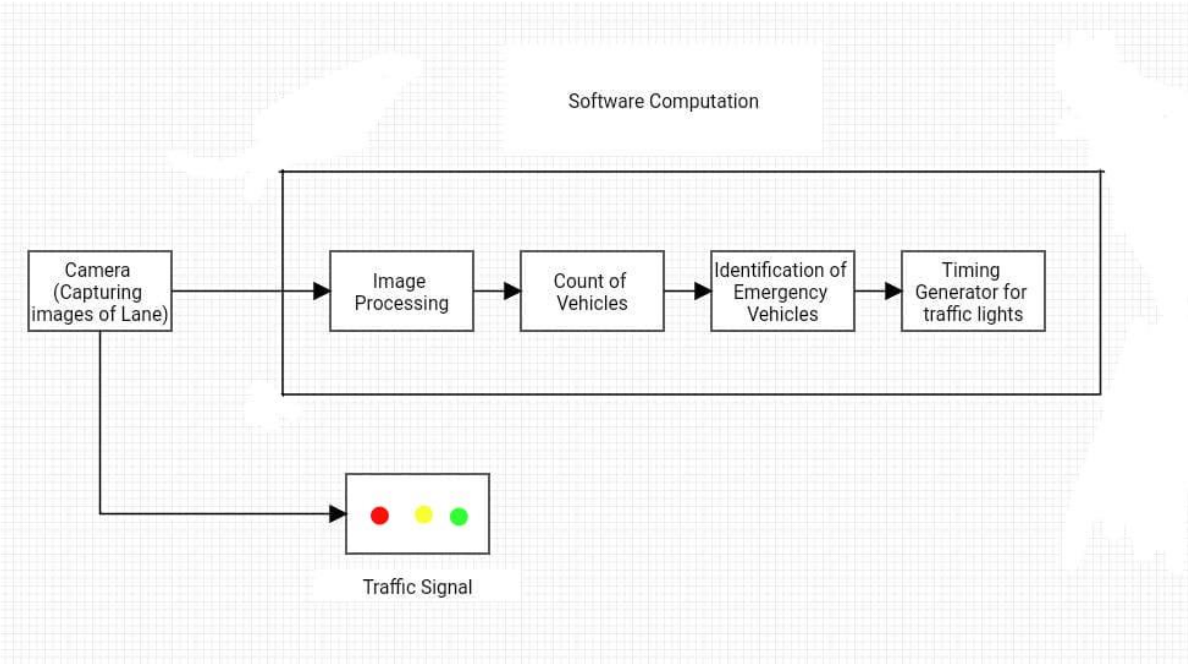


Fig 6.1: Video Frame Capture

2. Defining area (defining_area())
The definition_area() function plays an important role in applying user-defined region of interest (ROI) specifications to the video frame, which helps support the car. Correct in area of interest and count. This functionality is integrated into the graphical user interface (GUI) to provide users with an intuitive and interactive way to define the rectangular area within the video frame displayed in tkinter windows.

- Graphical User Interface (GUI) Integration:

This feature is seamlessly integrated into the tkinter-based GUI environment, allowing users to generate ROI using video frame interaction and Mouse click events. Define . Interactive ROI definition. The user defines the rectangle by clicking on the video screen and captures mouse click events to record the coordinates corresponding to each ROI's rectangle. Coordinate Registration and Persistence Mouse click events allow the capture and registration of coordinates that define the corner of a user-defined ROI. Save the recorded data by saving it in the text field ([] field) to ensure that the ROI is saved for later use during the vehicle search. Iterative area definition

- The definition_area() function supports ROI definition across multiple video frames or scenes, allowing users to define specific areas of interest based on custom situation or objective analysis. Users have the option to change or edit the ROI as needed to adjust the accuracy and coverage of the traffic detection system.

3. Vehicle Detection and Counting (count_vehicles()):
Vehicle identification and counting (count_vehicles()) .The count_vehicles() function integrates a pre-learned YOLO (look once) model to identify and count vehicles in the capture frame. The whole process is as follows

- Frame processing using the YOLO model :

Captured frames are resized into a model suitable for processing using the YOLO model (model.predict(image)). The YOLO model predicts checkboxes (values) for objects (including cars) in resized frames. Bounding Box Overlay .
The estimated bounding box (results) generated by the YOLO model is placed over the frames to display the image of the detected vehicles. Each bounding box contains information such as the detected object's coordinate, size, and class prediction.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134209

- Counting vehicles in ROI:

Counting vehicles detected in each ROI using predetermined regions (ROI) stored in Region[]. For each detected vehicle, its predicted class is retrieved from the classes.txt file to identify the vehicle type. The number of vehicles is calculated based on a predefined weight index (count_lst), which assigns different weights to different vehicle types.

- Weighted Vehicle Count Calculation:

Each vehicle type is assigned a weighted value (count_lst) that reflects its impact on traffic or operations. When detected, all vehicles are counted according to their weight values, allowing a comprehensive analysis of the vehicle mix.

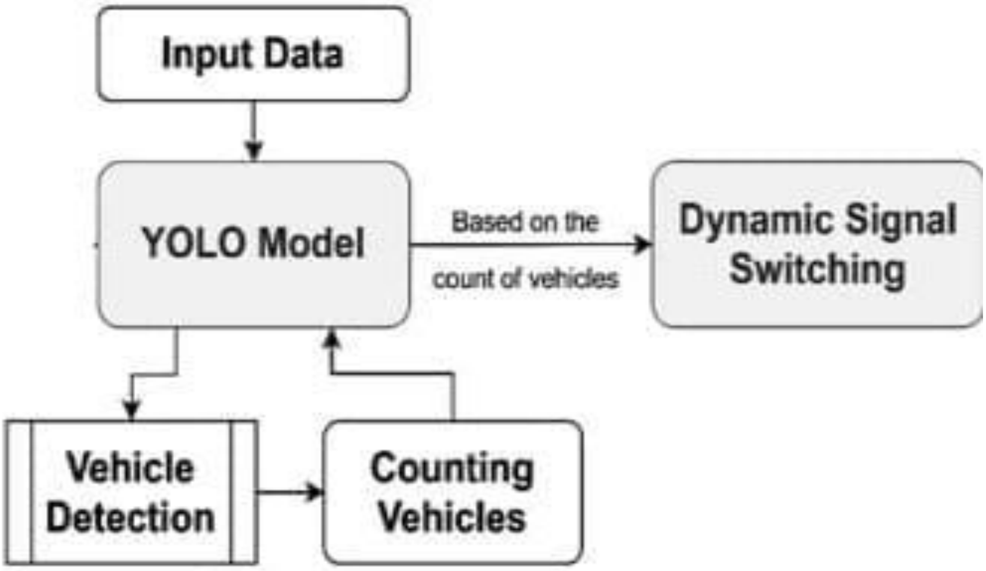


Fig 6.2: Vehicle Detection and Counting

4. User Interface (tkinter):

This project uses tkinter (CTk) to integrate a graphical user interface to enable interaction and control of the application process.

- Graphical Interface Components: After starting the application, a tkinter window (gui) will be created to hold the dialog and display the results. Provide the best user experience across different resolutions (screen width and screen height). Features and Controls™ Frame selection options: Users can choose to test on a new frame (new) or reuse a predefined region (ROI) in an old frame image (old). Display: A dedicated button (ambulance) allows users to simulate an emergency vehicle, show the vehicle count update, and complete the results. in the window Real-time feedback and results.

- ROI interpretation feedback: Capture mouse click events and record coordinates corresponding to the four corners of each ROI to facilitate real-time feedback real ROI. It provides users with instant feedback on traffic search results. Patterns & Layout.

- Style Configuration: Use custom configurations (corner_radius, fg_color, hover_color, border_color) to style user interface elements such as buttons (CTkButton), labels (CTkLabel), and typed text to enhance visual appeal.

5. Result Visualization (result_frame()):
The result_frame() function is responsible for visually presenting processed frames (final_frame) containing annotated vehicle counts and regions of interest (ROIs). This step enhances the interpretability and validation of vehicle detection outcomes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134209

Key aspects of this function include

- Frame Display and Annotation:

Image Rendering: The function creates a separate OpenCV window (Result) to display the processed frames (final_frame) with detected vehicles and annotated ROIs.

Visual Annotations: Detected vehicles within defined ROIs are highlighted using bounding boxes (cv2.rectangle()) and labeled with vehicle types (cv2.putText()), enhancing the visual representation of vehicle detection results.

- Interactive Inspection:

Real-time Inspection: Users can inspect the processed frames (final_frame) interactively within the Result window, enabling detailed examination of vehicle detection and annotation outcomes.

Frame Navigation: The Result window allows users to navigate through multiple frames, facilitating comprehensive validation and analysis of vehicle counts and detection accuracy.

- Result Verification:

Validation of ROI Definitions: The displayed frames assist in verifying the accuracy of defined ROIs by visually confirming the alignment of annotated vehicles with user-specified areas of interest.

Quality Assurance: Users can assess the quality and reliability of vehicle detection results through visual inspection, ensuring the robustness and effectiveness of the implemented vehicle counting algorithm.

- Enhanced User Experience:

Visual Feedback: The result_frame() function provides intuitive visual feedback on vehicle detection outcomes, aiding users in understanding and interpreting the processing results effectively.

User Interaction: The interactive nature of the Result window fosters user engagement and involvement in the validation and refinement of vehicle detection parameters and methodologies.

6. Ambulance simulation (oldone(ambulance_flag)):
The oldone(ambulance_flag) function provides the project with a simulation function that can simulate an emergency vehicle (ambulance_flag) in images. Increase the flexibility of vehicle search and operation based on the simulated emergency situation, improve the flexibility and responsiveness of vehicle calculation.
The important features of this function include

- Generating ambulance lane probability:

Random probability: This function generates a random probability (ambulance_lane_probability()) to simulate the special arrow of the ambulance in a special line or area Simulating emergency situations for vehicle detection. Considering the importance and function of the ambulance in vehicle counting, it will update the vehicle count and see the result. Product results are accurate and updated over time. User Interface Integration.

- Interactive feedback: Ambulance simulation functionality is seamlessly integrated into the graphical interface (tkinter) to provide users with interactive feedback on vehicle search and emergency response. : Users can promote greater collaboration and understand the role of the car's computing application by evaluating how the system affects the car's emergency detection. The simulation improves the ability of the vehicle counting system to react in different situations and verifies the robustness and adaptability of vehicle counting. Various scenarios, including emergency vehicle selection.

IV. RESULTS

Implementing a dynamic traffic control system featuring an ambulance clearance feature utilizing the YOLO (You Only Look Once) model promises transformative results in urban traffic management. The YOLO model's renowned efficiency in real-time object detection facilitates the system's ability to monitor traffic conditions instantaneously. By training

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ∺ Peer-reviewed & Refereed journal ∺ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134209

YOLO to recognize ambulance vehicles, the system can swiftly identify them as they approach intersections or encounter congested areas. Upon detection, the ambulance clearance feature seamlessly adjusts traffic signals, granting priority passage to the emergency vehicle. This agile response significantly reduces response times for emergencies, potentially saving lives in critical situations. Moreover, the system optimizes traffic flow by minimizing congestion and stoppage times, benefitting both emergency vehicles and regular commuters with smoother and more efficient travel. Safety is markedly enhanced, as the system's proactive management reduces the risk of accidents and fosters a safer road environment. With scalability and adaptability, the system can be tailored to diverse urban landscapes, providing tailored solutions to varying traffic dynamics. By integrating seamlessly with existing infrastructure and offering data-driven insights, this innovative approach promises not only immediate improvements but also a foundation for continued optimization and urban planning refinement.

V. CONCLUSION

In conclusion, the implementation of a smart traffic management system with ambulance clearance features presents a significant advancement in urban transportation and emergency response capabilities. This innovative system offers numerous benefits and transformative impacts. Firstly, it enhances emergency response by reducing ambulance response times through the prioritization of ambulance routes during emergencies, ultimately leading to improved patient outcomes and potentially saving lives. Secondly, the system improves overall traffic efficiency by dynamically managing traffic flow, minimizing congestion, and reducing travel times for all road users. Additionally, the provision of clear pathways for ambulances contributes to public safety and awareness, fostering a culture of responsiveness towards emergency vehicles among drivers. Furthermore, the system enables optimized resource allocation through real-time data analytics and predictive modeling, allowing authorities to allocate emergency services more effectively based on traffic conditions and demand. Lastly, by reducing congestion and idling times, the system also delivers environmental benefits, contributing to lower emissions and a greener urban environment. Overall, the adoption of a smart traffic management system with ambulance clearance features represents a crucial step towards building safer, more efficient, and sustainable cities.

REFERENCES

[1]. TomTom.com, 'Tom Tom global site visitors Index', 2019. [Online] available: https://www.tomtom.com/en_gb/traffic-index/rating/

[2]. Khushi, "smart manipulate of site visitor's mild gadget the usage of photograph Processing," 2019 international convention on present day developments in laptop, electric, Electronics and conversation (CTCEEC), Mysore, 2017, pp. ninetynine-103, doi: 10.1109/CTCEEC.2017.8454966.

[3]. Open information technology, 'assessment of the YOLO object Detection algorithm', 2019. [Online]. to be had: https://medium.com/@ODSC/ overview-of-the-yolo-object-detection-algorithm7b52a745d3e0. "wrong way DRIVERS DETECTION based totally ON OPTICAL waft" Gonc Institute for machine and Robotics Dep. of electrical Engineering and computer systems university of Coimbra - Portugal," no. 1, pp. 141–144, 2021

[4]. k. He, G. Gkioxari, P. greenback, and R. Girshick, "mask r-CNN," in 2017 IEEE international conference on laptop imaginative and prescient (ICCV), 2020, pp. 2980–2988.

[5]. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767,2020.

[6]. Renjith Soman "site visitors light manage and Violation Detection the use of image Processing"." IOSR magazine of Engineering (IOSRJEN), vol. 08, no. four, 2019, pp. 23-27

[7]. Tzutalin, 'LabelImg Annotation device', 2020. [Online]. available:https://github.com/tzutalin/labelImg [9] Konstantine Buhler, John Lambert, Matthew Vilim "YoloFlow real-time object tracking in Video" CS 229 direction venture.

[8]. Jing Tao, Hongbo Wang, Xinyu Zhang, Xiaoyu Li, Huawei Yang. "An item Detection machine primarily based on YOLO in visitors Scene"

[9]. D. G. Feitelson, D. Tsafrir and D. Krakov, "Experience with using the parallel workloads archive", *J. Parallel Distrib. Comput.*, vol. 74, no.3, pp. 2967-2982, 2014.