



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

Traffic Light System using Raspberry Pi and Open CV

Niranjan B S*, Sagar K R*, Vivek S*, Shashanka B* Dr. B.S.Nanda†,

Department of Electronics and Communication Engineering, PES College of Engineering, Mandya, Karnataka, India*

Professor, Department of Electronics and Communication Engineering, PES College of Engineering, Mandya,

Karnataka, India†

ABSTRACT: In metropolitan cities, traffic congestion has become a significant issue, leading to delays, fuel wastage, and environmental pollution. This project presents a Flask-based application leveraging OpenCV and Python for density-based traffic management. The system uses a laptop camera or pre-stored images to detect vehicle density across four lanes of a junction. YOLOv5, a state-of-the-art object detection algorithm, is employed to identify and count vehicles. Traffic signal timing is dynamically adjusted based on lane density to optimize traffic flow. Furthermore, the system prioritizes ambulances by instantly switching signals to the ambulance lane, ensuring swift passage. This integrated approach enhances traffic management efficiency and reduces congestion while providing a life-saving solution for emergency vehicles.

KEYWORDS: Raspberry Pi, OpenCV, Traffic Light Detection, Real-time Image Processing, Computer Vision, Python Programming, Colour Detection, Edge Computing, Automation, Smart Traffic Control, Surveillance Camera, Signal Recognition.

I. INTRODUCTION

The exponential rise in urban traffic has made traditional fixed-timer traffic signals inefficient. Managing traffic density dynamically using real-time detection offers a more effective solution. By integrating machine learning techniques such as YOLOv5 for vehicle and ambulance detection with OpenCV for image processing, this project aims to create a robust, scalable traffic management system. The Flask framework enables real-time updates and user-friendly interfaces, making it suitable for smart city applications. A Traffic Light System is an essential part of modern urban infrastructure to manage the flow of vehicles and pedestrians. By using the Raspberry Pi in combination with OpenCV, we can simulate and create a real-world traffic light system that is automated, responsive, and can be extended to include image processing capabilities (such as vehicle detection). Control a traffic light system using the Raspberry Pi (RPI). Implement traffic light signals (Red, Yellow, Green) using simple GPIO (General Purpose Input/Output) pins. Use OpenCV for basic image processing tasks, which can potentially enhance the system by detecting traffic or cars.

II. RELATED WORK

The development of intelligent traffic light systems has been a significant area of research aimed at mitigating urban traffic congestion and improving road safety. Traditional traffic light systems often rely on fixed-time intervals, leading to inefficiencies and unnecessary delays during varying traffic conditions. To address these limitations, researchers have explored adaptive approaches, with a growing emphasis on leveraging computer vision, embedded systems, and the Internet of Things (IoT).

[1]Salama (2009) proposed an intelligent cross-road traffic management system employing distributed photoelectric sensors to monitor vehicle movement and dynamically adjust traffic lights based on a calculated "relative weight" for each road. [2] Rani and Kumar (undated/2018) utilized Infrared (IR) sensors in conjunction with IoT to enable dynamic traffic light control based on real-time vehicle density, aiming for a cost-effective and automated solution. While sensor-based systems offer improved adaptability over fixed timers, their performance can be affected by environmental factors and limitations in obtaining comprehensive traffic information.

[3] Salvi (2006) pioneered an automated vehicle counting system based on blob analysis (background subtraction, blob detection, and tracking) for robust traffic surveillance. This approach allowed for real-time vehicle flow detection, crucial for effective traffic management. [4] Kalpana (2002) introduced an intelligent traffic signaling system that acquired traffic data through image processing techniques, focusing on detecting vehicle edges to count traffic. These early vision-based systems laid the groundwork for more sophisticated analysis.

Beyond basic density-based control, recent research has explored more advanced functionalities. [5] Heung (2007) investigated coordinated road junction traffic control using dynamic programming and local fuzzy-logic controllers to derive optimal green times for interconnected junctions, signifying a move towards network-level optimization. [6] The concept of emergency vehicle prioritization has also been a key focus, with some systems incorporating sound sensors to detect sirens, as demonstrated by Singh et al. (2021), or exploring RFID/wireless communication for priority clearance.

III. SYSTEM ARCHITECTURE

The architecture includes two primary modules:

Laptop Application: This is the central processing unit of the system, likely running on a laptop or a similar edge computing device deployed near a traffic intersection. **Camera Input:** This represents the video feed coming from a camera monitoring the traffic flow at the intersection. **Video Stream:** This is the continuous flow of visual data captured by the camera and fed into the application. **YOLOv5 Algorithm:** The output of the YOLOv5 algorithm includes a count of the total number of vehicles detected in the frame. This information is passed to the Density Calculator. **Ambulance Detection:** The YOLOv5 algorithm also specifically identifies ambulances within the traffic scene. This information is sent to the Ambulance Detector.

Traffic Signal Controller: This is the existing system that manages the traffic lights at the intersection. **Signal Control:** This represents the standard operation of the traffic signal controller, following its regular timing patterns. **Signal Adjustment:** The Traffic Signal Controller receives requests for signal adjustments from the Flask Server. **Signal Control (Output):** Based on the "Signal Adjustment" request, the Traffic Signal Controller modifies the traffic light timings (e.g., extending the green light for the lane with the approaching ambulance, shortening red lights on its path).

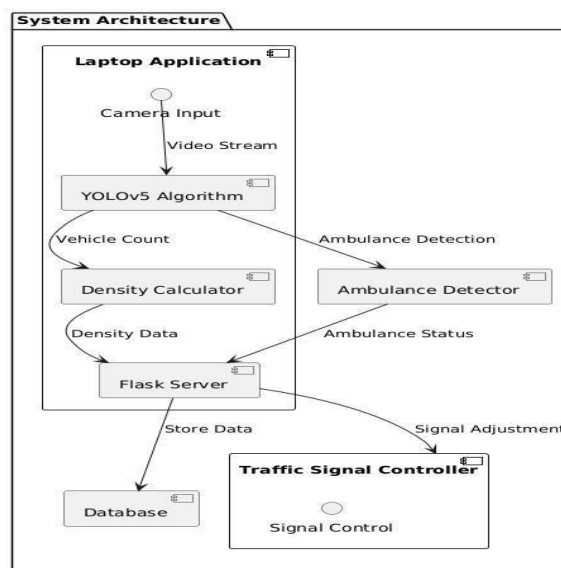


Fig.1 System Architecture

IV. IMPLEMENTATION

Laptop with camera: The laptop serves as the primary processing unit, running the Flask-based Python application. It is connected to a camera that streams live video footage or uses pre-recorded images as input for analysis.

YOLOv5 Algorithm: YOLOv5 (You Only Look Once, version 5) is a state-of-the-art object detection algorithm used to identify and count vehicles and ambulances. It processes video frames to extract critical information, including vehicle density and the presence of ambulances.

Density Calculation Module: This module calculates the number of vehicles in each lane based on YOLOv5's output. It assigns dynamic green light durations to lanes with higher vehicle density, ensuring efficient traffic management.

Ambulance Detection Module: This module identifies ambulances in the video feed. When detected, it overrides regular density-based traffic management, prioritizing the ambulance's lane by switching the signal to green.

Flask Web Server: Flask facilitates a user-friendly web-based interface, displaying real-time traffic status, lane density, and signal allocations. Users can monitor the system's operations seamlessly.

Traffic Signal Controller: Signals are controlled based on data from the density calculation and ambulance detection modules. Signals are dynamically adjusted to prioritize higher density lanes and ambulance paths.

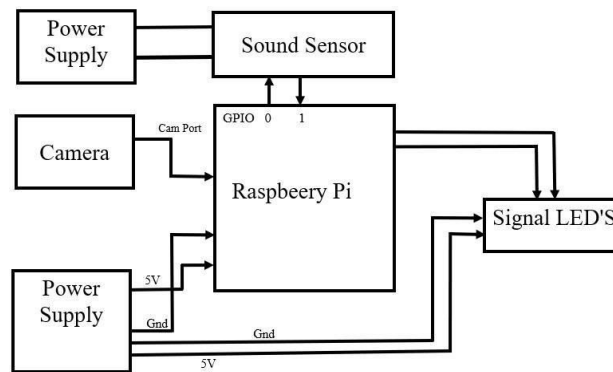


Fig.2 Block Diagram

A Traffic Light System using Raspberry Pi and OpenCV can be designed to manage traffic efficiently by detecting vehicles, pedestrians, or sound signals. The provided block diagram represents a simplified system where a Raspberry Pi controls signal LEDs based on input from a sound sensor and a camera module.

Pi is a credit-card sized computer that connects to a computer monitor or TV and uses input devices like keyboard and mouse. It is capable of performing various functionalities such as surveillance system, military applications, surfing internet, playing high definition videos, live games and to make databases. The device is implemented using a Raspberry pi 3B board and their specifications are as follows. Raspberry Pi is a credit-card sized computer manufactured and designed in the United Kingdom by the Raspberry Pi foundation with the intention of teaching basic computer science to school students and every other person interested in computer hardware, programming and DIY-Do-it Yourself projects. The Raspberry Pi is manufactured in three board configurations through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Egoman. These companies sell the Raspberry Pi online. Egoman produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pi is by their red colouring and lack of FCC/CE marks. The hardware is the same across all manufacturers. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU and was originally shipped with 256 bytes of RAM, later upgraded (Model B & Model B+) to 512 MB. It does not include a built-in hard disk or solid-state drive, but it uses an SD card for booting and persistent storage, with the Model B+ using a MicroSD. The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, Java and Perl.



Fig.3 Raspberry pi 3 model b+

The Raspberry Pi 3 Model B+ contains a wide range of improvements and features that will benefit the designers, developers, and even engineers who are looking to integrate Pi systems into their products. Here are some of the new Pi's specs: Quad core 64-bit processor clocked at 1.4GHz. 1GB LPDDR2 SRAM. Dual-band 2.4GHz and 5GHz wireless LAN. Bluetooth 4.2 / BLE. Higher speed ethernet up to 300Mbps. Power-over-Ethernet capability (via a separate PoE HAT). While the mechanical layout of the Pi has not changed (GPIO location, drill holes, etc.), the PCB itself has clearly undergone some physical changes. The main processor is no longer housed in a plastic package. Instead, it has a metal package, which may be beneficial for those who want to keep the temperature of the Pi as low as possible (with the aid of a heat sink). The top side also shows fewer components, and a four-pin header (used for PoE) has been included in the top right of the PCB. While the mechanical layout of the Pi has not changed (GPIO location, drill holes, etc.), the PCB itself has clearly undergone some physical changes. The main processor is no longer housed in a plastic package. Instead, it has a metal package, which may be beneficial for those who want to keep the temperature of the Pi as low as possible (with the aid of a heat sink). The top side also shows fewer components, and a four-pin header (used for PoE) has been included in the top right of the PCB.

Technology has been evolved over time with the purpose of making lives easy and convenient. This device was a major development in the technology that made computer learning too easy that anyone with little effort can make their feet wet with the process. In this tutorial, I'll discuss each and everything related to Raspberry Pi 3 B+, its main functions and features, benefits and everything you need to know, so you find all information in one place without wrestling your mind on the web surfing. Let's dive right in.

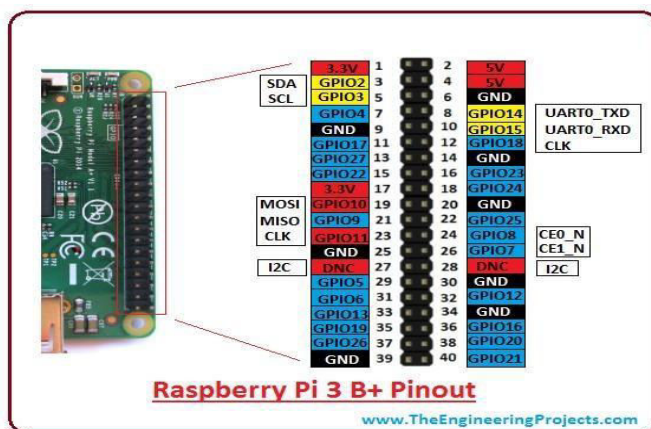


Fig.4 Raspberry pi pin diagram

40 Pin header is used to develop an external connection with the electronic device. This is the same as the previous versions, making it compatible with all the devices where older versions can be used. Out of 40 pins, 26 are used as a

digital I/O pins and 9 of the remaining 14 pins are termed as dedicated I/O pins which indicate they don't come with alternative function. Pin 3 and 5 comes with an onboard pull up resistor which 1.8 k Ω and Pin 27 and 28 are dedicated to ID EEPROM. In B+ model the GPIO header is slightly repositioned to allow more space for the additional mounting hole. The devices that are compatible with the B model may work with the B+ version; however, they may not sit identically to the previous version.

V. HARDWARE SPECIFICATIONS:

CPU: The CPU is a brain of this tiny computer that helps in carrying out a number of instruction based on the mathematical and logical formulas. It comes with a capacity of 64 bit. **Clock Speed and RAM:** It comes with a clock speed of 1.4 GHz Broadcom BCM2837B0 that contains quad-core ARM Cortex-A53 and RAM memory is around 1GB (identical to the previous version) **GPU:** It stands for graphics processing unit, used for carrying out image calculation. Broadcom video core cable is added in the device that is mainly used for playing video games. **USB Ports:** Two more USB ports are introduced in this new version, setting you free from the hassle of using an external USB hub when you aim to join a number of peripherals with the device. **Micro USB Power Source Connector:** This connector is used for providing 5V power to the board. It draws 170 to 200mA more power than B model. **HDMI and Composite Connection:** Both audio output socket and video composite now reside in a single 4-pole 3.5mm socket which resides near HDMI. And the power connector is also repositioned in new B+ model and lives next to HDMI socket. All the power and audio video composite socket are now placed on the one side of the PCB, giving it a clean and precise look. **USB Hard Drive:** The USB hard drive is available on the board that is used to boot the device. It is identical to the hard drive of regular computer where windows is used to boot the hard drive of the computer. **PoE:** B+ model comes with a facility of Power over Ethernet (PoE); a new feature added in this device which allows the necessary electrical current using data cables. **Other Changes:** The B+ version comes with little improvement in the features and poses slightly different layout in terms of location of the components. The SD memory slot is replaced by a micro SD memory card slot (works similar to the previous version). The status LEDs now only contain red and green colour and relocated to the opposite end of the PCB.

Camera: The Quantum QHM495LM is a webcam produced by Quantum Hi-Tech, an Indian technology company. Here are some details about the Quantum QHM495LM camera: **Resolution:** The Quantum QHM495LM typically offers a resolution of 640x480 pixels (VGA resolution). **Sensor:** It features a CMOS sensor. **Frame Rate:** The frame rate of the camera is typically around 30 frames per second (fps), which is suitable for video calling and basic video recording. **Interface:** It usually connects to the computer via a USB 2.0 interface, which provides compatibility with most computers and laptops. **Microphone:** Some variants of the Quantum QHM495LM come with a built-in microphone, enabling audio along with video during video calls. **Adjustability:** It often comes with an adjustable clip or stand, allowing users to mount it on their computer monitors or place it on their desks as needed. **Compatibility:** The camera is generally compatible with various operating systems, including Windows, macOS, and Linux, making it versatile for different users.

VI. RESULTS

The process of building a traffic light detection system using Raspberry Pi and OpenCV with YOLOv5. It starts with data acquisition by capturing traffic light images, followed by annotation to label red, yellow, and green lights. YOLOv5 configurations are set for model training on a Raspberry Pi- compatible dataset.

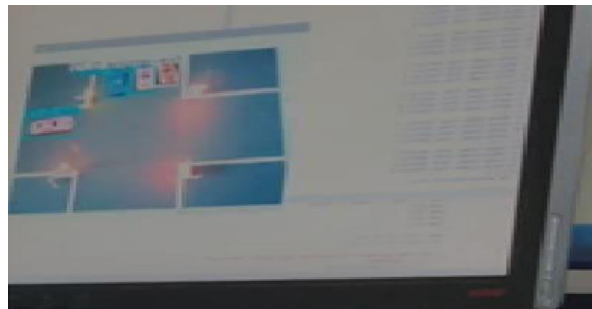
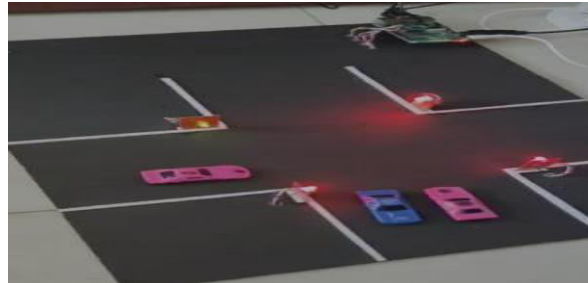


Fig.5 Circuit Connection and Yolo V5 Vehicle detected

The model is then trained to recognize traffic light states. After detection, the system either classifies the light or updates the model with new data for accuracy. This process ensures real-time, automated traffic light detection suitable for smart traffic systems or autonomous vehicle support. A traffic light system using Raspberry Pi and OpenCV offers a sophisticated solution for intelligent traffic management, moving beyond traditional timed systems to a more adaptive approach. The core idea involves utilizing a camera (e.g., Raspberry Pi Camera Module or USB webcam) to capture real-time video footage of traffic intersections. This video feed is then processed by the Raspberry Pi, acting as the central processing unit, using the OpenCV library for computer vision tasks.

OpenCV plays a crucial role in detecting and counting vehicles by applying various image processing techniques such as background subtraction, edge detection (e.g., Canny), and contour analysis. Some advanced systems might even incorporate machine learning models like SVM or YOLO for more accurate vehicle classification and detection. Based on the calculated traffic density for each lane or direction, the Raspberry Pi dynamically adjusts the timing of the traffic lights. For instance, a lane with higher vehicle density will be allocated a longer green light duration to reduce congestion and improve traffic flow.

The physical traffic lights, typically represented by LEDs (red, yellow, green), are connected to the Raspberry Pi's GPIO pins, often via a breadboard and sometimes through relay modules for controlling higher voltage lights. Python programming is commonly used to write the logic that interprets the traffic density data from OpenCV and sends the appropriate signals to the GPIO pins to switch the LEDs on or off. This allows for a system that can significantly reduce waiting times, lower fuel consumption, and enhance road safety by adapting to real-time traffic conditions, rather than relying on fixed, inefficient timings.

VII. CONCLUSION

Our project successfully demonstrated the implementation of a traffic light system utilizing a Raspberry Pi and OpenCV. The system effectively identifies the presence of vehicles through image processing, allowing for dynamic control of traffic signals rather than relying on fixed timers. This intelligent approach significantly improves traffic flow efficiency by reducing unnecessary waiting times at intersections. The use of a cost-effective and compact Raspberry Pi, coupled with the powerful computer vision capabilities of OpenCV, makes this a viable and scalable solution for modernizing urban traffic management infrastructure. Future enhancements could include pedestrian detection, integration with broader smart city platforms, and real-time data analytics for further optimization.

VIII. FUTURE SCOPE

Further enhancements include expanding the system to cover multiple intersections and enabling intercommunication between them. This would allow for synchronized traffic light timings across a network, preventing bottlenecks and optimizing overall city-wide traffic flow, rather than just isolated intersections. Cloud-based data storage and analytics could be integrated to process vast amounts of traffic data, identify long-term patterns, predict congestion, and inform urban planning and infrastructure development. The project could also incorporate Internet of Things (IoT) principles, connecting with other smart city infrastructure like environmental sensors (for air quality or noise levels) or public transport systems to create a more holistic traffic management approach. Vehicle-to-Infrastructure (V2I) communication presents a significant future direction, where vehicles could directly communicate with traffic lights to receive real-time signal phase and timing information, allowing them to adjust their speed and behaviour for smoother passage and reduced fuel consumption. This is particularly crucial for the development of autonomous vehicles.

REFERENCES

1. "An Automated Vehicle Counting System Based on Blob Analysis for Traffic Surveillance" by G. Salvi. IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 2, (2006), pp. 188-200.
2. "Coordinated Road Junction Traffic Control by Dynamic Programming" by Tsin Hing Heung, IEEE Intelligent Transportation Systems Conference, 2007, pp. 678-683. R.
3. "Intelligent Traffic Signaling System" by Ch. Jaya Lakshmi S. Kalpana, IEEE Trans. Intel. Transport. Syst., vol. 1, no. 2, pp. 119-130, June 2002.
4. "Dynamic Traffic Management System Using Infrared Red(IR) and Internet of Things (IoT)" by Paul Jasmine Rani, Khoushik Kumar, IEEE transport system conference, june 2016.
5. "Intelligent Cross Road Traffic Management System" by Ahmed S. Salama, IEEE Intelligent Transportation Systems Conference, 2009.
6. Singh., et al. "Traffic Signal Control Using Sound Sensors" IJCRT, 2021
7. Rani, Paul Jasmine, and Khoushik Kumar. "Dynamic Traffic Management System Using Infrared Red (IR) and Internet of Things (IoT)."IEEE,2018.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details