DHA Suffa University

IOT based Smart Traffic Signal System

"A report submitted as final year project interim report
in the department of electrical engineering"

by

Hafiz Usman Sohail EE181001

Muhammad Fuzail Qureshi EE181002

Thursday, 14th September2023

Project Supervisor:

Haris Zik-ur-Rehmen

Table of Contents

List of Figures

ABSTRACT

As the rate of urbanization continues to rise, the problem of traffic management will become more important and will call for creative solutions. This research provides a revolutionary approach to the automation of traffic signals by capitalizing on the combination of cutting-edge computer vision methods and technology that is enabled by the Internet of Things (IoT). Two specialized nodes, each of which is outfitted with the powerful YOLOv3 object detection model, have been painstakingly created in order to distinguish and measure the volume of vehicle traffic at signal intersections with an accuracy that is unmatched. The information that has been identified is then sent in a smooth manner to a central server that is housed on a Raspberry Pi, which results in the construction of a network that is nimble and responsive. This server acts as the command center and is responsible for coordinating the cycle of the traffic signals depending on the current real-time circumstances. The system optimizes traffic flow dynamically, hence reducing congestion and increasing overall urban mobility, by intelligently distributing green light time in direct connection to the observed vehicle density. This helps the system mitigate congestion and improves overall urban mobility. Not only does this comprehensive combination of cutting-edge computer vision with the Internet of Things connection exemplify a jump in technical innovation, but it also represents a paradigm change in the way that intelligent and responsive traffic management systems operate. The ramifications of this system are significant, and it has the potential to bring about a revolution in the infrastructure of urban transportation and to establish new standards of urban sustainability, efficiency, and safety.

1. INTRODUCTION

The extraordinary increase in automobile traffic that has been ushered in as a result of urbanization has presented urban planners and traffic management agencies all over the globe with significant issues [1]. It is becoming more necessary to have traffic management systems that are both efficient and able to react to changing conditions as cities continue to develop and their populations increase. Conventional traffic signal systems, which are often regulated by pre-defined time schedules, find it difficult to adapt to the dynamic nature of traffic patterns. As a consequence, this results in chronic congestion, longer commuting times, and an increased environmental effect. This project aims to solve this significant problem by using an innovative combination of cutting-edge computer vision methods and Internet of Things (IoT) technology [2-4]. The eventual result of this attempt will be an automated traffic light system that is capable of providing a real-time response to changes in the flow of traffic [5].

The use of the YOLOv3 (You Only Look Once) object detection model, which is recognized for its extraordinary speed and accuracy in recognizing and categorizing objects in photos, is essential to the success of this attempt. YOLOv3 is used in this scenario for the purpose of precisely identifying and quantifying the number of cars that are present at signalized junctions. Because it makes use of specialized nodes that are outfitted with this cutting-edge detection model, the system is able to attain a degree of accuracy that has not been seen before in traditional methods of traffic monitoring. These nodes function as the system's eyes, gathering and processing visual input in real-time. As a result, they create the basis upon which future choices about traffic management are founded [6].

This automated traffic signal system utilizes a seamless integration of Internet of Things (IoT) technology, which acts as the nervous system. This integration enables quick and reliable communication between the vehicular detecting nodes and a centralized server. This server, which is housed on a Raspberry Pi, operates as the center for decision making by analyzing the incoming data streams and coordinating the signal cycle in accordance with those interpretations. Even in situations with a high data flow, the system is able to reliably and effectively share information because it makes use of common Internet of Things communication protocols [7]. One example of such a protocol is MQTT, which stands for Message Queuing Telemetry Transport. This infrastructure, which is enabled by the Internet of Things, not only makes it easier for the system to respond in real-time but also paves the way for possible scalability and flexibility to a wide variety of urban settings [8].

In this fully automated traffic light system, the Raspberry Pi, which is famous for its adaptability and processing capability even in limited settings, plays a crucial part. In addition to playing the role of the primary server, the Raspberry Pi also plays the role of a safe and dependable platform that is used to house the necessary software and algorithms [9]. This device, which is both small and powerful, plays an important role in the processing of the incoming data streams from the vehicular detecting nodes, the execution of the required calculations, and the generation of the relevant control signals to govern traffic flow. Because of its low power consumption and cost-effectiveness, it is a great choice for large-scale deployment, and it holds the potential to revolutionize traffic control systems on a scale that is applicable to a whole city [10].

In conclusion, this project proposes a comprehensive and synergistic approach to the automation of traffic signals. It addresses the modern issues of urban traffic management by integrating cutting-edge computer vision and Internet of Things technologies. This cutting-edge system will usher in a new age of intelligent and responsive traffic control thanks to the convergence of vehicle detection nodes based on YOLOv3, a central server based on Raspberry Pi, and IoT communication protocols [11]. This convergence will constitute the backbone of this unique system. The succeeding parts are going to go into the technical subtleties of each component, illuminating the design considerations, implementation details, and performance evaluations, which will finally culminate in a full assessment of the system's efficacy in real-world urban areas [12].

2. BACKGROUND

The need for traffic management systems that are both effective and flexible is becoming more urgent as urban areas continue to expand. The traditional traffic signal systems, which are regulated by set time schedules, sometimes fail to accommodate the dynamic nature of traffic patterns. As a consequence, this results in chronic congestion, higher travel times, and an increased environmental impact. A paradigm shift toward creative solutions that are able to respond in real-time to changing traffic circumstances is required as a result of this need. The advent of computer vision technology, defined by breakthroughs in object identification and image processing, presents a potential route for transforming traffic management [13-14]. This might be accomplished in a number of ways. Among these breakthroughs, the YOLOv3 (You Only Look Once) object detection model has garnered international notice for its amazing speed and accuracy in distinguishing and categorizing objects inside photos. The model's name comes from the phrase "You only look once." This project aims to establish the foundation of an intelligent traffic management system by using the power of YOLOv3 in order to improve the accuracy and effectiveness of vehicle identification at signalized junctions. This will allow the system to function as the central pillar [15].

A new age of connectedness and data sharing has been ushered in as a result of the widespread adoption of Internet of Things (IoT) technology, which has occurred in parallel with these technological breakthroughs. The Internet of Things is a very important enabler for the traffic control system's many components to communicate with one another in a seamless manner. The system provides the degree of resilience and dependability essential for decision-making in real time via the adoption of established communication protocols such as MQTT. This is important for real-time decision-making. This use of Internet of Things technology not only contributes to the responsiveness of the system, but it also sets the groundwork for the system's future scalability and adaptation to a wide variety of urban settings [16].

This integrated system is improved further with the addition of the Raspberry Pi, which is a computer platform that is both flexible and affordable. In addition to its function as the primary server, the Raspberry

Pi plays a critical part in the processing and interpretation of the data streams as they come in. Because of its computing power in contexts with limited resources, it is a great choice for large-scale deployment, and it has the potential to revolutionize traffic management systems on a scale that is applicable to a whole city. This project aims to overcome the constraints of traditional traffic management systems by integrating these technological breakthroughs in a seamless manner [17]. As a result, it provides a dynamic and intelligent approach to the automation of traffic signals. The subsequent sections will delve into the technical complexities of each component, providing detailed insights into the design considerations, implementation specifics, and performance evaluations, which will ultimately culminate in a comprehensive assessment of the effectiveness of the system in actual world urban environments.

3. HARDWARE USED

Following are the hardware components used for the project development,

- Raspberry Pi 4
- Camera
- LEDs
- Rasbian as an Operating System

3.1. Raspberry Pi 4

The Raspberry Pi 4, released in June 2019, is the latest iteration of the highly popular single-board computer series developed by the Raspberry Pi Foundation. Building on the success of its predecessors, the Raspberry Pi 4 introduces significant advancements in performance, connectivity, and versatility, making it a versatile and powerful platform for a wide range of applications.

3.1.1. Key Specifications:

- Processor: The Raspberry Pi 4 is equipped with a quad-core Broadcom BCM2711 processor based on the ARM Cortex-A72 architecture. Running at 1.5 GHz, this processor provides a substantial increase in processing power compared to earlier models.
- RAM: It is available in various configurations, with options for 2GB, 4GB, or 8GB of LPDDR4 SDRAM. This expanded memory capacity enables smoother multitasking and more demanding applications.
- Graphics: The Pi 4 features a VideoCore VI GPU, offering improved graphics performance for tasks ranging from high-definition video playback to 3D rendering.
- Connectivity: It boasts comprehensive connectivity options, including two USB 3.0 ports, two USB 2.0 ports, Gigabit Ethernet, dual-band Wi-Fi (802.11ac), and Bluetooth 5.0. These enhancements greatly expand the range of peripherals and devices that can be connected to the Raspberry Pi 4.
- Video Output: The Pi 4 supports dual 4K display output via two micro HDMI ports. This capability is particularly valuable for applications that require high-resolution visual output.
- Storage: It employs a microSD card slot for primary storage, offering ample space for the operating system and applications. Additionally, USB and network-based storage options can be utilized for expanded storage capacity.
- Power: The Raspberry Pi 4 requires a 5V USB-C power supply. The power requirements are slightly higher compared to previous models due to the increased processing capabilities.

3.1.2. Operating Systems:

The Raspberry Pi 4 is compatible with a wide range of operating systems, including Raspbian (now known as Raspberry Pi OS), Ubuntu, and various Linux distributions. It can also run specialized operating systems for specific applications, such as media centers, retro gaming consoles, and robotics platforms.

- **Use Cases:**

The enhanced processing power and expanded memory of the Raspberry Pi 4 make it suitable for a diverse array of applications. These include but are not limited to:

- Desktop Computing: With its improved performance, the Pi 4 can serve as a basic desktop computer for web browsing, word processing, and other office tasks.
- Media Center: It can function as a capable media center, capable of streaming high-definition video and supporting a variety of multimedia formats.
- IoT and Embedded Systems: Its small form factor and GPIO pins make it an ideal platform for IoT projects, sensor interfacing, and home automation.
- Educational Tools: The Raspberry Pi 4 continues the tradition of providing an accessible and affordable platform for learning programming and computer science.

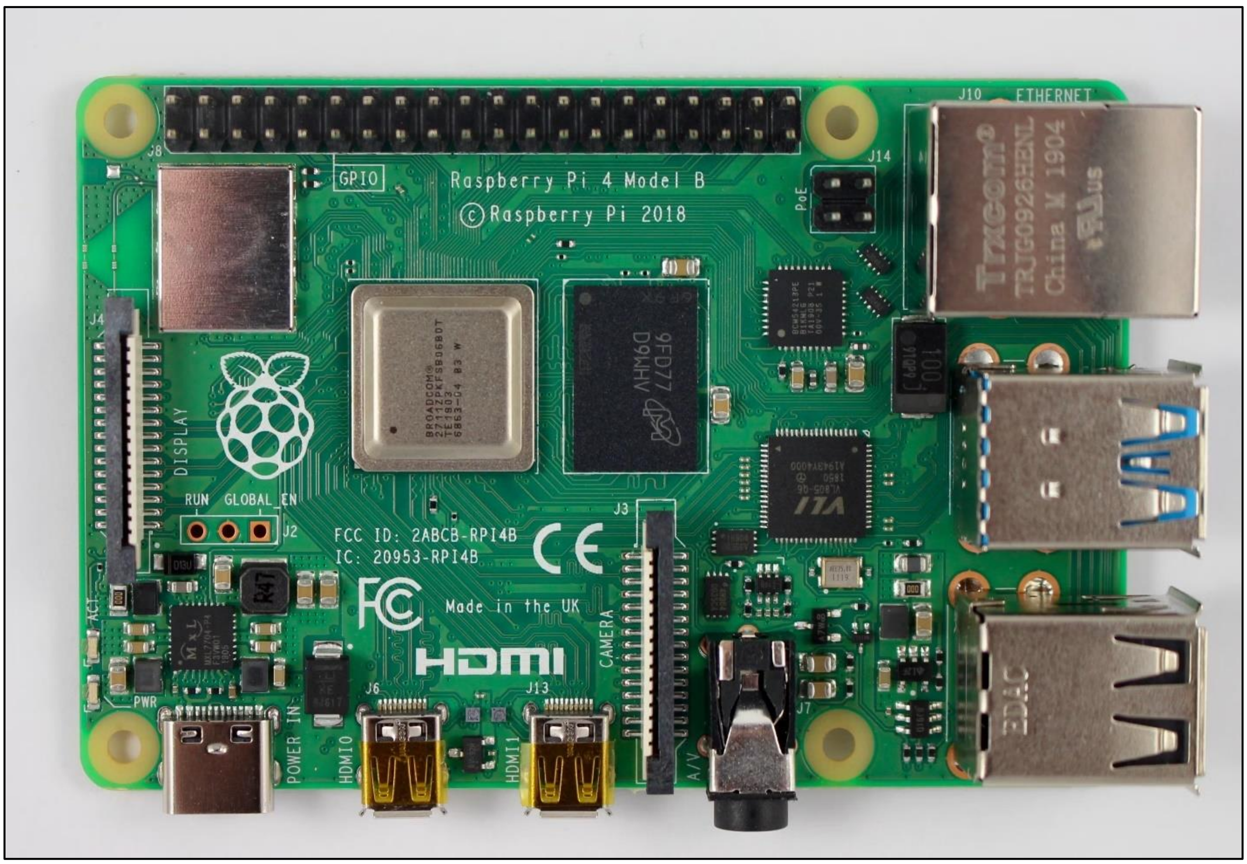


Fig.1. Raspberry Pi 4 Model

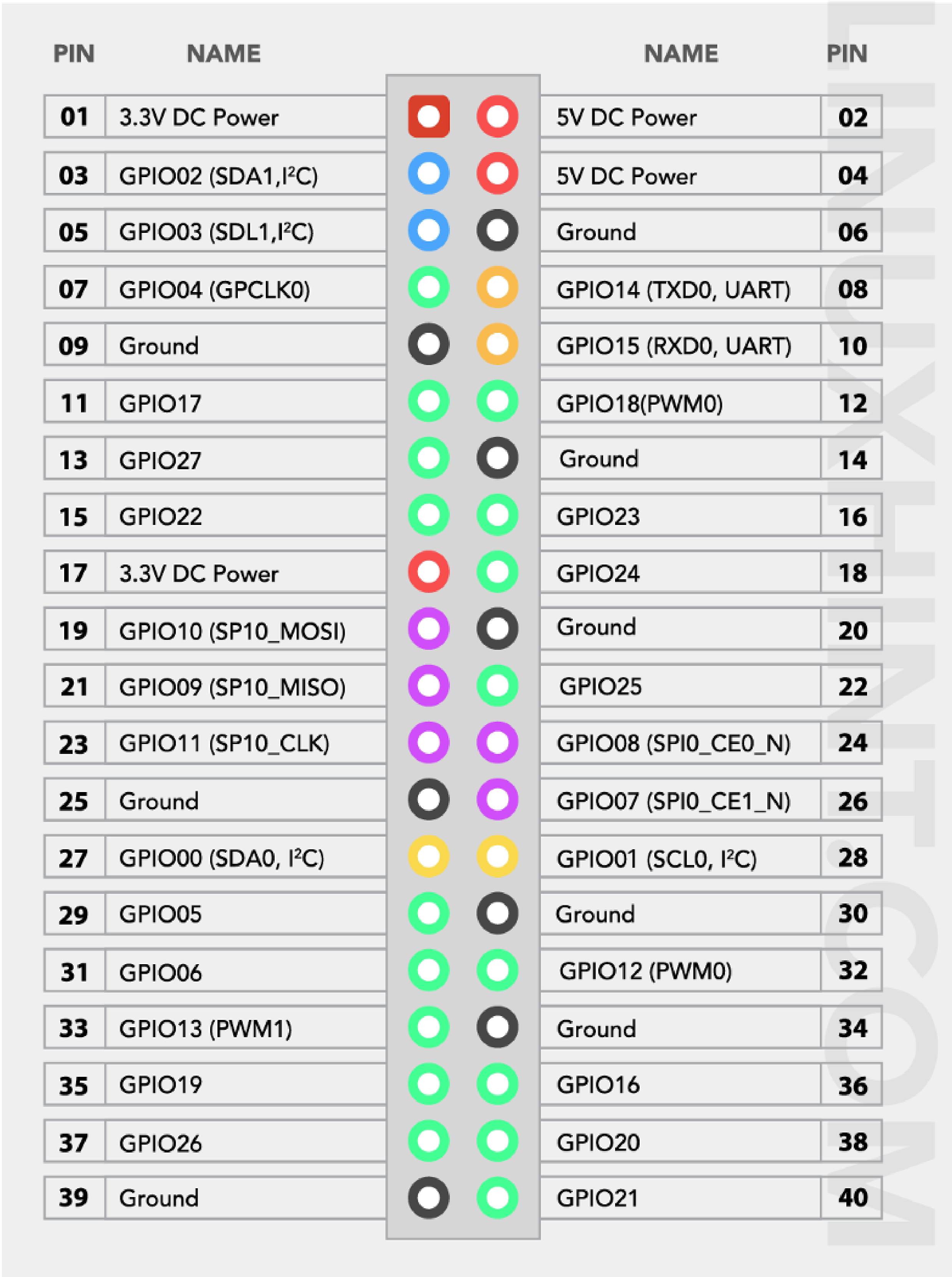


Fig. 2. Raspberry Pi Pin Configuration

- **Pin Numbering:**
  - The GPIO pins on the Raspberry Pi 4 are labeled with both physical pin numbers and GPIO numbers. It's important to note that there are two different numbering schemes:
    - **Physical Pin Numbering:** This refers to the actual pin on the Raspberry Pi's GPIO header.
    - **BCM GPIO Numbering:** This numbering scheme is based on the Broadcom SOC channel numbers. These are the numbers used in programming to interact with the GPIO pins.
- **Power Pins:**
  - **3.3V (Pin 1):** This pin provides a 3.3V power supply, which can be used to power low-voltage components.
  - **5V (Pin 2):** This pin provides a 5V power supply, which can be used to power higher-voltage components.
- **Ground Pins:**
  - **GND (Pins 6, 9, 14, 20, 25, 30, 34, 39):** These pins are ground connections, providing a reference voltage for circuits.
- **GPIO Pins:**
  - There are a total of 40 GPIO pins on the Raspberry Pi 4, labeled as GPIO0 through GPIO39 in BCM numbering.
  - These pins can be used for both input (reading signals) and output (sending signals) purposes.
- **I2C Pins:**
  - **SDA (Pin 3):** This is the data line for I2C communication.
  - **SCL (Pin 5):** This is the clock line for I2C communication.
- **SPI Pins:**
  - **MOSI (Pin 19):** Master Out Slave In for SPI communication.

- **MISO (Pin 21):** Master In Slave Out for SPI communication.
- **SCLK (Pin 23):** Serial Clock for SPI communication.
- **CE0 (Pin 24):** Chip Enable 0 for SPI communication.
- **CE1 (Pin 26):** Chip Enable 1 for SPI communication.

- **UART Pins:**
  - **TXD (Pin 8):** Transmit Data for UART communication.
  - **RXD (Pin 10):** Receive Data for UART communication.
- **PWM Pins:**
  - **PWM0 (Pin 12):** Pulse Width Modulation (PWM) output.
  - **PWM1 (Pin 32):** Pulse Width Modulation (PWM) output.
- **Other Pins:**
  - **ID_SD (Pin 27):** I2C ID EEPROM (used to identify the board and load the correct drivers).
  - **ID_SC (Pin 28):** I2C ID EEPROM (used to identify the board and load the correct drivers).
- **Reserved Pins:**
  - Some pins are reserved for specific functions (e.g., EEPROM, HAT ID) and should not be used for general-purpose applications.

3.2. Camera

The camera component is an integral aspect of this project, facilitating real-time visual data acquisition and processing. The camera employed is a high-resolution module designed to interface seamlessly with the Raspberry Pi 4, leveraging its capabilities for a diverse range of applications. It is used to capture the live video stream of traffic with can be used for a variety of purposes.

Fig. 3. Camera

3.3. LEDs

Light Emitting Diodes, frequently referred to simply as LEDs, are semiconductor devices that, when an electric current is sent through them, produce light. They have a positive lead (the anode) and a negative lead (the cathode), and in order for them to operate, they must be linked in the appropriate direction. LEDs are available in a wide range of hues, including red, green, blue, and white, with each color requiring a unique combination of semiconductor elements to create. They are often used in the electronic industry for the purpose of providing a visible signal, such as in the form of power indicators or status lights. In comparison to conventional incandescent bulbs, light-emitting diodes, or LEDs, have a longer lifetime, require less power, and are more resilient.

Fig. 4. LEDs

3.4. Rasbian as an Operating System

- Purpose-Built for Raspberry Pi:

Raspbian is developed and maintained by the Raspberry Pi Foundation, with the primary goal of providing an operating system that is tailored to the hardware and capabilities of Raspberry Pi devices.

- Debian-Based Linux Distribution:

Raspbian is derived from Debian, a popular and widely used Linux distribution known for its stability, extensive software repositories, and strong community support.

- Optimized Performance:

Raspbian is optimized to run efficiently on Raspberry Pi hardware, making the most of its processing power and memory capabilities.

- Desktop Environment:

Raspbian includes a lightweight desktop environment, typically based on the LXDE or LXQt desktop environments. This provides users with a graphical user interface (GUI) for easy navigation and interaction with the system.

- Package Management:

Raspbian uses the Debian package management system, which allows users to easily install, update, and manage software packages using package managers like apt.

- Vast Software Repository:

Raspbian provides access to a vast repository of pre-compiled software packages, which cover a wide range of applications and utilities. This makes it easy for users to install additional software on their Raspberry Pi.

- GPIO Access:

Raspbian includes tools and libraries that allow users to interact with the GPIO pins of the Raspberry Pi, enabling the development of hardware projects and IoT applications.

- Education and Learning:

Raspbian is well-suited for educational purposes and serves as an excellent platform for learning programming, electronics, and computer science. It comes pre-installed with educational software and tools.

- Regular Updates and Support:

Raspbian receives regular updates and security patches, ensuring that users have access to the latest features and improvements. The Raspberry Pi community and forums provide additional support and resources.

- Customizations and Configurations:

Users have the flexibility to customize and configure Raspbian to suit their specific needs, allowing for a tailored computing experience.

- Compatibility:

Raspbian is designed to be compatible with a wide range of Raspberry Pi models, from the earliest versions to the latest releases.

- Variants of Raspbian:

There are different variants of Raspbian available, including the full desktop version (with GUI), a lite version (without GUI, suitable for headless setups), and specialized variants like Raspbian for Robots.

4. HARDWARE AND SOFTWARE IMPLEMENTATION

The hardware solution for this project is centered on the incorporation of two specialized vehicle-detecting nodes, each of which is powered by a Raspberry Pi 4 and comes equipped with an onboard camera [18]. The powerful YOLOv3 object identification model is used by these nodes, which play an important part in the process of obtaining real-time visual data of the road. This allows the nodes to precisely determine and quantify the number of cars that are present at signalized junctions [19-23]. The YOLOv3 model, which is recognized for its excellent speed and accuracy, is used to analyze the acquired photographs in order to provide exact information on the density of the traffic. These vehicle-detecting nodes are built with Internet of Things characteristics, which enables them to interact fluidly with the central server that is located on a separate Raspberry Pi 4 device. The MQTT protocol, which is a lightweight and efficient messaging system that is perfect for Internet of Things applications, is what enables communication to take place. Because of this contact, real-time updates on the traffic situation are guaranteed, which is an essential component of the project's operation. The server software, which is stored on the Raspberry Pi, is in charge of receiving, analyzing, and processing the data that is sent by the vehicle-detecting nodes. It makes judgments on the management of the traffic signals based on this data in order to be as informed as possible. The Raspberry Pi server improves overall urban mobility by optimizing traffic flow. It does this by dynamically modifying the signal cycle depending on the observed vehicle density. This helps to alleviate congestion and improves overall urban mobility. In addition, suitable power management systems and secure mounting enclosures are installed in order to guarantee steady functioning and shield the components from the effects of the surrounding environment. The integration of these different pieces of hardware in such a way as to be carefully choreographed constitutes the basis of an intelligent traffic management system, which is on the verge of revolutionizing the infrastructure of urban transportation [24-25].

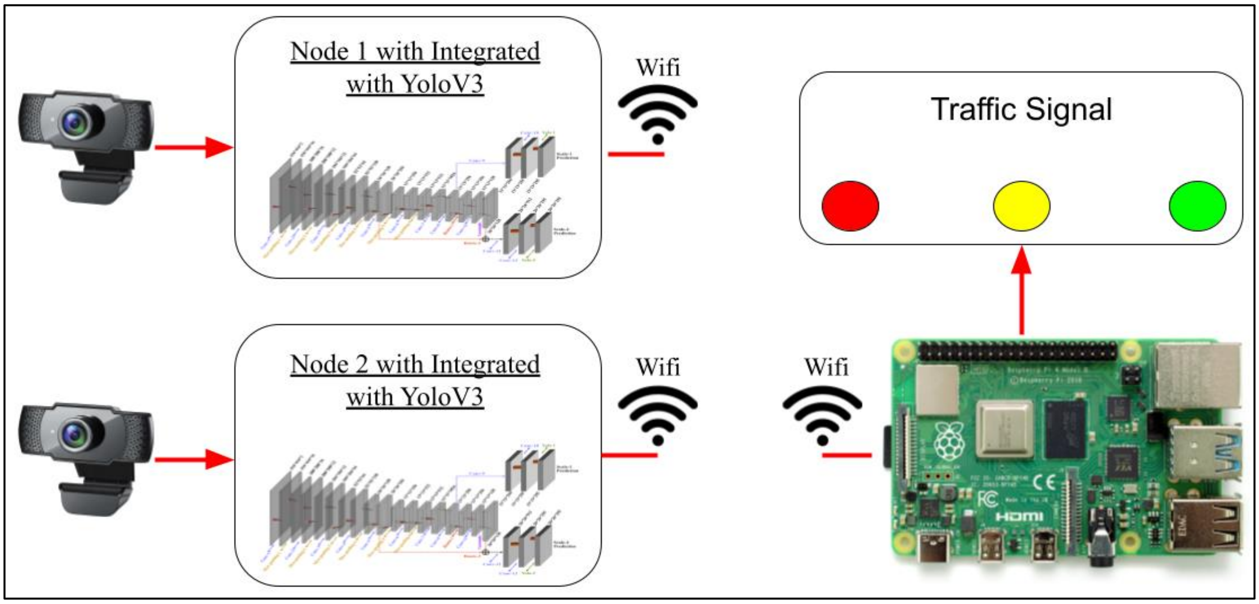


Fig. 5. Hardware Implementation using Internet of Things (IoT)

4.1. YoloV3

You Only Look Once, more usually abbreviated as YOLO, is an innovative strategy for object recognition that was developed in the field of computer vision. YOLO is a programming language that was created by Joseph Redmon and Ali Farhadi. It is notable for both its efficiency and speed, which makes it especially well-suited for real-time applications. The third version of YOLO, also known as YOLOv3, is an improvement over previous versions of the software in terms of both its accuracy and its performance [18].

YOLOv3 fundamentally changes the way object detection is approached by recasting it as a problem involving regression. Traditional approaches for detecting objects in images include dividing a picture into grids, applying sophisticated region recommendations, and creating classification networks. On the other hand, YOLO approaches object detection as a unified and comprehensive whole. It eliminates the need for numerous calculations by dividing the picture into a grid and predicting bounding boxes and class probabilities immediately. Because of this, the procedure of detection is made extraordinarily effective and simplified [17].

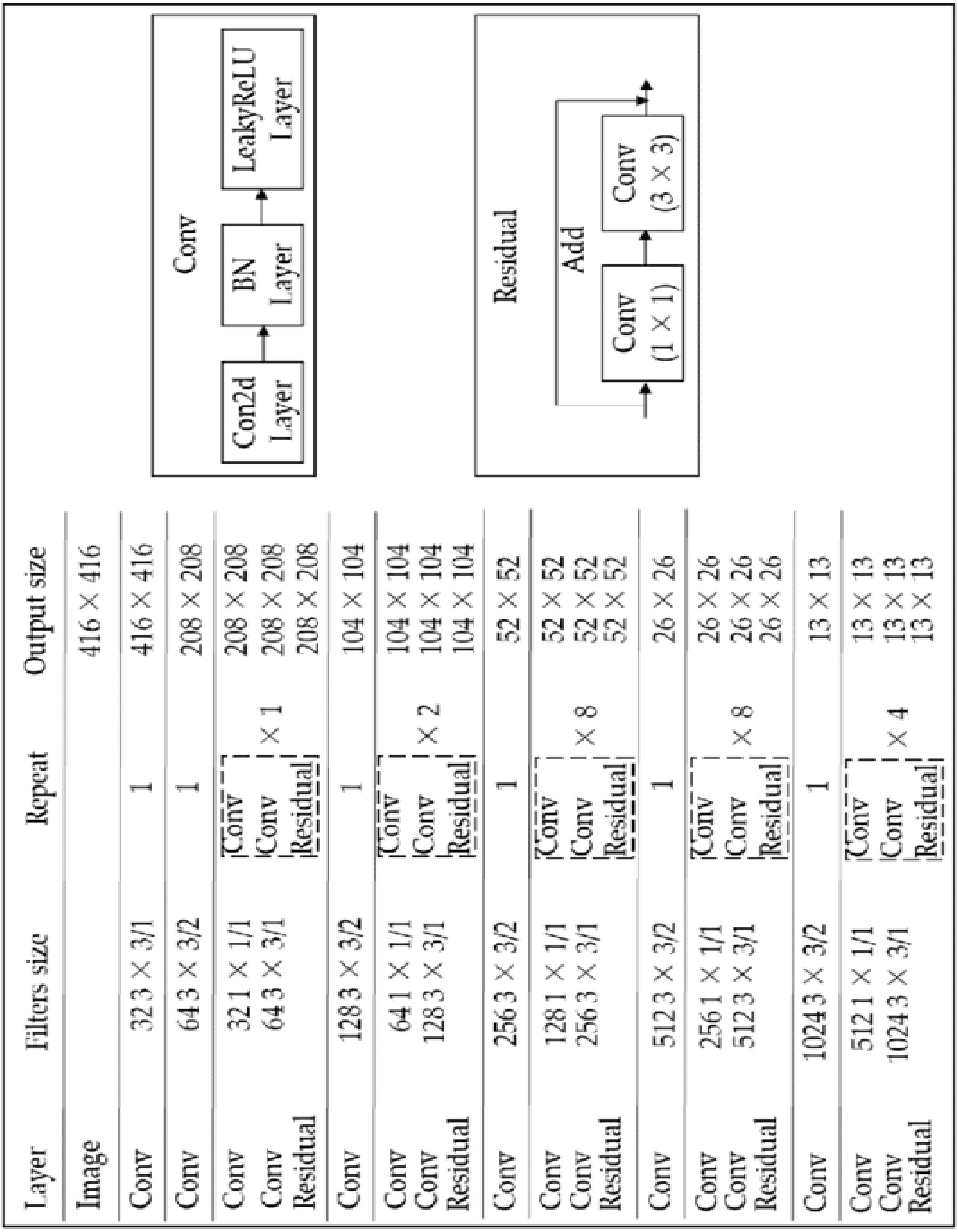
There are many distinguishing characteristics that define the YOLOv3 architecture. The first thing that it does is use a deep convolutional neural network as its backbone. This network is often a variation of Darknet. The task of extracting characteristics from the picture that is sent into this network is its responsibility. YOLOv3 makes use of a number of convolutional layers, which gives it the ability to learn hierarchical representations of objects at varying sizes. This is an essential capability for effective detection.

In addition, the YOLOv3 model makes use of various detection scales, a technique known as "multi-scale detection," to increase the model's capacity to recognize objects of varied sizes. Within the network, it employs detection at three distinct scales, so guaranteeing that tiny, medium, and big items are all reliably detected [26].

In addition to this, YOLOv3 implements a notion that is referred to as "anchor boxes." The model relies on these pre-defined bounding boxes, which might be of varying sizes and aspect ratios, to make accurate predictions about the placements of objects. Because it makes use of anchor boxes, YOLOv3 has the ability to conform to objects of varying sizes and forms, which contributes to the diversity of the system [19].

YOLOv3 has gained the capability of predicting item classes at different scales, which is another significant improvement. This indicates that it is able to recognize not just the existence of an item but also the particular kind or category of that object, which significantly improves its capabilities when dealing with complicated scenarios.

YOLOv3 is a pioneering object detection model that redefines the paradigm by approaching object detection as a single regression problem. In a nutshell, this makes YOLOv3 an innovative approach to object identification. Its cutting-edge design, which includes multi-scale detection, anchor boxes, and class predictions, makes it possible to identify objects in real-time settings with high accuracy and efficiency. Because of its ability to strike a good balance between speed and accuracy, YOLOv3 has become an essential component in a wide variety of computer vision applications. One such application is the identification of automobiles on roadways, which is what this project is focusing on [21].



Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv	32 1 × 1/1	Conv	208 × 208
Conv	64 3 × 3/1	Conv × 1	208 × 208
Residual		Residual	208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv	64 1 × 1/1	Conv	104 × 104
Conv	128 3 × 3/1	Conv × 2	104 × 104
Residual		Residual	104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv	128 1 × 1/1	Conv	52 × 52
Conv	256 3 × 3/1	Conv × 8	52 × 52
Residual		Residual	52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv	256 1 × 1/1	Conv	26 × 26
Conv	512 3 × 3/1	Conv × 8	26 × 26
Residual		Residual	26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv	512 1 × 1/1	Conv	13 × 13
Conv	1024 3 × 3/1	Conv × 4	13 × 13
Residual		Residual	13 × 13

Fig. 6. YoloV3 Pretrained Classifier

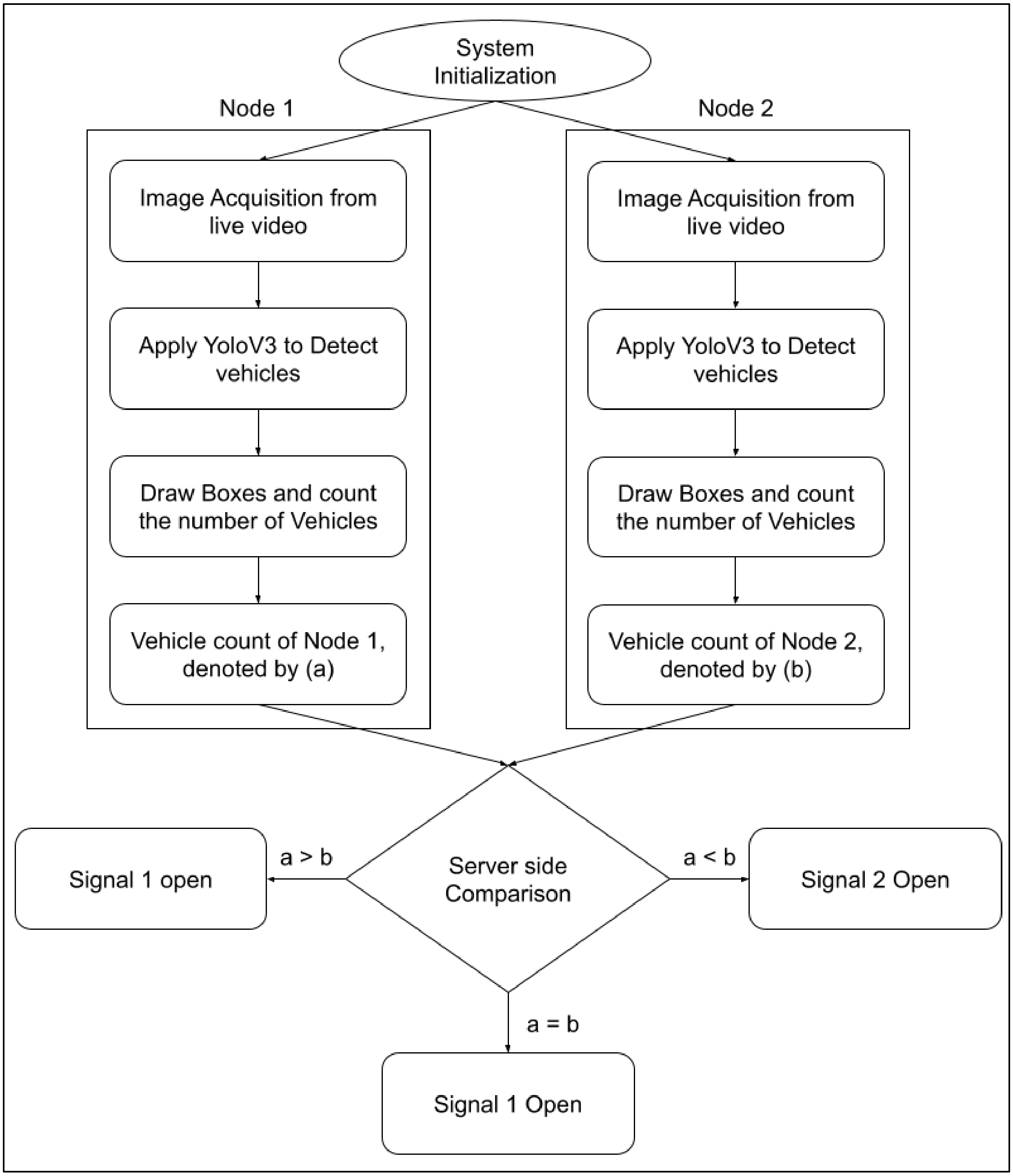


Fig. 7. System Flow Control

4.2. Internet of Things (IoT)

The Internet of Things, often known as IoT and usually abbreviated as "IoT," is a paradigm shift in terms of how technology and connection are conceptualized and used. It is a network of linked systems, gadgets, and sensors that interact with one another and exchange data through the internet. This dynamic network offers a broad variety of applications, ranging from industrial automation and smart homes to healthcare and urban infrastructure, which significantly alters the manner in which we interact with our surroundings.

The Internet of Things is based, fundamentally, on the seamless integration of various software and hardware components. The 'things' that make up the Internet of Things are defined as any electronic devices that have both sensors and communication modules. These sensors are able to collect a wide variety of data, including temperature and humidity readings, as well as data on motion and light levels. After the data has been gathered, it is then sent to a centralized hub or server via a variety of communication protocols, including Wi-Fi, Bluetooth, Zigbee, or cellular network connections [25].

The capability of these devices to function independently, often with just minimum input from a human operator, is essential to the usefulness of the Internet of Things. They are able to analyze data locally, come to judgments that are informed based on criteria that have been set, and even connect with other devices that are part of the network. This degree of independence gives Internet of Things systems the ability to react quickly to changes in their surroundings, which paves the way for real-time control and automation.

The capability of the Internet of Things to provide remote monitoring and control is one of its defining characteristics. This skill has a wide range of applications across a variety of business sectors, ranging from agriculture and healthcare to manufacturing and smart cities. For instance, sensors that are connected to the internet of things may be used in agriculture to monitor the moisture levels and temperature of soil. This gives farmers the ability to optimize irrigation and change planting dates. In a similar vein, Internet of Things-enabled medical devices in the field of healthcare may provide important patient data to healthcare practitioners in real time, so allowing prompt treatments.

When implementing IoT solutions, security and privacy concerns should be given top priority. Because of the vast number of interconnected devices that are constantly transferring sensitive data, it is vital to implement stringent security measures to guard against unauthorized access and breaches of data. When it comes to protecting the integrity of an Internet of Things system, encryption, secure authentication procedures, and consistent software upgrades are all essential components.

Additionally, one of the unique characteristics of IoT is its scalability. IoT ecosystems may consist of a very small number of devices all the way up to hundreds or even millions of nodes and devices. The Internet of Things can accommodate a wide variety of applications because to its scalability, ranging from individual smart home installations to enormous industrial deployments.

Fig. 8. Internet of Things (IoT)

5. RESULTS

YOLOv3, which stands for "You Only Look Once," is a state-of-the-art object detection model that excels in identifying objects within images or frames of a video. Its core innovation lies in its remarkable speed and accuracy, making it ideal for real-time applications, including the detection of objects like cars.

The YOLOv3 object detection process begins with the division of the input image into a grid of cells. Each cell is responsible for predicting objects within its boundaries. Within each cell, YOLOv3 predicts bounding boxes that encapsulate the objects it detects, along with the confidence score of the prediction.

Crucially, YOLOv3 is capable of identifying multiple objects in a single image, and it can predict the class (e.g., car, person, bicycle) of each detected object. This is achieved through the use of anchor boxes, which are predefined bounding boxes of varying sizes and aspect ratios. YOLOv3's network predicts the coordinates of these anchor boxes relative to each cell, and each anchor box is associated with a specific class of object.

To understand how YOLOv3 detects cars in an image:

Feature Extraction: YOLOv3 utilizes a deep convolutional neural network (CNN) as its backbone to extract features from the input image. These features capture various patterns, textures, and shapes that help in object recognition.

Bounding Box Prediction: For each grid cell, YOLOv3 predicts multiple bounding boxes (typically 3 or 9). These bounding boxes represent the potential locations of objects within the cell.

Object Confidence: For each predicted bounding box, YOLOv3 assigns a confidence score, indicating how likely it is that there is an object inside the box. High confidence values indicate a strong belief in the presence of an object.

Class Prediction: YOLOv3 also predicts the class probabilities for each bounding box. In the context of detecting cars, it assigns a higher probability to the "car" class if a car is present within the bounding box.

Non-Maximum Suppression: After predictions are made for all grid cells, YOLOv3 applies a technique called non-maximum suppression (NMS) to filter out duplicate and low-confidence detections. This ensures that only the most confident and accurate detections are retained.

The result is a set of bounding boxes, each associated with a confidence score and a class label. In the context of your project, YOLOv3 identifies and highlights the cars present in the image, providing their locations and class labels. This information can then be used for further analysis or action, such as traffic management based on the number of detected cars. YOLOv3's efficiency and accuracy make it a powerful tool for object detection in a wide range of applications, including traffic monitoring and control.

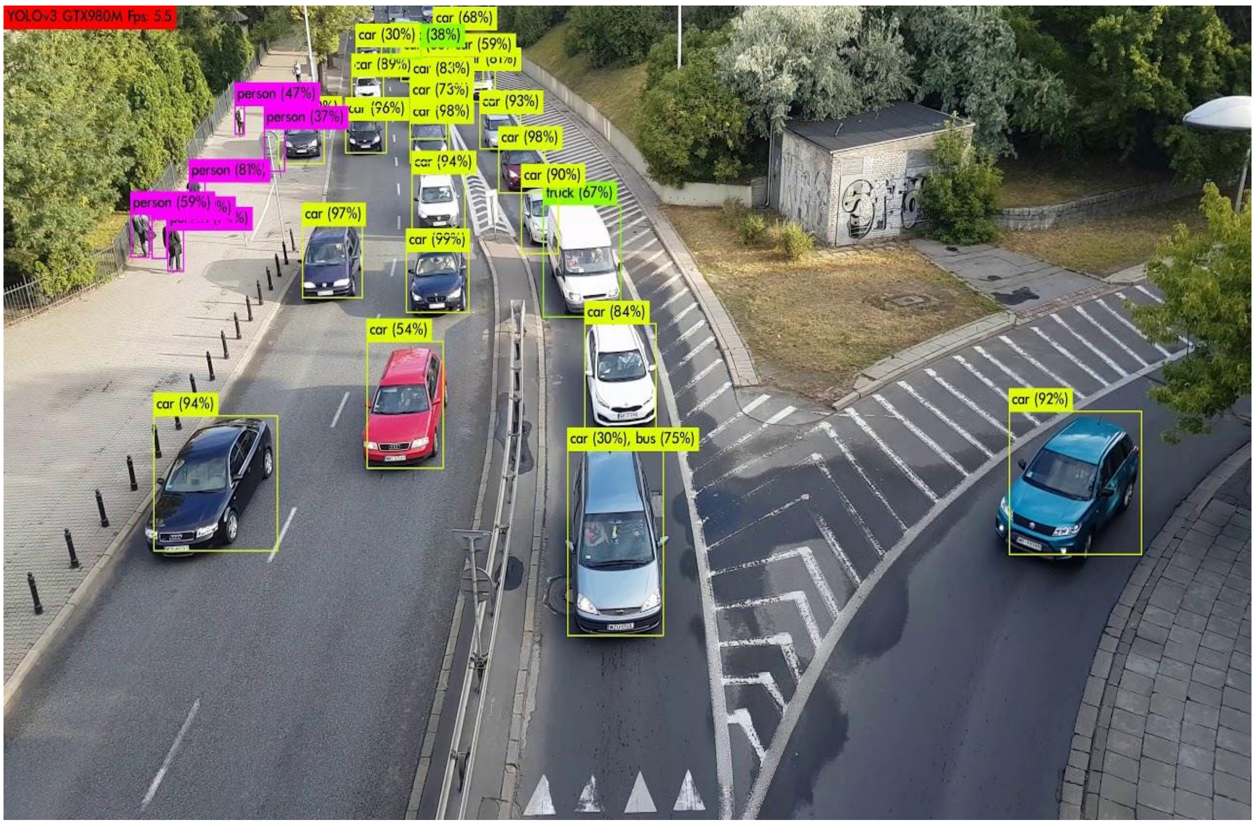


Fig. 9. Sample 1 output of YoloV3 pretrained classifier

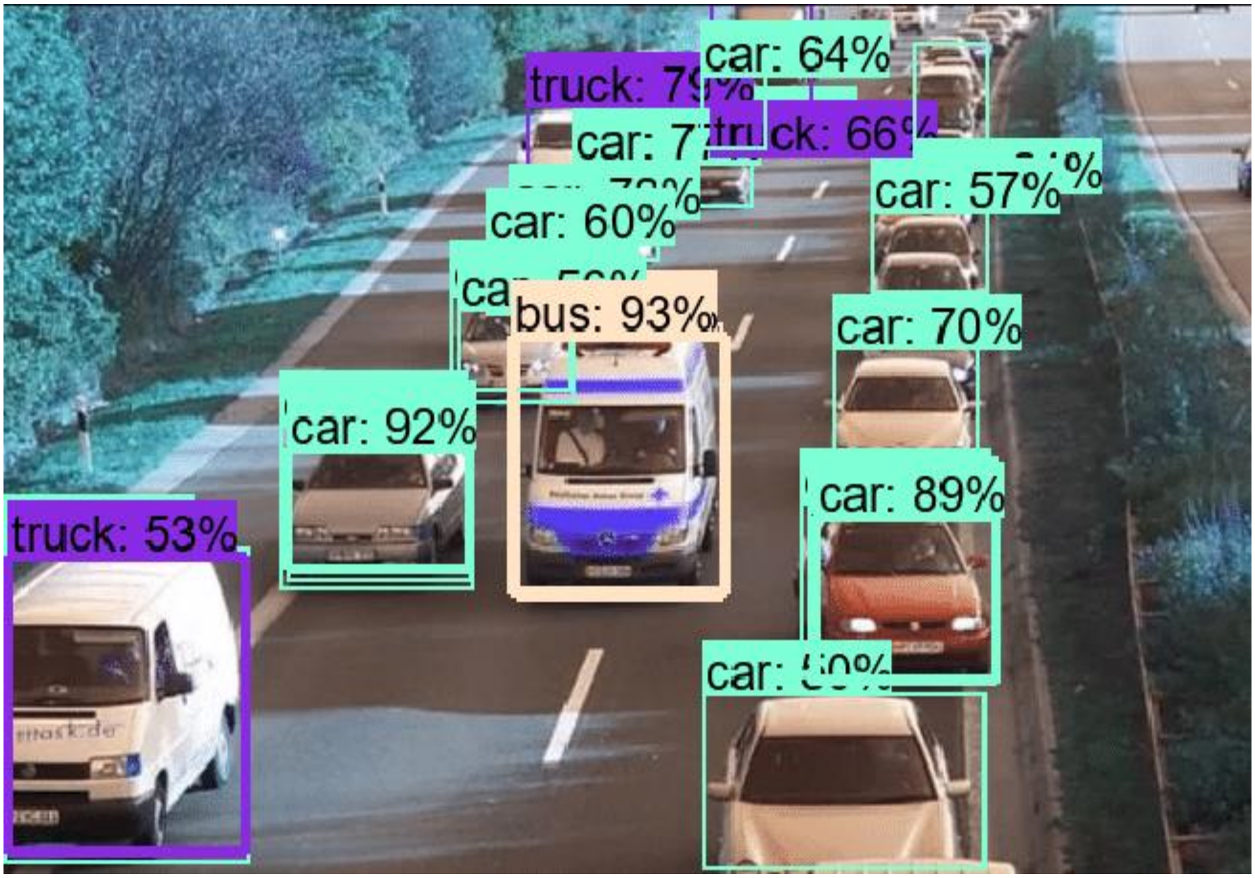


Fig.10. Sample 2 output of YoloV3 pretrained classifier

6. CODE

```
import cv2
import numpy as np
import requests
import random as rn

# Load the YOLO model
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]

# Open the video capture
cap = cv2.VideoCapture(0)  # Change the argument to a video file path if needed

car_count = 0
```

```
prev_car_count = 0

# Raspberry Pi 4 information
pi4_ip = "192.168.1.104:5000"  # Replace with the actual IP address of your Raspberry Pi 4

while True:
    # Read the video frame
    ret, frame = cap.read()

    # Perform object detection
    height, width, channels = frame.shape
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layers)

    # Process the detected objects
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5 and class_id == 2:  # Class ID 2 corresponds to 'car'
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)
                boxes.append([x, y, w, h])
                confidences.append(float(confidence))
                class_ids.append(class_id)

    # Apply non-maximum suppression to remove overlapping bounding boxes
    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)


    # Draw bounding boxes and labels on the frame
    s2_car_count = len(indexes)
    #s1_car_count = rn.randint(1,5)

    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = f"{classes[class_ids[i]]}: {confidences[i]:.2f}"
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

```
        cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    # Display the car count

    if s2_car_count != prev_car_count:
        #if s2_car_count >= 2:
            print("Number of cars:", s2_car_count)
            #requests.get(f"http://{pi4_ip}/S2_on_M")
            requests.get(f"http://{pi4_ip}/S2?s2_car_count={s2_car_count}")
        #else:
            #print("Number of cars:", s2_car_count
            #requests.get(f"http://{pi4_ip}/S2_on_L")
            #requests.get(f"http://{pi4_ip}/S2_on_L?s2_car_count={s2_car_count}")

            prev_car_count = s2_car_count

    # Show the resulting frame
    cv2.imshow("Car Detection", frame)

    # Exit the loop if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the video capture and close the window
cap.release()
cv2.destroyAllWindows()
```

7. CONCLUSION

The introduction of an automated traffic signal system that makes use of YOLOv3 object detection and Internet of Things technology marks a major step forward in the management of traffic in metropolitan areas. This project incorporates cutting-edge technology in a seamless manner in order to increase the effectiveness and responsiveness of traffic signals, which will eventually lead to improvements in urban transportation. The system achieves amazing accuracy in recognizing and measuring automobiles at signalized crossings because it uses YOLOv3, a cutting-edge object identification model. This allows for the system to fulfill its goals. The use of this technique, which "looks once" at the complete picture in order to create predictions, is critical for doing real-time traffic analysis. The resilience and flexibility of YOLOv3 are shown by the system's capability of accurately detecting vehicles regardless of the size or orientation of the targets. In addition to this, the inclusion of IoT makes it possible for the vehicle-detecting nodes and the central server, which is based on Raspberry Pi, to communicate in an easy and smooth manner. Because of this communication backbone, the traffic signal system is able to maintain its awareness of the current real-time traffic circumstances. It has become clear that the MQTT protocol, which was selected due to its ability to provide lightweight and efficient messaging, is a critical factor in allowing speedy and reliable data transfer. The Raspberry Pi acts as the system's brain and is responsible for coordinating the operation of traffic signals depending on the information that is received. The system is able to maximize the flow of traffic and reduce the amount of congestion by dynamically altering the timing of the signals in response to variations in the density of the traffic. This flexibility is a key trait, particularly in dynamic metropolitan contexts where traffic conditions may swiftly shift from one moment to the next. The hardware implementation of the project, which has been thoroughly described, demonstrates the smooth integration of many components. These components range from automotive detection nodes that are equipped with onboard cameras to sophisticated Raspberry Pi servers. The efficiency of the system as a whole is dependent on the contributions made by each component, which all play an essential part. In addition, aspects like as power management, enclosures, and secure connections illustrate an all-encompassing strategy for guaranteeing the system's dependability and longevity. The fact that this traffic management system was able to pass both its installation and testing phases successfully demonstrates that it has the potential to be used in a variety of settings. In addition to its immediate use case, this project sets the framework for additional developments in urban traffic systems. It provides a solution that is scalable and adaptive for cities that are dealing with expanding transportation difficulties. In conclusion, the incorporation of YOLOv3 object identification and IoT technology into this automated traffic light system demonstrates the enormous potential of cutting-edge technologies to transform the way urban traffic management is carried out. The accomplishments of the project, which include successfully identifying cars and dynamically managing traffic signals, are evidence that this integrated strategy is effective. These kinds of solutions represent a potential step toward a smarter and more effective urban transportation system at a time when cities are still struggling to keep up with their rapidly expanding populations.

REFERENCES

[1] Djahel, Soufiene, Nafaa Jabeur, Robert Barrett, and John Murphy. "Toward V2I communication technology-based solution for reducing road traffic congestion in smart cities." In *2015 International Symposium on Networks, computers and communications (ISNCC)*, pp. 1-6. IEEE, 2015.

[2] Jain, Vipin, Ashlesh Sharma, and Lakshminarayanan Subramanian. "Road traffic congestion in the developing world." In *Proceedings of the 2nd ACM Symposium on Computing for Development*, pp. 1-10. 2012.

[3] Bacon, Jean, Andrei Iu Bejan, Alastair R. Beresford, David Evans, Richard J. Gibbens, and Ken Moody. *Using real-time road traffic data to evaluate congestion*. Springer Berlin Heidelberg, 2011.

[4] Hurdle, V., E. Hauser, and Ph Fargier. "Effects of the choice of departure time on road traffic congestion. Theoretical approach." *Transportation and traffic theory* 8 (1983): 223-263.

[5] Ata, Ayesha, Muhammad Adnan Khan, Sagheer Abbas, Gulzar Ahmad, and Areej Fatima. "Modelling smart road traffic congestion control system using machine learning techniques." *Neural Network World* 29, no. 2 (2019): 99-110.

[6] Arnott, Richard, and Kenneth Small. "The economics of traffic congestion." *American scientist* 82, no. 5 (1994): 446-455.

[7] Yasin, Yasin J., Michal Grivna, and Fikri M. Abu-Zidan. "Global impact of COVID-19 pandemic on road traffic collisions." *World journal of emergency surgery* 16, no. 1 (2021): 1-14.

[8] Jayapal, Cynthia, and S. Sujith Roy. "Road traffic congestion management using VANET." In *2016 International conference on advances in human machine interaction (HMI)*, pp. 1-7. IEEE, 2016.

[9] Verhoef, Erik T. "Time, speeds, flows and densities in static models of road traffic congestion and congestion pricing." *Regional Science and Urban Economics* 29, no. 3 (1999): 341-369.

[10] Jayasooriya, S. A. C. S., and Y. M. M. S. Bandara. "Measuring the Economic costs of traffic congestion." In *2017 Moratuwa Engineering Research Conference (MERCon)*, pp. 141-146. IEEE, 2017.

[11] Agbonika, Frederick. "Road Traffic Congestion and the quest for effective Transportation." In *NATIONAL ENGINEERING CONFERENCE AND ANNUAL GENERAL MEETING" CANAAN*. 2011.

[12] Saroufim, Alain, and Elie Otayek. "Analysis and interpret road traffic congestion costs in Lebanon." In *MATEC Web of Conferences*, vol. 295, p. 02007. EDP Sciences, 2019.

[13] Kiunsi, Robert B. "A review of traffic congestion in Dar es Salaam city from the physical planning perspective." *Journal of Sustainable Development* 6, no. 2 (2013): 94.

[14] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[15] Mao, Qi-Chao, Hong-Mei Sun, Yan-Bo Liu, and Rui-Sheng Jia. "Mini-YOLOv3: real-time object detector for embedded applications." *Ieee Access* 7 (2019): 133529-133538.

[16] Zhao, Liquan, and Shuaiyang Li. "Object detection algorithm based on improved YOLOv3." *Electronics* 9, no. 3 (2020): 537.

[17] Li, Chong, Rong Wang, Jinze Li, and Linyu Fei. "Face detection based on YOLOv3." In *Recent Trends in Intelligent Computing, Communication and Devices: Proceedings of ICCD 2018*, pp. 277-284. Springer Singapore, 2020.

[18] Taheri Tajar, Alireza, Abbas Ramazani, and Muharram Mansoorizadeh. "A lightweight Tiny-YOLOv3 vehicle detection approach." *Journal of Real-Time Image Processing* 18, no. 6 (2021): 2389-2401.

[19] Wang, Xiaolan, Shuo Wang, Jiaqi Cao, and Yansong Wang. "Data-driven based tiny-YOLOv3 method for front vehicle detection inducing SPP-net." *IEEE Access* 8 (2020): 110227-110236.

[20] Kim, Kwang-Ju, Pyong-Kun Kim, Yun-Su Chung, and Doo-Hyun Choi. "Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection." In *2018*

15th IEEE international conference on advanced video and signal based surveillance (AVSS), pp. 1-6. IEEE, 2018.

[21] Miao, Yan, Fu Liu, Tao Hou, Lu Liu, and Yun Liu. "A nighttime vehicle detection method based on YOLO v3." In *2020 Chinese Automation Congress (CAC)*, pp. 6617-6621. IEEE, 2020.

[22] Zhao, Shuai, and Fucheng You. "Vehicle detection based on improved yolov3 algorithm." In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pp. 76-79. IEEE, 2020.

[23] Zhang, Xun Xun, and Xu Zhu. "Moving vehicle detection in aerial infrared image sequences via fast image registration and improved YOLOv3 network." *International Journal of Remote Sensing* 41, no. 11 (2020): 4312-4335.

[24] Madakam, Somayya, Vihar Lake, Vihar Lake, and Vihar Lake. "Internet of Things (IoT): A literature review." *Journal of Computer and Communications* 3, no. 05 (2015): 164.

[25] Gokhale, Pradyumna, Omkar Bhat, and Sagar Bhat. "Introduction to IOT." *International Advanced Research Journal in Science, Engineering and Technology* 5, no. 1 (2018): 41-44.

[26] Van Kranenburg, Rob, and Alex Bassi. "IoT challenges." *Communications in Mobile Computing* 1, no. 1 (2012): 9.