CROWD SIZE ESTIMATION SYSTEM USING OPENCV & RASPBERRYPI

A project report submitted in partial fulfillment of the requirements

for the award of credits to

Bachelor of Technology

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

MADANU THRINADH
(20BQ1A0498)

MANNAVA SUDHEER
(20BQ1A04A3)

JAKKAMASETTI KIRAN NAGA SAI
(20BQ1A0471)

KOPURI PREETHAM
(21BQ5A0409)

IOT TOOLS AND APPLICATIONS

(Skill Advanced Course)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE and permanently affiliated to JNTUK, Accredited by NBA and NAAC)

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508

OCTOBER 2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY: NAMBUR

(AUTONOMOUS)

CERTIFICATE

This is to certify that the Mini Project titled "**CROWD SIZE ESTIMATION USING OPENCV AND RASPBERRYPI**" is a bonafide record of work done by **M.THRINADH (20BQ1A0498),M.SUDHEER (20BQ1A04A3),J.KIRAN NAGA SAI (20BQ1A0471), K.PREETHAM (21BQ5A0409)** under the guidance of **Dr.M.VENKATESH , Professor** as part of the **IOT TOOLS AND APPLICATIONS** in partial fulfillment of the requirement of the degree for Bachelor of Technology in Electronics and Communication Engineering during the academic year 2023–24.

MS.T.SHALINI | **DR.M.Y.BHANU MURTHY**
Course Instructor | **Head of Department**

DECLARATION

We, **M.THRINADH (20BQ1A0498),M.SUDHEER (20BQ1A04A3),J.KIRAN NAGA SAI (20BQ1A0471), K.PREETHAM (21BQ5A0409)** hereby declare that the project report entitled "**CROWD SIZE ESTIMATION USING OPENCV AND RASPBERRYPI**" is done by us under the guidance of **Dr.M.VENKATESH , Professor** as part of the Advanced Skill Advanced Course **IOT TOOLS AND APPLICATIONS(Skill Advanced Course)** in partial fulfillment of the requirement of the degree for Bachelor of Technology in Electronics and Communication Engineering during the academic year 2023–24.

DATE:

PLACE: VVIT, Nambur

SIGNATURE OF THE CANDIDATES

M.THRINADH (20BQ1A0498)

M.SUDHEER (20BQ1A04A3)

J.KIRAN NAGA SAI (20BQ1A0471)

K.PREETHAM (21BQ5A0409)

ACKNOWLEDGEMENT

We express our sincere thanks wherever it is due

We express our sincere thanks to the Chairman, Vasireddy Venkatadri Institute of Technology, Sri Vasireddy VidyaSagar for providing us well equipped infrastructure and environment.

We thank Dr. Y. Mallikarjuna Reddy, Principal, Vasireddy Venkatadri Institute of Technology, Nambur, for providing us the resources for carrying out the project.

Our sincere thanks to Dr. K. Giri babu, Dean of Studies for providing support and stimulating environment for developing the project.

Our heartfelt thanks to Dr. M. Y. Bhanu Murthy, Head of the Department, Department of ECE, for his co-operation and guidance which help us to make our project successful and complete in all aspects.

We also express our sincere thanks and are grateful to our guide Dr. M.VENKATESH , Professor, Department of ECE, for motivating us to make our project successful and fully complete. We are grateful for his precious guidance and suggestions.

Also, it our duty to extend heartfelt thanks to our course instructor Mr.T.Shalini, Assistant Professor for her valuable instructions to conclude this work.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

NAME OF THE CANDIDATES :

M.THRINADH (20BQ1A0498)

M.SUDHEER (20BQ1A04A3)

J.KIRAN NAGA SAI (20BQ1A0471)

K.PREETHAM (21BQ5A0409)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Crowd size estimation plays a crucial role in various domains such as event management, crowd control, and urban planning. In this paper, we propose a crowd size estimation system that utilizes the powerful combination of OpenCV and Raspberry Pi. OpenCV, a popular computer vision library, provides a wide range of tools and algorithms for image processing and analysis, making it well-suited for crowd analysis tasks. Raspberry Pi, a low-cost single-board computer, serves as the hardware platform for our system, offering portability, affordability, and ease of integration.

The proposed system employs a camera module connected to the Raspberry Pi to capture live video footage of the crowd. The video frames are then processed using OpenCV algorithms to detect and track individuals within the crowd. By leveraging techniques such as background subtraction, foreground segmentation, and object tracking, the system can effectively identify and count individuals in real-time.Experimental results demonstrate the effectiveness and reliability of the proposed system in accurately estimating crowd sizes across various scenarios.Overall, the combination of OpenCV and Raspberry Pi offers a versatile and practical solution for crowd size estimation, with potential for further improvements and applications in the field of computer vision and intelligent systems.

CHAPTER-1

INTRODUCTION

In today's world, effectively managing crowds in various settings has become crucial, from public events and transportation hubs to emergency response situations. One of the fundamental challenges in crowd management is accurately estimating the number of people within a crowd. This project addresses a problem with wide-ranging applications, including event planning, security, urban infrastructure design, and resource allocation. To tackle this challenge, we turn to computer vision, a field that has seen remarkable advancements in recent years.

OpenCV, an open-source computer vision library, is a powerful tool that provides the means to process and analyze images and videos effectively. We leverage the portability and computational capabilities of the Raspberry Pi, a low-cost, single-board computer, to make our solution versatile and affordable. The primary goal of this project is to develop a system that can count and estimate the size of crowds accurately and in real-time. Our approach involves capturing image frames or video feeds, processing them using OpenCV's algorithms for object detection, and analyzing the data to count individuals.

The project also aims to provide real-time data visualization and reporting to assist decision-makers in managing crowds effectively. By achieving these objectives, we contribute to the development of cost-effective, adaptable solutions for crowd size estimation, which has a significant impact on public safety and event planning. As events and public gatherings continue to grow in scale and complexity, the need for precise crowd size estimation becomes increasingly evident. Traditional crowd counting methods often rely on manual counts or approximations, which are labor-intensive and prone to errors.

Leveraging the power of technology to automate crowd size estimation not only reduces human effort but also enhances accuracy. This project bridges the fields of computer science, image processing, and hardware engineering, making it a multidisciplinary endeavor. The utilization of Raspberry Pi ensures that our solution is accessible and cost-effective, enabling its deployment in a wide range of scenarios. By the end of this project, we hope to contribute to the ongoing advancements in computer vision and its practical applications in crowd management.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

Raspberry Pi 3B+ Model:

A Raspberry Pi is a small, affordable, and versatile single-board computer (SBC) that has gained immense popularity for its wide range of applications. The Raspberry Pi is a credit-card-sized computer developed by the Raspberry Pi Foundation, a non-profit organization based in the United Kingdom. It was originally created to promote computer science education and provide an affordable platform for learning programming and electronics, but it has since evolved into a powerful and flexible tool used in various fields. The Raspberry Pi 3 Model B+ (often abbreviated as Raspberry Pi 3B+) is a specific iteration of the Raspberry Pi single-board computer series.

Here's a detailed description of the Raspberry Pi 3B+ model :

The Raspberry Pi 3 Model B+ is an improved version of its predecessor ,the Raspberry Pi 3 Model B. It was released in March 2018 and offers several enhancements in terms of performance, connectivity, and thermal management.The Raspberry Pi 3B+ was a popular model in the Raspberry Pi lineup due to its improved processing power, wireless connectivity options, and versatility for various projects. However, newer models with even more features and capabilities have been released.The Raspberry Pi 3 Model B+ is a versatile and capable single-board computer with improved processing power and connectivity options. Its affordability and strong community support make it a popular choice for hobbyists, educators, and professionals working on diverse projects and applications.

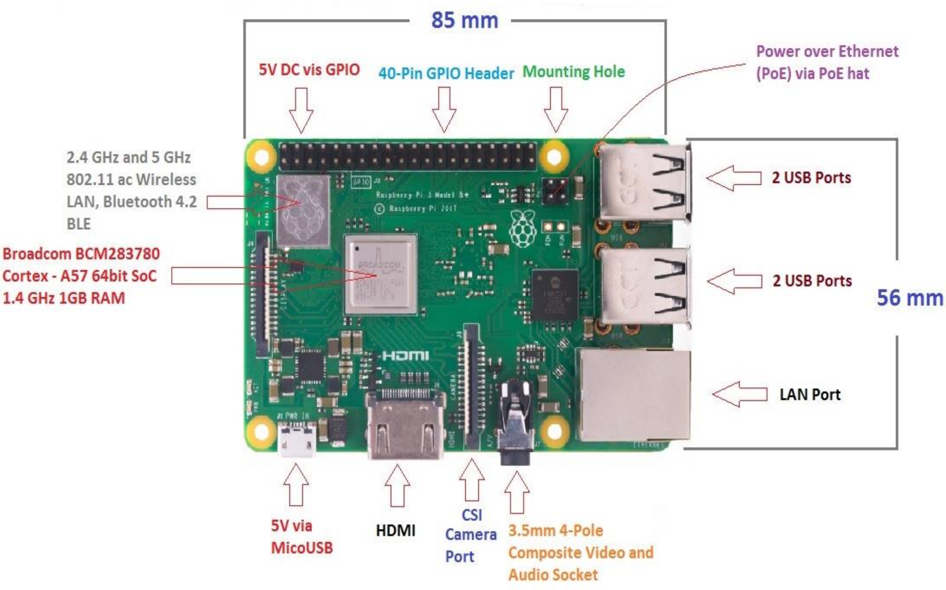

Fig 2.1 Raspberry Pi 3B+

Applications:

Raspberry Pi is a versatile single-board computer with a wide range of applications. In the realm of education, it serves as an excellent tool for teaching computer science and electronics, enabling students to learn programming and hardware concepts in a hands-on way. Moreover, it finds extensive use in home automation, transforming ordinary homes into smart ones. With Raspberry Pi, DIY enthusiasts can create custom smart home systems, controlling lights, cameras, and more. Beyond education and home automation, Raspberry Pi's applications extend to media centers, web servers, network attached storage , gaming consoles, robotics, IoT projects, digital art, and scientific research. Its affordability, coupled with a supportive community, has made it a go-to choice for diverse projects, catering to a broad spectrum of interests and needs.

SD Card Reader:

Fig 2.2 SD card reader

An SD card reader, also known as a Secure Digital card reader, is a hardware device used to read data from and write data to SD (Secure Digital) memory cards. These readers are commonly used to transfer files, photos, videos, and other data between SD cards and computers or other devices that support SD card storage. An SD card reader is a hardware device or component that allows you to read and write data to and from Secure Digital (SD) memory cards. SD cards are commonly used for storing data in portable devices such as digital cameras, smartphones, tablets, and various other electronic gadgets.

SD card readers are used to transfer data to and from an SD card and a computer or other compatible device. This data can include photos, videos, documents, music, and more. They are also used for tasks like formatting the card or updating firmware. It's essential to ensure that your SD card reader is compatible with your operating system, whether it's Windows, macOS, Linux, or another system. Most modern SD card readers are plug-and-play and work with a wide range of operating systems.

SD card readers typically connect to a computer or device via USB interface. They often feature a USB Type-A or USB Type-C connector, making them compatible with various USB ports on laptops, desktops, and other devices Secure Digital (SD) cards have become widely popular for various storage and data transfer.

It has 2 key advantages:

1. Versatility
2. Portability

Camera Module:

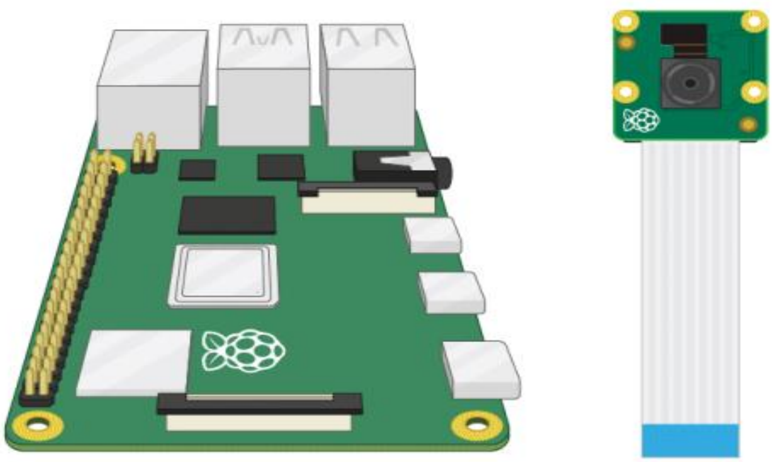

Fig 2.3 camera Module

The camera module, also known as CCM (compact Camera Module), has been widely used in video conferencing, Security systems and real-time monitoring as a video input device. It is a compact electronic device designed to capture still images and videos. These modules are commonly used in various applications, including smartphones, tablets, digital cameras, and embedded system

Camera modules are primarily designed for capturing high-quality images and videos. They can vary in resolution, with some modules capable of capturing high-definition (HD), Full HD, or even 4K video, depending on the model.

HDR technology is used in some camera modules to capture photos with a wider range of brightness levels, resulting in more balanced and detailed images. Camera modules may include user interfaces such as buttons, dials, touchscreens, or graphical menus that allow users to adjust settings, switch between shooting modes, and review captured images and videos. The heart of a camera module is its image sensor, which is typically a CMOS (Complementary Metal-Oxide-Semiconductor) or CCD (Charge-Coupled Device) sensor. This sensor is responsible for capturing light and converting it into an electrical signal.

USB cable:

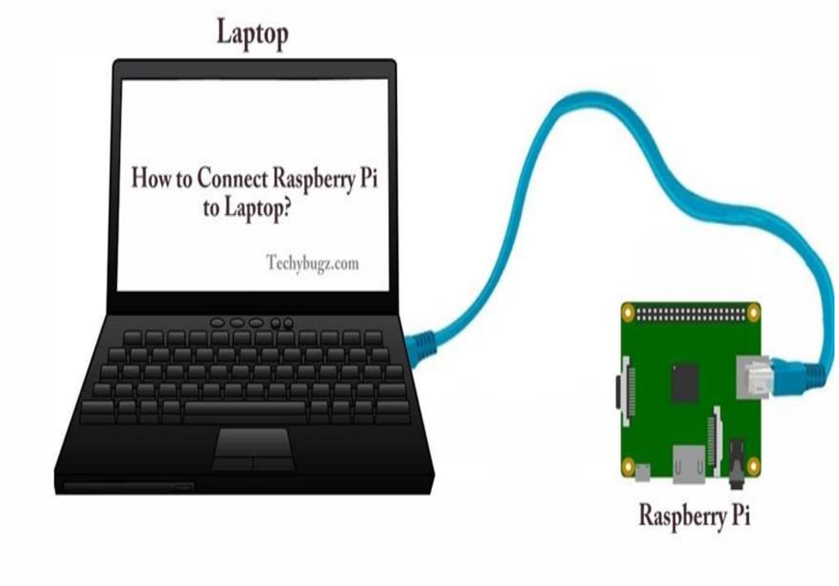

Fig 2.6 USB wire

The term USB stands for "Universal Serial Bus". USB cable assemblies are some of the most popular cable types available, used mostly to connect computers to peripheral devices such as cameras, camcorders, printers, scanners, and more. The USB cable standard allows for these advantages over serial cable types:

- USB cables are "Hot Pluggable", in other words you can connect and disconnect the cables while the computer is running without fear of freezing the computer

- USB cables are fast, transferring up to 480Mbps. Compare that to serial communication which transfers data at about 20Kbps
- USB cables carry power as well as signals. This allows for "USB powered" gadgets as well as recharging batteries in cameras and other USB peripherals
- USB cables are designed with several distinct connector types, making it easy to identify which plug goes into the computer and which plug goes into the peripheral device
- USB cables are a universal standard and are fairly easy to find and to afford

SOFTWARE REQUIREMENTS:

Download Raspberry Pi OS:

1.Go to the official Raspberry Pi website: https://www.raspberrypi.org/

2.Navigate to the "Downloads" section.

3.Download the latest version of Raspberry Pi OS (formerly Raspbian). You can choose between the full desktop version (with a graphical user interface) and the Lite version (without a GUI). The full desktop version is recommended for beginners.

Prepare the SD Card:

1.Insert your SD card into your computer's SD card reader.

2.Use a tool like "Etcher" (https://etcher.io/) to write the downloaded Raspberry Pi OS image to the SD card. This tool makes the process easy and ensures that the image is correctly written to the SD card. Follow the instructions in Etcher to select the image and the target drive (your SD card)

For Raspberry pi Display:

To use your laptop's screen as a display for your Raspberry Pi, you can set up what is known as "Remote Desktop" or "VNC" (Virtual Network Computing) to establish a graphical connection between your Raspberry Pi and your laptop. This allows you to control the Raspberry Pi from your laptop's screen.

Here are the steps to do this:

1. **Enable VNC on Raspberry Pi:**

Boot your Raspberry Pi with the prepared SD card. If you've enabled SSH during the installation process (or later), you can access your Raspberry Pi using SSH from your laptop.

Open a terminal on your laptop and run: ssh pi@raspberrypi.local

* Replace raspberrypi.local with your Raspberry Pi's hostname or IP address if you know it.

Once connected, run the following command to enable VNC:

sudo raspi-config

***** Navigate to "Interfacing Options" and select "VNC." Confirm that you want to enable VNC.

2.Install a VNC Client on Your Laptop:

On your laptop, you need a VNC client software to connect to your Raspberry Pi. There are various VNC client applications available for different operating systems. Here are a few examples:

For Windows: RealVNC, TightVNC, UltraVNC.

For Linux: Remmina, Vinagre, TigerVNC Viewer.

Libraries Required:

OpenCV:

OpenCV stands for Open Source Computer Vision Library. It is an open-source computer vision and machine learning software library designed for various computer vision tasks. It provides a wide range of functions and tools for tasks such as image and video processing. It is cross-platform and compatible with Windows, macOS, Linux, and even mobile platforms like Android and iOS. This makes it versatile for a variety of applications. OpenCV has a Python library, making it accessible and easy to use for Python developers.

OpenCV allows you to read and write images and videos in various formats. It can capture frames from cameras and process video streams.You can perform a wide range of image manipulation tasks, including resizing, rotating, cropping, filtering, and transforming images. OpenCV supports object detection and recognition through methods like Haar cascades, HOG (Histogram of Oriented Gradients), and deep learning-based models like YOLO and SSD.

imutils:

imutils is a Python library that provides a collection of convenience functions and classes to simplify common tasks in image processing and computer vision. It is often used in conjunction with the OpenCV library, making it easier to work with images and video streams in OpenCV by providing functions that streamline various operations. One of the primary features of imutils is its resize function, which simplifies the process of resizing images while maintaining the aspect ratio. It offers functions for rotating images, which is useful for tasks like image alignment or correcting image orientation.

imutils provides a imshow function that simplifies the display of images and allows you to efficiently display images in OpenCV windows. It includes tools for working with video streams, such as the ability to read frames from a video file or camera feed and efficiently display them. imutils offers functions for contour manipulation and simplification, which can be handy when working with object detection and image segmentation. It provides various utility functions for tasks like finding the closest color match, calculating the distance between two points.

matplotlib:

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations in 2D and 3D. It is widely used for creating publication-quality plots, charts, and graphs for scientific, engineering, and data analysis purposes. Matplotlib offers a wide range of plotting options, including line plots, scatter plots, bar plots, histograms, pie charts, and more. It is compatible with various GUI toolkits, making it usable across different platforms and environments, including Jupyter notebooks, web applications, and desktop applications.

Matplotlib provides a simple and intuitive interface for creating basic plots, while also allowing fine-grained customization for advanced users.Matplotlib supports multiple backends, including Agg, Qt, Tkinter, and more, allowing you to choose the most appropriate backend for your application.It is extensible through various toolkits and libraries built on top of Matplotlib, such as Seaborn for statistical data visualization and Basemap for geographic data. Matplotlib is widely used in the scientific and data analysis communities, and many other libraries and tools. Matplotlib is a versatile and powerful library that is an essential tool for data visualization and exploration in Python.

numpy:

NumPy is a Python library that provides support for large, multi-dimensional arrays and matrices of data, along with an assortment of high-level mathematical functions to operate on these arrays efficiently. NumPy offers various ways to create arrays, including manually creating them, generating sequences, and importing data from files. The np.array() function is commonly used for creating arrays.

NumPy provides powerful indexing and slicing capabilities, allowing you to access and manipulate specific elements or subsets of arrays efficiently. NumPy offers a comprehensive suite of linear algebra functions, such as matrix multiplication, eigenvalue decomposition, and singular value decomposition. NumPy includes functions for generating random numbers and random samples from various probability distributions, which are useful for tasks like simulations and statistical analysis. NumPy is implemented in C and optimized for performance, making it much faster for numerical computations compared to native Python lists.

requests:

The requests library is a popular Python library for making HTTP requests to web services, APIs, or websites. It simplifies the process of sending HTTP requests and handling responses. requests provides a straightforward and user-friendly API for sending HTTP GET, POST, PUT, DELETE, and other types of requests. You can specify URL endpoints, headers, parameters, and request data easily. It allows you to handle HTTP responses efficiently. You can access response content, status codes, headers, and cookies, making it easy to parse and process the data returned from web services.

CHAPTER 3
WORKING AND IMPLEMENTATION

To interface a Pi Camera with a Raspberry Pi:

1. Hardware Requirements:
   - Raspberry Pi (with an SD card containing operating system)
   - Raspberry Pi Camera Module
   - Power supply for the Raspberry Pi

2. Enable the Camera Interface:
   - Open the Raspberry Pi Configuration Tool by running “sudo raspi-config” in the terminal.
   - Go to “Interfacing Options”.
   - Select “Camera” and enable it. Reboot if prompted.

3. Connect the Camera:
   - Ensure your Raspberry Pi is powered off.
   - Connect the ribbon cable from the Camera Module to the Camera Serial Interface (CSI) port on the Raspberry Pi. Make sure it's properly seated.

4. Install Camera Software:
   - Ensure your Raspberry Pi is connected to the internet.
   - Open the terminal and update the package list: “sudo apt-get update”.
   - Install the camera software: “sudo apt-get install python3-picamera”

5. Take Photos or Record Video:
   - We can now use Python to interact with the camera.

Working:

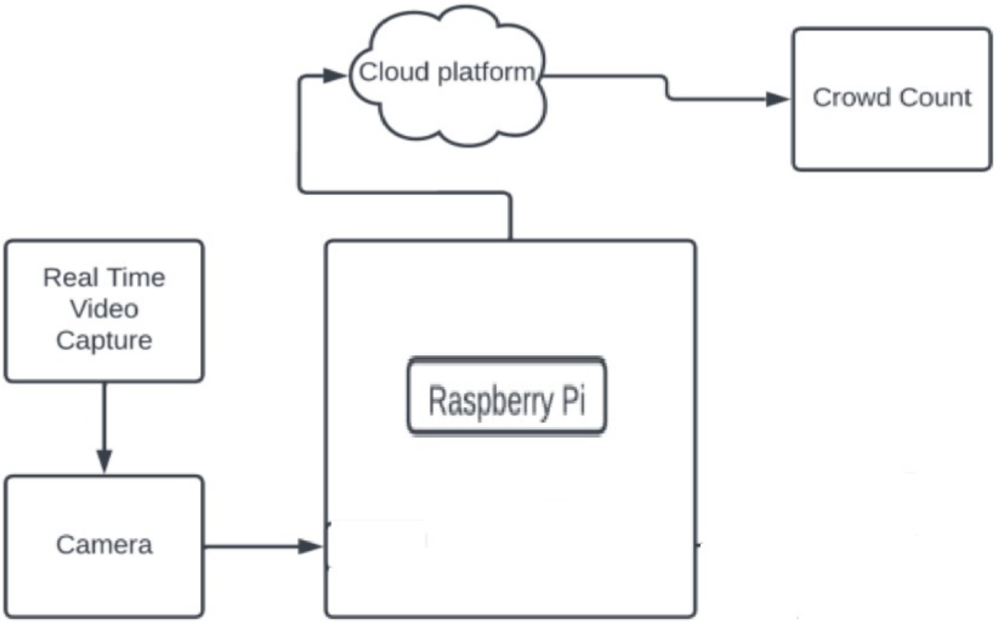

Fig 3.1 Block Diagram

From the above Block Diagram the Raspberry Pi's camera module captures a continuous video stream of the crowd or area We want to analyze. This stream consists of a series of individual frames, typically at a rate of 30 frames per second or lower. Each frame from the video stream undergoes preprocessing. This may include operations like resizing the frame to a manageable resolution, converting it to grayscale to simplify processing, and applying noise reduction techniques to improve the accuracy of object detection.

OpenCV, along with machine learning models or cascades, is used to detect people within each frame. Object detection algorithms analyze the frame to identify regions or bounding boxes where people are present. This step involves pattern recognition and classification of objects as "people”. Once people are detected within a frame, a counting mechanism keeps track of the number of detected individuals.

This could be as simple as incrementing a counter for each detected person or employing more complex algorithms to handle cases where people might enter or exit the frame over time. The count of detected people in each frame is accumulated over time to estimate the total crowd size. This estimation has displayed in a separate window for monitoring. We sent the log of the crowd size data over time for analysis and visualization purposes to Thingspeak channel.

The project require optimization by adjusting detection parameters, camera placement, or other factors to improve accuracy. Extensive testing in various crowd scenarios helps ensure the system's reliability. It can be deployed in real-world scenarios where crowd size estimation is needed, such as public events, transportation hubs, or building occupancy monitoring.

Implementation:

Implementing a crowd size estimation system with email alerts involves integrating various components, including hardware and software. Here's a step-by-step to set up a basic crowd size estimation system:

1.Set Up Raspberry Pi:

- Install the operating system on your Raspberry Pi.
- Connect the camera module.

2.Write Python Code:

- Create a Python script.
- Record videos.
- Configure the script to Thingspeak channel to post data graphically.

3. Data Posting Logic:

- Implement logic in your Python code to post data on Thingspeak channel.

4.Testing:

- Test your system by deploying at various crowd areas.
- Ensure that data has been posting on Thingspeak channel.

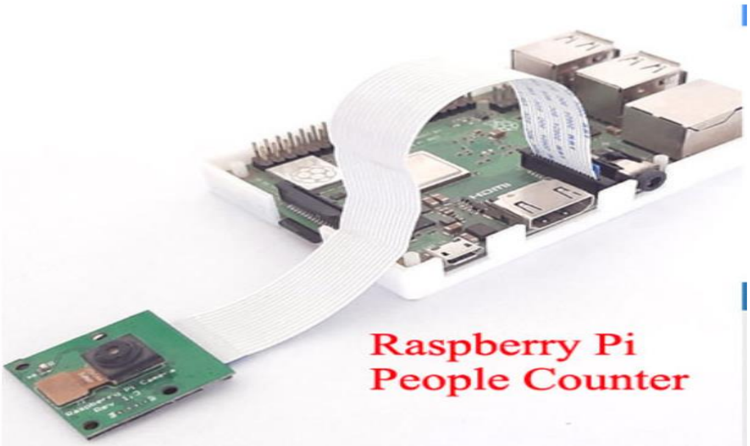

Fig 3.2 Hardware Setup

CHAPTER 4

RESULTS AND DISCUSSION

The code utilizes the Histogram of Oriented Gradients (HOG) method and a pre-trained people detection model to identify individuals in a real-time video stream.As the code runs, it continuously captures frames from the camera and analyzes them for the presence of people. Detected individuals are highlighted with green bounding boxes in the video stream. Each bounding box represents a person detected in the frame. The number of people detected in each frame is counted and stored in the variable result1.This count represents the instantaneous crowd size within the field of view of the camera at that moment.

The code is integrated with ThingSpeak, an IoT platform for data collection and visualization. At regular intervals specified by sample_time, the code sends the current crowd count (result1) to a ThingSpeak channel via HTTP request. The ThingSpeak channel is configured with the field name "field1" to receive and store the crowd size data.Over time, the ThingSpeak channel accumulates a historical record of crowd size measurements.

The console provides feedback and status updates during code execution.It displays the number of detected people in each frame.It also shows the ThingSpeak URL being constructed and the status of sending the data to ThingSpeak. In summary, this project allows real-time crowd detection and counting while providing the ability to log and visualize the data using ThingSpeak.

OUTPUT:

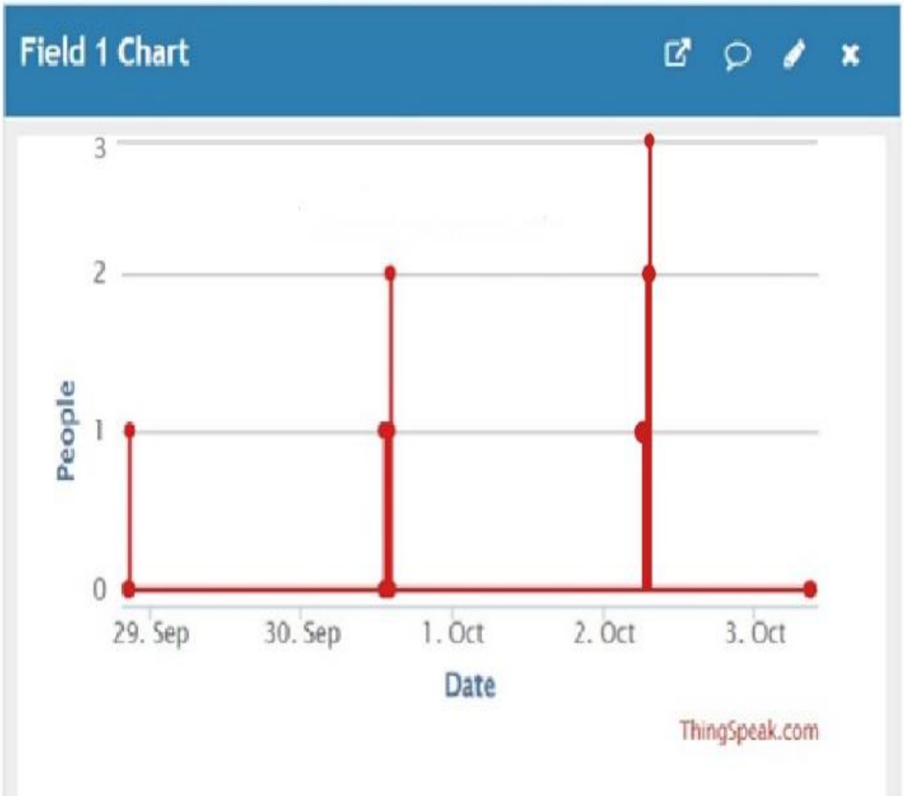

Fig 4.1 This graph shows how many people were counted in an area at different times. We used a camera with a Raspberry Pi to count them, and the graph helps us understand when there were more or fewer people.

CHAPTER-5

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

The core of the project revolves around the HOG (Histogram of Oriented Gradients) object detection algorithm. This algorithm is capable of recognizing the human shape by analyzing patterns of gradient orientations in images. HOG is configured with a pre-trained people detection model, making it capable of detecting human figures in real-time. The code captures video frames in real-time from a connected camera, usually a webcam. These frames are processed one by one to identify individuals. As each frame is processed, the algorithm draws green bounding boxes around detected people, making it easy to visualize the crowd detection process.

Within each frame, the algorithm counts the number of detected individuals and records this count as result1. This count represents the instantaneous crowd size within the camera's field of view at any given moment. The project integrates with ThingSpeak, an Internet of Things (IoT) platform. It uses the ThingSpeak API to send the crowd count data to a specific ThingSpeak channel. The channel is pre-configured to receive this data and store it in a field called “field1”.

The console provides real-time feedback during code execution. It displays the number of detected people in each frame, offering insights into the ongoing crowd monitoring process. Additionally, it shows the construction of the ThingSpeak URL and indicates whether the data has been successfully sent to ThingSpeak.

The project is versatile and can be applied in various real-world scenarios, including:

- Event Management: Monitoring and managing crowd size at concerts, sports events, or festivals.
- Public Safety: Ensuring social distancing during a pandemic by tracking crowd density in public places.

- Retail Analytics: Optimizing staffing and layout in retail stores based on customer traffic patterns.

This code serves as a solid foundation for more advanced crowd monitoring systems. It's a starting point that can be extended and enhanced to meet specific requirements. It provides a valuable tool for capturing real-time crowd data, which can be further analyzed and visualized for decision-making.

FUTURE SCOPE:

Data Visualization and Analysis:

Enhance the project by incorporating advanced data visualization tools and techniques. Create interactive graphs, charts, and heatmaps to represent crowd density and behavior over time. Implement statistical analysis to extract meaningful insights from historical crowd data. This can include trend analysis, anomaly detection, and seasonality patterns.

Machine Learning Integration:

Integrate machine learning models, especially deep learning techniques such as Convolutional Neural Networks (CNNs), to improve crowd detection accuracy and robustness. Train models to recognize specific attributes or behaviors within the crowd. Implement transfer learning to adapt pre-trained models for crowd monitoring tasks in various environments and conditions.

Real-Time Alerts and Automation:

Develop a system for real-time alerts and automated responses based on predefined crowd size thresholds or behavior patterns. Alerts can be sent to event organizers, security personnel, or relevant authorities.Implement automation for crowd management tasks, such as adjusting access control, deploying additional staff, or redirecting crowds to alleviate congestion.

Privacy Considerations:

Address privacy concerns related to video surveillance. Implement privacy-preserving techniques, such as anonymization of video data or compliance with data protection regulations like GDPR. Provide options for blurring or obfuscating individual faces in the video feed to protect privacy.

Integration with IoT Sensors:

Extend the project's capabilities by integrating with IoT sensors and other data sources. These sensors can provide additional context, such as environmental conditions (temperature, humidity) or crowd sentiment (e.g., via sentiment analysis of social media data).

Edge Computing:

Optimize the code for edge computing devices, allowing for decentralized crowd monitoring. This reduces reliance on a central server and can be beneficial for scenarios with limited network connectivity.

REFERENCES

- https://en.wikipedia.org/wiki/ThingSpeak
- https://en.wikipedia.org/wiki/Raspberry_Pi
- https://en.wikipedia.org/wiki/OpenCV
- Crowd Size Estimation Using OpenCV and Raspberry Pi (circuitdigest.com)
- https://en.wikipedia.org/wiki/NumPy
- https://en.wikipedia.org/w/index.php?search=requests+library&title=Special%3ASearch&ns0=1