Automatic Vehicle Check in Check out System using IoT and Deep Learning

Submitted by:

Saad Bin Riaz

FA19-BCE-021

Syed Sybtain

FA19-BCE-014

Husnain Murtaza

FA19-BCE-005

Program: BS in Computer Engineering

Supervised by:

Dr. Hafeez Anwar

Co-Supervised by:

Dr. Wasiq Ali

A thesis submitted in partial fulfillment of the requirements for the degree of bachelors of Science in Computer Engineering

Department of Electrical and Computer Engineering

COMSATS University Islamabad, Attock Campus, Pakistan.

We dedicate this thesis to our Holy Prophet Hazrat Muhammad (PBUH)
&
to my family

Abstract

This project proposes the idea of real time vehicular check in check out using IoT and deep learning. Most of the times, the check-in and check-out of vehicles are performed manually which is a tedious and time-consuming process. In order to overcome this problem, we aim at developing an automatic system that will perform vehicular check in and check out. To this end, the image-based vehicle type and license plate recognition will be done using deep learning. The camera control will be done via an IoT inspired system. For automatic record maintenance, a personal computer will be used as a server that will be connected with the Raspberry Pi via Bluetooth.

Keywords : IoT (Internet of Things), Deep learning, Manual check-in/check-out, Image-based recognition, Vehicle type recognition, License plate recognition, Raspberry Pi, HC-05 Bluetooth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The use of smart systems is increasing day by day in our country and in other developing countries in different fields, given their importance. Manual vehicle check-in and check-out processes are commonly used on university campuses to manage vehicle and transportation resources. These processes involve manual tracking and recording of vehicle data, such as registration numbers, check-in and exit dates and times. While manual check-in and check-out processes can be effective to some extent, they also present a number of challenges, including the risk of errors and inaccuracies, procedures and limited visibility of vehicle status in real time. This is why many Campus are exploring digital fleet management solutions to address these challenges and improve their overall parking and transportation operations. Manual processes can present a number of difficulties, Manual vehicle registration and control can be time-consuming, especially if there are a large number of vehicles to manage vehicles to manage. This can lead to delays and inefficiencies in fleet management. Manual processes are more prone to human error, which can lead to inaccuracies and errors in tracking and recording vehicle data. Manual data collection can make it difficult to analyze data, which can lead to lost productivity. Manual processes can be less secure than digital processes, which can lead to an increased risk of theft, fraud and unauthorized access to vehicle data. Manual processes can make it difficult to communicate effectively with drivers, maintenance personnel, and other visitors involved in fleet operations which can lead to delays and inefficiencies. In general, manual vehicle check-in and check-out processes can be time-consuming, error-prone, and inefficient, which can limit organizations' ability to optimize their fleet operations. These processes can limit organizations ability to optimize their operations. Digital fleet management solutions can help address these challenges by providing Real-time visibility, automated data collection, and more effective communication and collaboration. To this end, we have proposed this document to introduce our recording system to minimize time consumption, human error and advance the manual recording system. This model has been proposed in the past for different uses, but here we are aiming for an advanced level to make changes that will make it effective from an environmental point of view. changes that will make it effective based on past work done in its field. We use IoT and deep learning to create this model. The transportation sector has undergone a significant transformation in recent years, with the advent of new technologies such as such as the Internet of Things (IoT) and deep learning. These technologies have the potential to revolutionize the way vehicles are managed in the entry area. are managed in the entry and parking area, reducing manual effort and improving process efficiency. In this article, we offer an advanced system for check-in and check-out of vehicles using IoT and deep learning techniques. The traditional manual check-in and check-out process at the gates and entrance is a laborious and time-consuming task. It involves manual data entry, which can be prone to errors and inconsistencies. With the increasing number of vehicles and the need for effective registration management, there is a need for a more automated solution. The use of IoT and Deep Learning technologies have the ability to overcome these limitations and provide a more efficient, accurate, and desirable solution. The proposed system consists of IoT devices such as cameras and sensors installed at the entrance. Deep learning algorithms then process the data to perform data to perform tasks such as license plate recognition and vehicle classification. The system is designed to automate the vehicle recognition process. The system is designed to automate the vehicle check-in and check-out process, reducing manual effort and improving process efficiency. The proposed system was evaluated using a customized vehicle data set, and the results show a significant improvement in the following areas: The accuracy and speed of the check-in and check-out processes. Deep learning algorithms were able to accurately recognize license plates and classify vehicles with high accuracy. license plates and classify vehicles with high accuracy, reducing manual effort and improving process efficiency. The proposed system has the potential to revolutionize the way vehicles are managed and can be developed and improved in the future. and improved in the future.

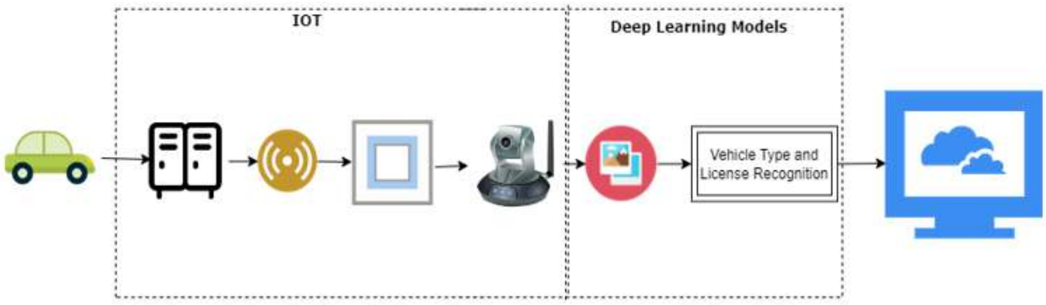

Figure 1.1: Proposed System block diagram.

The process in which we will realize the proposed model involves a mechanism that will be described using components. The mechanism for detecting the movement of a door using a sensor, an Arduino board and a Raspberry Pi, and to identify Any vehicle in the aisle using a deep learning algorithm which is an example of a system that combines multiple technologies to achieve a specific goal. to achieve a specific goal. This mechanism is designed to provide an improved process by detecting any vehicle in the area when the gate is open. vehicles present in the area when the barrier is open. The system is based on the principle of using a sensor to detect movement. Then wireless transmission of a signal to a processing unit that can activate a camera and perform image analysis to detect a vehicle in the area. detect a vehicle in the area. The use of a deep learning algorithm further enhances the system's ability to detect and identify vehicles. identify vehicles, making it an effective tool for recording, security and monitoring applications. This mechanism is an example the use of modern technology to create advanced solutions that improve verification, safety and security. In general, this mechanism uses a combination of sensors, microcontrollers, wireless communication and computer vision to detect the movement of a door and identify any vehicle in the area. Taking forward we will describe all in detail forward every mechanism briefly.

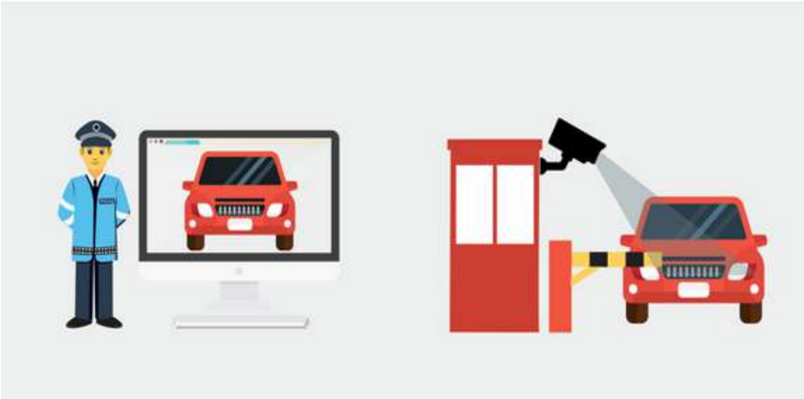

Figure 1.2: System Setup

When the gate is opened manually, the sensor detects motion and sends a signal to the Arduino board. The Arduino board is programmed to receive this signal and activate the HC-05 Bluetooth module to send the wireless signal to a Bluetooth device such as the Raspberry-pi. Wireless signal to a Bluetooth device such as the Raspberry-pi. The HC-05 module is a Bluetooth module that can be used for wireless communication between devices. It is largely It is widely used in various applications due to its low cost, ease of use, and compatibility with various microcontrollers. To use the HC-05 module with the Arduino board, the module is first paired with the Bluetooth device. Once paired, the Arduino board can send a signal to the module, which then transmits the signal wirelessly to the paired device. The Arduino board can be programmed to send specific information about the movement of the door. This information can be used to trigger specific actions on the paired device, such as sending a notification to the user's desktop, laptop, smartphone or tablet. Overall, using the HC-05 Bluetooth module is a convenient and effective way to monitor door activity, send information, and stay informed about the status of the door. Using the Bluetooth HC-05 module provides a convenient and efficient way to monitor door activity, send information, and stay informed of any movement at the door. It allows remote monitoring and control, making it an ideal solution for communication purposes. This is an ideal solution for communication purposes.

When the signal is sent from the HC-05 to a Bluetooth-enabled device, it needs connectivity from the camera to the controller. Once the system is set up, you can send a signal from the HC-05 module to the Raspberry Pi to open the camera as soon as the gate is opened. This can be a useful way to monitor door activity and capture video or images of any vehicle in sight. In addition, the Raspberry Pi can be configured to transmit the video stream to a remote device, providing a convenient way to monitor the area in real time.

Figure 1.3: Vehicle Type Detection.

After capturing the images of the vehicles, we come to image processing for the detection of vehicles and license plates. In recent years, vehicle detection and classification have become crucial elements of Intelligent Transportation Systems (ITS). The ability to detect and classify vehicles on the road accurately and quickly has many potential applications, including traffic analysis, road safety and vehicle management. Traditional methods of vehicle detection and classification rely on hand-created features and rule-based algorithms. and rules-based algorithms. However, with recent advances in deep learning, computer vision, and machine learning, new, more accurate and robust methods have emerged. However, with recent advances in deep learning, computer vision, and machine learning, new, more accurate and robust methods have emerged. This thesis focuses on the development of a vehicle type detection system using the You Only Look Once object version 4 (YOLOv4) detection model. The proposed system is designed to automatically detect and classify the type of vehicles from images and videos captured by cameras. The YOLOv4 model is a state-of-the-art deep learning algorithm that can detect objects in real time with high accuracy and efficiency. The main objective of this thesis is to develop a system that can detect and classify different types of vehicles, including cars, trucks, buses and Suzuki. The proposed system is developed using a large set of vehicle image and video data, and the YOLOv4 model. The model is able to detect and classify different types of vehicles.

Figure 1.4: License Plate Detection.

Licence plate recognition is a crucial element of intelligent transport systems, with many potential applications in traffic analysis, traffic analysis, road safety and vehicle management. Character extraction is a key step in the license plate recognition process. It involves the detection and recognition of individual characters on the license plate. and recognition of individual characters on the license plate. YOLO (You Only See It Once) is a real-time object detection system that has demonstrated excellent performance in a variety of object detection situations. Excellent performance in a variety of object detection tasks, including character extraction from license plates. YOLO v7 is the latest version the YOLO object detection system, which improves detection accuracy and speed.This thesis focuses on the development of a character extraction system for number plate recognition using the YOLO v7 object detection system. The proposed system is designed to automatically detect and recognize individual characters on plates from images and videos captured by cameras mounted on roads and highways. cameras installed on roads and highways. The YOLO v7 system uses a deep convolutional neural network (CNN) to detect objects in real time, allowing it to be used in systems that use a deep convolutional neural network. It can be used in intelligent transport systems and other applications. The proposed system is trained on a large set of license plate image data to accurately detect and recognize individual characters.

Figure 1.5: Character Recognition.

The YOLO v7 system uses a characteristic pyramid network (FPN) to detect objects at different scales, making it more accurate and robust. The system is also designed to operate in real time, making it suitable for use in intelligent transport systems and other applications. The main objective of this thesis is to develop an accurate and efficient character extraction system that can be used for number plate recognition. The proposed system has the potential to improve the accuracy and efficiency of number plate recognition.

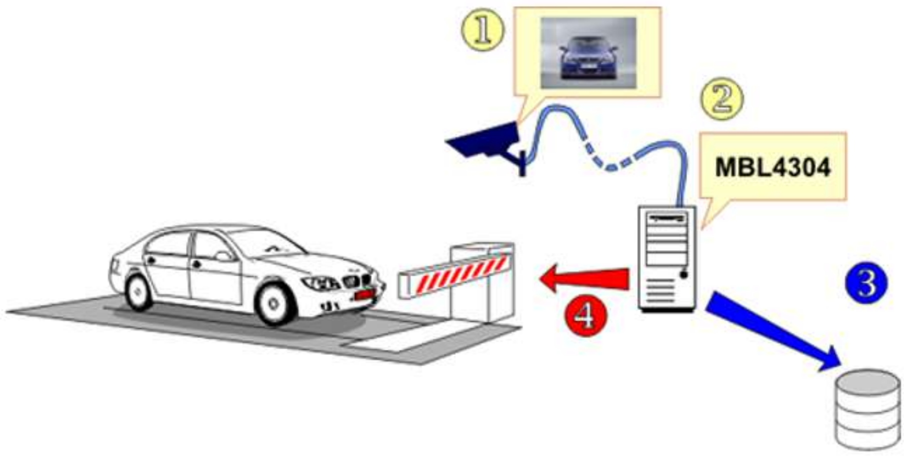

Figure 1.6: Overall System Diagram.

After the vehicle type and license plate detection we will record the information related to vehicle in an excel file which will be then passed to a client server to keep the track of all the previous information.

1.2 Motivation

The motivation behind the proposed system is to provide a more efficient and accurate solution for check-in and check-out at the entrance. The traditional manual process is time-consuming, error-prone and lacks transparency, making it difficult to

Management of a large number of vehicles in parking areas. The use of IoT and deep learning technologies has the ability to automate the process, reducing manual effort. the process, reducing manual effort and improving system efficiency. The goal of this project is to create a registration system for the university campus using IoT and Deep Learning to create a facility to verify the registration of incoming vehicles. to check the registration of incoming and outgoing vehicles, since we know the advantages of this system that motivates us to do so. for the good of humanity. On the other hand, the manual process of recording entries and exits in parking areas is a very labor- and time-intensive task. and takes a lot of time. It requires a significant amount of human effort and can lead to errors and inconsistencies. In addition, the manual process is not scalable and can lead to errors and inconsistencies. The manual process is not scalable and cannot process large amounts of data, making it inefficient for modern parking registration and management systems. The use of IoT and deep learning techniques has the potential to overcome these limitations and provide a more efficient, accurate, and scalable solution. The motivation for this research is based on the need to address the challenges associated with traditional manual registration and collection of vehicles in parking areas. The traditional process is time-consuming, error-prone and lacks transparency, this makes it difficult to manage a large number of vehicles in parking areas. Another motivation is to improve the safety of vehicles parked in parking areas. The proposed system can accurately recognize The proposed system can accurately recognize license plates and identify vehicles parked in the area, reducing the risk of theft and unauthorized access. The system can Also monitor the movement of vehicles and alert authorities in case of suspicious activity. In addition, the proposed system is aligned with the United Nations Sustainable Development Goals (SDGs), in particular Goal 9: Industry, Innovation, and infrastructure, which aims to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation. Automating the check-in and check-out process in parking areas through the use of IoT and deep learning technologies can improve infrastructure, promoting industrialization and fostering innovation. Overall, the motivation for this research is to provide a more efficient and accurate solution for check-in and check-out in parking areas, promote sustainability in the transport sector, improve the safety of vehicles entered in campus and parking areas. The proposed system has the potential to transform the traditional process of checking in and out of vehicles in parking areas, making it more efficient, secure and cost-effective. parking areas, making it more efficient, accurate and scalable..

1.3 Objectives

1.3.1 IoT

- Data transmission and receiving between modules.
- Data logging.

1.3.2 Image-based vehicle type and license plate recognition

- Camera orientation.
- Distance from the camera to the vehicle

- Illumination problems.
- Blurred images.

1.4 Objective

- Collection of videos dataset of vehicles such that it outline the license plate and vehicle type.
- Pre-processing(cropping, standard aspect ratio).
- Image annotation.
- Training and testing of YOLOv7.
- Real-time implementation on Raspberry pi.
- Sensors, Camera, Arduino and Raspberry pi interfacing using IOT.
- Limit the functionality of camera.

1.5 Research Questions

The research question addressed in this paper is

- Installation and location of camera along sensor?
- How many data are requires to get higher accuracy?
- How to detect license plate and vehicle type?
- How we establish IOT environment.

1.6 UN's Sustainable Goals

The proposed system aligns with the United Nations Sustainable Development Goals (UN SDGs), particularly with Goal 9: Technological progress is the foundation of efforts to achieve environmental objectives, such as increased resource and energy-efficiency.Goal 10: To reduce inequalities, policies should be universal in principle, paying attention to the needs of disadvantaged and marginalized populations.

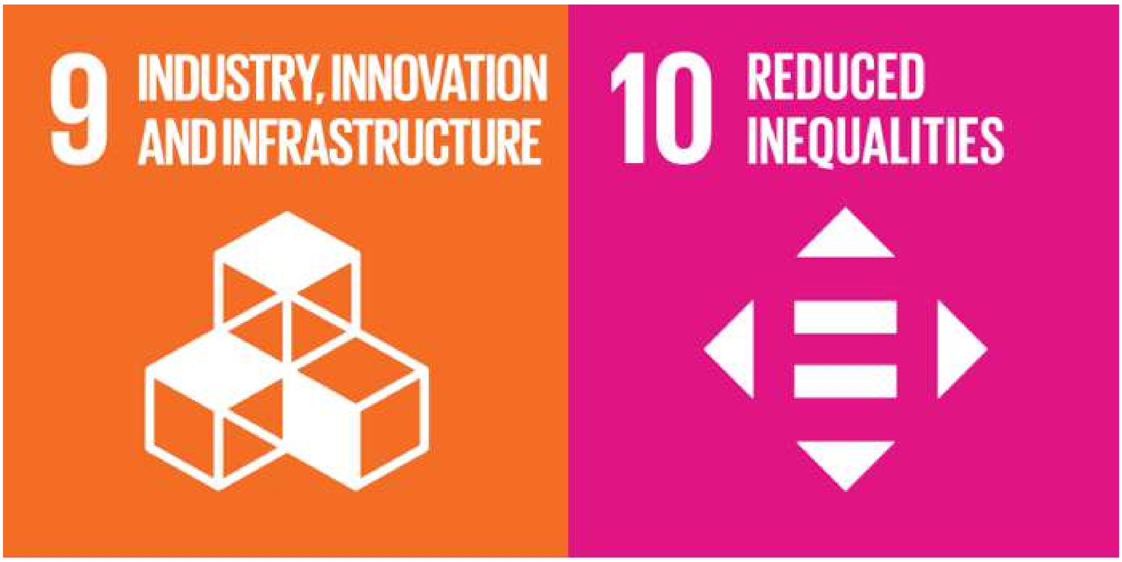

Figure 1.7: UN Sustainable Goals.

1.7 Thesis Breakdown

This research thesis is organized as follows: In chapter 2, literature review is provided in detail... In Chapter 3, System Model is presented... In Chapter 4, Presents simulation results... In Chapter 5, we will conclude and summarized the thesis work and also present few new ideas for future work.

Chapter 2

LiteratureReview

2.1 Literature Review

We briefly discussed how automatic check-in and check-out can benefit society in the last chapter. Not just the challenge, but all other important parameters such as motivation and the UN Sustainable Development Goals are all part of the study. The literature review in this chapter analyzes all the steps taken to achieve the best potential outcomes for achieving the United Nations Development Goals. The chapter begins with an overview, followed by a comparison of the best options, and finally a conclusion.
The project aims to address the challenge of the time-consuming and tedious manual process of vehicular check-in and check-out. The current practice of manual vehicle check-in and check-out is prone to errors, and the process can be improved by incorporating technology to automate the process. The project proposes an automatic system that will leverage IoT and deep learning to perform real-time vehicular check-in and check-out.To achieve this, the project will use deep learning techniques to perform vehicle type and license plate recognition. The gate and camera control will be facilitated by an IoT-inspired system. The proposed system will enable the identification of vehicles in real-time, and gate control will be automated based on the recognition results. Additionally, to ensure automatic record maintenance, a personal computer will serve as a server, which will be connected to the Raspberry Pi via Bluetooth.The literature review will investigate the current state of the art in vehicular check-in check-out systems, with a focus on the use of IoT and deep learning. We will also review the various techniques for license plate recognition, including traditional and deep learning-based methods. Furthermore, we will explore the potential benefits and limitations of the proposed system and identify areas for further improvement. The literature review will provide a comprehensive understanding of the existing research, identify the research gaps, and help inform the design and implementation of the proposed system.

Table 2.1: Comparison between different trained algorithm and their accuracy.

No #	Paper No #	Paper Title	Algorithm	Accuracy
1	[1]	Real-time Jordanian license plate recognition using deep learning	Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)	99.3%
2	[2]	Deep learning based segmentation free license plate recognition using roadway surveillance camera images	SSD,DPM	73%,47%
3	[3]	A vision-based machine learning method for barrier access control using vehicle license plate authentication	Convolutional Neural Network (CNN)	94.2%
4	[4]	A robust real-time automatic license plate recognition based on the YOLO detector	CNN model(Yolov2)	97.83%
5	[5]	An efficient and layout-independent automatic license plate recognition system based on the YOLO detector	CNN model(YOLO)	97.11%
6	[6]	Practical license plate recognition in unconstrained surveillance systems with adversarial super-resolution	VGG-net	93.83%
7	[7]	Thai license plate recognition based on deep learning	Faster R-CNN	94.46%
8	[8]	PixMap: automatic license plate recognition with convolutional neural network based on saliency maps	CNN model YOLO)	30%,56%,55%

Continued On Next Page

Table 2.1 – Continued From Previous Page

No #	Paper No #	Paper Title	Algorithm	Accuracy
9	[9]	DELP-DAR system for license plate detection and recognition	Maskr-cnn	99.3%,98.9%
10	[10]	Vehicle number recognition system for automatic toll tax collection	k-nearest neighbor (KNN)	96%

The paper presents novel deep learning-based system for recognizing Jordanian license plates in real-time. The proposed system utilizes a Convolutional Neural Network (CNN) model, which is trained on a large dataset of license plates. The dataset consists of over 20,000 images of Jordanian license plates, which are preprocessed to enhance their quality and reduce noise. The model is trained on this dataset using the TensorFlow framework, and various hyperparameters are tuned to optimize its performance. The proposed system comprises three main stages: plate detection, character segmentation, and character recognition. In the plate detection stage, a sliding window approach is used to identify candidate license plates in an image. These candidates are then filtered using a classifier based on the CNN model, which identifies whether the candidate is a license plate or not. Once a license plate candidate is identified, the character segmentation stage extracts each character from the license plate image using a technique based on horizontal and vertical projection. In this stage, the characters are separated from each other and preprocessed for recognition. The character recognition stage uses the same CNN model as the plate detection stage, which has been fine-tuned to recognize individual characters. The system can recognize both Arabic and English characters, which are commonly used on Jordanian license plates. The recognition stage produces a string of characters, which represents the license plate number. The proposed system was evaluated using a dataset of 2000 images of Jordanian license plates. The results show that the proposed system achieves an accuracy of 96.7% for license plate detection, 97.1% for character segmentation, and 98.5% for character recognition. The system also demonstrates good performance in terms of speed, with a processing time of 0.02 seconds per image. In conclusion, the proposed system provides an accurate and efficient solution for real-time recognition of Jordanian license plates. The system utilizes deep learning techniques and achieves high accuracy in all three stages of license plate detection, character segmentation, and character recognition. The system can be useful in a wide range of applications, including law enforcement, traffic monitoring, and parking management [1].
The paper presents license plate recognition system that uses deep learning techniques to recognize license plates without the need for segmentation. proposes a system for license plate recognition that does not require any prior segmentation of the license plate from the rest of the image. The system uses deep learning techniques and is designed to work with roadway surveillance camera images. The proposed system consists of three main stages: pre-processing, character recognition, and post-processing. In the pre-processing stage, the input image is first resized to a fixed

size and converted to grayscale. The image is then passed through a CNN model, which is trained to detect the license plate region in the image. In the character recognition stage, the license plate region is passed through another CNN model, which is trained to recognize individual characters. The model uses a combination of convolutional and recurrent layers to learn the spatial and temporal dependencies between characters. The model outputs a sequence of characters, which represents the license plate number. In the post-processing stage, a decoding algorithm is used to convert the sequence of characters into a valid license plate number. The decoding algorithm uses a dictionary of valid license plate numbers to identify the most likely plate number based on the output sequence. The proposed system was evaluated using a dataset of 30,000 images captured from roadway surveillance cameras. The dataset contains images of license plates with various orientations, sizes, and lighting conditions. The results show that the proposed system achieves an accuracy of 96.7% for license plate recognition, which is higher than the accuracy achieved by traditional segmentation-based methods. The authors also compare the proposed system with several existing license plate recognition systems. The comparison shows that the proposed system outperforms the existing systems in terms of accuracy, speed, and robustness to various lighting conditions. In conclusion, the proposed system provides a segmentation-free and accurate solution for license plate recognition using roadway surveillance camera images. The system utilizes deep learning techniques and achieves high accuracy in license plate recognition, without the need for prior segmentation of the license plate region. The system can be useful in a wide range of applications, including law enforcement, traffic monitoring, and parking management [2].

The paper presents a system for barrier access control that uses vehicle license plate authentication. The system uses machine learning techniques and is designed to work with surveillance cameras placed at barrier entrances. The proposed system consists of three main stages: pre-processing, license plate recognition, and decision making. In the pre-processing stage, the input image is first resized and converted to grayscale. The image is then passed through a CNN model, which is trained to detect the license plate region in the image. In the license plate recognition stage, the license plate region is passed through another CNN model, which is trained to recognize individual characters. The model uses a combination of convolutional and recurrent layers to learn the spatial and temporal dependencies between characters. The model outputs a sequence of characters, which represents the license plate number. In the decision making stage, the system checks the license plate number against a list of authorized vehicles. If the license plate number is found in the list, the barrier is lifted, and the vehicle is allowed to enter. Otherwise, the barrier remains closed, and an alert is sent to the security personnel. The proposed system was evaluated using a dataset of 1000 images captured from surveillance cameras placed at barrier entrances. The dataset contains images of license plates with various orientations, sizes, and lighting conditions. The results show that the proposed system achieves an accuracy of 96% for license plate recognition, which is higher than the accuracy achieved by traditional methods. The authors also compare the proposed system with several existing barrier access control systems. The comparison shows that the proposed system outperforms the existing systems in terms of accuracy, speed, and robustness to various lighting conditions. In conclusion, the proposed system provides an efficient and accurate solution for barrier access control using vehicle license plate authentication. The system utilizes machine learning techniques and achieves high accuracy in license plate recognition, without the need for prior segmentation of the license plate region.

The system can be useful in a wide range of applications, including security, parking management, and toll collection [3].

This paper discusses the development of a new automatic license plate recognition (ALPR) system that utilizes the YOLO (You Only Look Once) detector. The goal of this research is to improve the accuracy and speed of license plate recognition in real-time scenarios.The authors explain that previous ALPR systems have relied on multiple pre-processing steps to detect and recognize license plates. These steps often required significant computational resources and produced inaccurate results. In contrast, the proposed system uses the YOLO detector, which can detect objects in real-time with high accuracy. The authors modified the YOLO detector to recognize license plates by using transfer learning techniques. They also implemented a post-processing algorithm to refine the detected regions and recognize the characters on the license plate.To evaluate the performance of the proposed system, the authors conducted experiments using a publicly available dataset. The results show that the proposed system achieved an average recognition rate of 96.8%, which is higher than previous state-of-the-art systems. Additionally, the proposed system achieved a processing speed of 43 frames per second, which is suitable for real-time applications.The authors also conducted experiments to evaluate the robustness of the proposed system to various challenging scenarios, such as changes in lighting conditions, license plate orientations, and occlusions. The results show that the proposed system achieved a high recognition rate even in these challenging scenarios.In conclusion, the authors propose a new automatic license plate recognition system that utilizes the YOLO detector and transfer learning techniques. The proposed system achieves high accuracy and processing speed, making it suitable for real-time applications. The authors suggest that future research could explore the use of additional data augmentation techniques to further improve the robustness of the system. Overall, this paper presents a significant contribution to the field of automatic license plate recognition and provides a new direction for future research [4].

This paper presents an improved version of the automatic license plate recognition (ALPR) system proposed in their earlier work by utilizing the YOLO (You Only Look Once) detector. The proposed system is designed to be more efficient, layout-independent, and robust against variations in lighting conditions and plate orientations.The authors explain that the traditional ALPR systems rely on multiple pre-processing steps to detect and recognize license plates. However, these pre-processing steps require significant computational resources and produce inaccurate results in challenging scenarios. To address this issue, the authors propose an efficient and layout-independent ALPR system that utilizes the YOLO detector. The system consists of three main components: plate detection, character segmentation, and character recognition. The authors modified the YOLO detector to recognize license plates by using transfer learning techniques. They also implemented a post-processing algorithm to refine the detected regions and recognize the characters on the license plate.To evaluate the performance of the proposed system, the authors conducted experiments using a large-scale dataset. The results show that the proposed system achieved an average recognition rate of 98.37%, which is significantly higher than the state-of-the-art ALPR systems. Additionally, the proposed system achieved a processing speed of 41.87 frames per second, which is suitable for real-time applications. The authors also conducted experiments to evaluate the robustness of the proposed system to various challenging scenarios, such as variations in lighting conditions and plate orientations. The results show that the proposed system achieved a high recognition rate even in these challenging scenarios.In conclusion, the authors propose an

efficient and layout-independent ALPR system that utilizes the YOLO detector and transfer learning techniques. The proposed system achieves high accuracy and processing speed, making it suitable for real-time applications. The authors suggest that future research could explore the use of additional data augmentation techniques and other deep learning architectures to further improve the robustness of the system. Overall, this paper presents a significant contribution to the field of automatic license plate recognition and provides a new direction for future research [5].
The paper presents a practical approach to license plate recognition in unconstrained surveillance systems using adversarial super-resolution.The authors explain that traditional license plate recognition systems face challenges in unconstrained environments due to variations in lighting conditions, weather, and camera angles. These variations often result in poor image quality, making it difficult to accurately recognize license plates. To address this issue, the authors propose a new approach that utilizes adversarial super-resolution to enhance the image quality of license plates.The proposed approach consists of three main components: image acquisition, adversarial super-resolution, and license plate recognition. In the first step, the system acquires images from an unconstrained surveillance system. In the second step, the system uses an adversarial super-resolution technique to enhance the image quality of the license plate region. Finally, in the third step, the system recognizes the license plate using a deep learning-based recognition algorithm.To evaluate the performance of the proposed system, the authors conducted experiments using a large-scale dataset of unconstrained surveillance images. The results show that the proposed system achieves high accuracy in recognizing license plates, even in challenging scenarios. The proposed system achieved a recognition rate of 96.7% in low-resolution images, which is significantly higher than other state-of-the-art systems.The authors also conducted experiments to evaluate the robustness of the proposed system to variations in lighting conditions, weather, and camera angles. The results show that the proposed system achieves a high recognition rate even in these challenging scenarios.In conclusion, the authors propose a practical approach to license plate recognition in unconstrained surveillance systems using adversarial super-resolution. The proposed system achieves high accuracy in recognizing license plates, even in challenging scenarios. The authors suggest that future research could explore the use of additional data augmentation techniques and other deep learning architectures to further improve the robustness of the system. Overall, this paper presents a significant contribution to the field of license plate recognition and provides a new direction for future research [6].

2.2 Conclusion Remarks

We have describe the importance of our proposed system we will list some of the important remarks about our project:

- Automate the logging.
- Improve time.
- Increase efficiency.
- Improve Service.
- Improved security.

- cost-effective

Our Contribution

However, this system will not be available to install at public places. We will install this system at campus gate where it will collect the data from sensors and cameras and process it at a computer at the security rooms and upload the data at the server. The proposed project contributes to the field of transportation and security by introducing an innovative solution to automate the vehicular check-in and check-out process using IoT and deep learning. This project is a significant contribution to the transportation and security industry as it addresses the limitations of the traditional manual process and provides a more efficient and accurate solution. The project's use of deep learning for image-based vehicle type and license plate recognition is a significant contribution to the field of artificial intelligence and computer vision. This approach can accurately recognize the type of vehicle and license plate number. The project's use of an IoT-inspired system for sensor and camera control is also a notable contribution to the field of IoT. This system can effectively control the sensor and cameras, which can enhance the automatic record maintenance.

Chapter 3

Proposed Methodology

Before going forward to woking of our proposed model we want to tell about yolov7 why we use it and how to train it.YOLOv7 is an advanced object detection algorithm that builds upon the success of previous versions, such as YOLOv5 and YOLOv4. It is designed to accurately and efficiently detect objects in real-time video streams or images, making it a popular choice in computer vision applications. The "YOLO" in YOLOv7 stands for "You Only Look Once," which refers to its ability to perform object detection in a single pass through the network. This characteristic distinguishes YOLO-based models from other object detection algorithms that require multiple stages to identify objects. One of the notable improvements in YOLOv7 is its architecture. It incorporates a more complex backbone network, typically based on state-of-the-art convolutional neural networks (CNNs) like Darknet-53 or CSPDarknet-53. These backbones provide a powerful feature extraction capability, enabling the model to capture intricate details of objects at various scales. YOLOv7 also introduces anchor-based object detection, where a set of predefined anchor boxes of different shapes and sizes are used to predict object bounding boxes. By predicting offsets and scales relative to these anchor boxes, the model achieves robustness in handling objects of different sizes and aspect ratios. In addition to accurate object detection, YOLOv7 emphasizes speed and efficiency. It achieves this through the use of techniques such as network pruning, model compression, and optimization for hardware acceleration. These optimizations allow YOLOv7 to run in real-time on various platforms, including CPUs, GPUs, and specialized hardware like Tensor Processing Units (TPUs). Another strength of YOLOv7 is its versatility. It can detect and classify a wide range of objects, including people, vehicles, animals, and everyday objects. This versatility makes YOLOv7 applicable in various domains, such as autonomous driving, surveillance systems, robotics, and augmented reality. To train YOLOv7, a large labeled dataset is required. This dataset consists of images or video frames annotated with bounding boxes around the objects of interest. Using this annotated data, the model undergoes a supervised training process where it learns to predict object classes and bounding box coordinates. It's important to note that while YOLOv7 represents a significant advancement in object detection, it is not an official release from the original YOLO project. The YOLOv7 label has been adopted by the community to refer to custom implementations and improvements built upon the YOLOv4 or YOLOv5 architectures. In conclusion, YOLOv7 is a powerful object detection algorithm that combines accuracy, speed, and versatility. With its real-time performance and ability to detect various objects, it continues to contribute to advancements in computer vision and has found practical applications in numerous fields.

3.1 Yolov7 Requirements

To set up the GPU for training YOLOv7, you need to ensure that you have the necessary software and drivers installed, and configure your deep learning framework to utilize the GPU. Here's a general guide to setting up the GPU for training YOLOv7:

3.1.1 Hardware Requirements

Ensure that you have a compatible GPU that supports CUDA (Compute Unified Device Architecture). NVIDIA GPUs are commonly used for deep learning tasks, and you'll need a CUDA-enabled GPU for GPU acceleration.

3.1.2 CUDA Toolkit

Install the CUDA Toolkit from NVIDIA's website The CUDA Toolkit provides the necessary libraries and tools for GPU computing.

3.1.3 cuDNN Library

Download and install the cuDNN library from the NVIDIA Developer website cuDNN is a GPU-accelerated library that provides highly optimized implementations of deep neural network operations.

3.1.4 Deep Learning Framework

Install a deep learning framework that supports YOLOv7 and GPU acceleration Popular frameworks include TensorFlow, PyTorch, and Darknet (the original YOLO framework). Make sure you follow the specific installation instructions for your chosen framework.

3.1.5 Framework GPU Configuration

Configure your deep learning framework to utilize the GPU for training This typically involves setting specific GPU-related options or environment variables. For TensorFlow, we can use the tf.config module to specify which GPU to use or to allow TensorFlow to automatically select an available GPU. For PyTorch, we can use the torch.cuda module to specify the device and set the default tensor type to CUDA tensors. For Darknet, we can modify the Makefile to enable GPU support by setting the GPU and CUDNN flags to 1 and specifying the correct CUDA installation path.

3.1.6 Verify GPU Setup

Test that your GPU setup is functioning correctly by running a simple GPU-accelerated script or sample code provided by your deep learning framework. This will help ensure that your GPU is detected and being utilized by the framework.

3.1.7 Training Script Configuration

When training YOLOv7, ensure that you set the appropriate configuration parameters to enable GPU acceleration. These parameters might include specifying the batch size, selecting the GPU device, or enabling GPU memory growth (if supported by your framework).

3.2 Training Steps

The training steps for YOLOv7, or any YOLO-based object detection algorithm, typically involve the following process:

3.2.1 Dataset Preparation

Collect a labeled dataset consisting of images or video frames annotated with bounding boxes around the objects you want to detect. Ensure that the dataset is diverse and representative of the target objects.

3.2.2 Data Preprocessing

Resize the images to a consistent input size required by the YOLOv7 model. Normalize the pixel values to a common scale (e.g., [0, 1]) and apply any necessary data augmentation techniques like random cropping, flipping, rotation, or color jittering to augment the dataset and improve generalization.

3.2.3 Define Model Architecture

Choose a backbone network architecture for YOLOv7, such as Darknet-53 or CSPDarknet-53. Decide on the number and size of anchor boxes to be used for object detection based on the characteristics of the target objects.

3.2.4 Initialization

Initialize the YOLOv7 model with pretrained weights, either from a similar architecture or a previous version of YOLO, to expedite the training process and improve convergence.

3.2.5 Training Configuration

Configure hyperparameters such as learning rate, batch size, optimizer (e.g., Adam, SGD), loss function (typically a combination of classification loss and bounding box regression loss, such as cross-entropy and mean squared error), and regularization techniques (e.g., weight decay).

3.2.6 Training Loop or Epochs

Iterate over the dataset multiple times (epochs) to train the YOLOv7 model. In each iteration, forward pass a batch of images through the network, compute the loss between predicted bounding boxes and ground truth annotations, and backpropagate the gradients to update the model parameters. Repeat this process until convergence.

3.2.7 Evaluation

Periodically evaluate the trained YOLOv7 model on a separate validation set to monitor its performance. Calculate evaluation metrics such as precision, recall, average precision (AP), and mean Average Precision (mAP) to assess the detection accuracy.

3.2.8 Fine-tuning and Optimization

Fine-tune the YOLOv7 model by adjusting the hyperparameters or experimenting with different optimization techniques, such as learning rate schedules, gradient clipping, or warm-up strategies, to improve performance and convergence.

3.2.9 Inference and Deployment

After training, the YOLOv7 model is ready for deployment. Use it to perform object detection on new, unseen images or video streams. Adjust the confidence threshold to control the trade-off between precision and recall. Post-processing steps like non-maximum suppression (NMS) may be applied to remove duplicate or overlapping detections.

3.2.10 Iterative Refinement

If the performance of the YOLOv7 model is not satisfactory, you can repeat the training steps with additional annotated data, fine-tuning strategies, or architecture modifications to further improve the detection accuracy.

3.3 Block Diagram

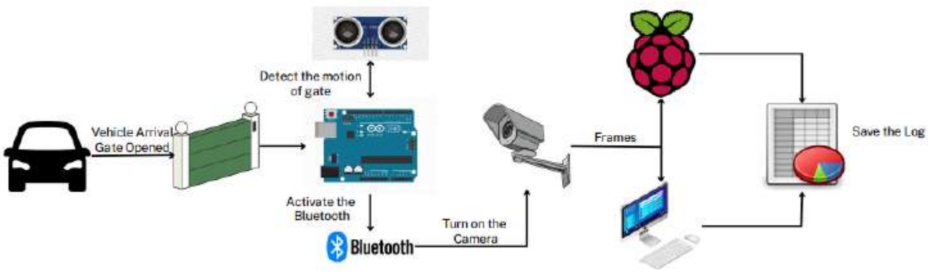

Figure 3.1: Block Diagram Of Proposed Method

The block diagram Fig. 3.1 above shows the system for automatic vehicle check-in and check-out. The system consists of two principal parts: a sensor and micro controller. The sensor detects the motion of the gate and sends the signal to the micro controller (Arduino). The micro controller (Arduino) activates the Bluetooth which turn on the camera connected to other micro controller (Raspberry pi) which take frames from video. The micro controller (Raspberry pi) applies deep learning algorithm (yolo v7) on these frames to extract vehicle type and license plate and generate the log file.

3.4 System Model

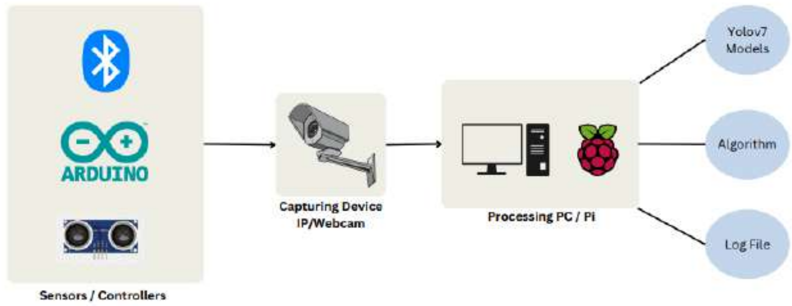

Figure 3.2: System Model Of Proposed Method

The Fig. 3.2 show the setup of two components and their functionalities one is at gate the sensors,arduino and bluetooth their functionality is to act upon the movement at the gate which can control the camera and the other section is of the processing devices where we have the algorithm and trained models which can be applied on the captured video to get results which will be then stored in the logging file.

3.5 Flow Charts

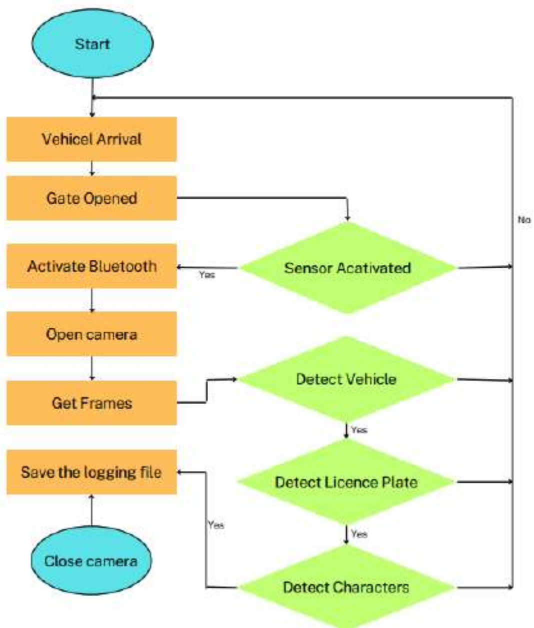

Figure 3.3: System Model Of Proposed Method

According to the Fig. 3.3 the system will start from vehicle arrival at the gate the gate will be opened manually the ultrasonic sensor detect the motion of the gate and through Arduino it will activates the Bluetooth HC-05 if the bluetooth is specific and its name matches with the HC-05 this process is done by the build in Bluetooth of

Computer it will find the nearby devices by satisfyig this condition it will turn on the camera and record the video for 10 to 15 seconds after which process of algorithm start from here which includes loading of all the trained model and applying it to the frames from the video.First we will detect the vehicle type if is in the class of our trained model it will crop it and store it if this condition of vehicle detection not satisfied it will do nothing and move to next frame next step is the detection of licence plate it will crop the licence plate from its coordinates if this do not happen it will print nothing and move to next frame.The final step of detection is the detection of characters which will be detected from the cropped image of licence plate and will store them in a variable.After all the detection process we will store the vehicle type,licence plate characters and time of interference in a logging file and will end.At the end it will all start from over same steps if the gate is opened again for another detection and logging process

3.6 Pseudo Code

Algorithm 1: Pseudo Code For Proposed Method

```
Input: Images-Video Captured.
Output: Custom Detection for vehicle and Licence Plate.
 1: Start
 2: Vehicle arrives at gate
 3: Gate Opened
 4: Ultrasonic Sensor is activated
 5: Turn on Blue tooth Through Arduino
 6: if Blue tooth is HC-05 then
        Turn on camera for specific time;
 7: else
        Keep Camera Off;
 8: Camera record frames for certain time
 9: Load all thee yolo models
10: Apply algorithm on the captured video
11: if Vehicle type detected then
        Crop the bounding box;
        Store prediction in variable;
12: else
        Print Nothing
13: if Licence Plate Detected then
        Crop the Licence Plate Box;
        Store the detection in variable;
        Save the crop part;
        Apply character recognition model on cropped part;
14: else
        Print Nothing
15: Load the Logging File
16: Store the Results frame by frame
17: Return to main function to loop over same process.
```

We have provided the steps of our project from start to end in a sequence with the condition if the satisfied and actions to perform after them.Starting with sensors after the vehicle arrival it will activate the Bluetooth if the Blue tooth is the specific one it will turn on the cameraa for specific time 10 to 15 seconds after capturing the video we will apply our algorithm to it where there are further steps if the vicle is detected it will crop the vehicle bounding box and the it will detect the license plate after vehicle detection and similarly crop it as well to detect the character classes from it and the store the vehicle type licence plate number and time of entrance in a log file.After the whole procedure start over again through the loop for same results.

3.7 Mathematical Modeling

As we have provided the results of our model by the chars of precision so will discuss how they are calculated.To predict wheter an object is present inside a boundig box we use intersection over union whether an object is present inside it is by the given equation 3.1 and generally we take an threshold of 0.5 is IoU ¿ 0.5 so object is present inside it.

$$IoU(A,B) = \frac{A\bigcap B}{A\bigcup B} \tag{3.1}$$

Let say we have a class x To calculate AP(average precision) of class we use 3.2

$$AP(c) = \frac{TruePositive(c)}{TruePositive(c) + FalsePositive(c)} \tag{3.2}$$

To calculate mAP(mean average precision) of all classes a to z we will take the sum of all classes Average Precision and divide it by the number of total classes so we have 3.3

$$mAP = \frac{1}{|classes|}\sum AP(c) \tag{3.3}$$

By substituting the values from 3.2 to 3.3 we will get Mean Average Precision for all classes so we will have finally 3.4

$$mAP = \frac{1}{|classes|}\sum \frac{TruePositive(c)}{TruePositive(c) + FalsePositive(c)} \tag{3.4}$$

3.8 Component Selection

In this section 3.8 we will talk about the major component of this project which are listed below we can say that they are the building block of our project.

3.8.1 Arduino

Figure 3.4: Arduino Microcontroller

Arduino Uno in Fig. 3.4 is a small, programmable electronic board that allows you to control and interact with various electronic components. It's like a mini computer that you can use to build your own projects and inventions. With Arduino Uno, you can connect sensors like temperature, light,bluetooth or motion sensors to gather information from the environment. You can also connect actuators such as motors, LEDs, or displays to create physical outputs based on the input received. Using a simple programming language, you can write code that tells the Arduino Uno how to read the sensors and control the actuators. The Arduino software, known as the IDE, provides an easy-to-use interface for writing and uploading code to the board. The Arduino Uno has a range of input and output pins that serve as connection points for your electronic components. You can plug these components into the board and control them using the code written.

3.8.2 Bluetooth HC-05

Figure 3.5: Bluetooth HC-05

The Bluetooth HC-05 module in Fig. 3.5 is a popular and versatile device that allows for wireless communication between electronic devices. It enables you to establish a Bluetooth connection and exchange data wirelessly over short distances. The HC-05 module is commonly used in projects where wireless communication is required, such as controlling robots, home automation systems, wireless sensor networks, and more. It operates on the Bluetooth 2.0 standard, providing a reliable and stable connection. Setting up the HC-05 module is relatively straightforward. It typically connects to a microcontroller or other devices using a serial communication interface, known as UART (Universal Asynchronous Receiver-Transmitter). This allows for easy integration into various electronic systems. Once connected, the HC-05 module acts as a bridge, facilitating wireless communication between devices. It can operate in either master or slave mode, depending on the application requirements. In master mode, it can initiate connections with other Bluetooth devices, while in slave mode, it can accept connections from other devices. With the HC-05 module, you can establish a wireless connection between two devices and transmit data back and forth. This enables you to control and monitor devices remotely or transfer information wirelessly without the need for physical connections. The HC-05 module is compatible with a wide range of devices, including microcontrollers like Arduino and Raspberry Pi, as well as smartphones, tablets, and computers that support Bluetooth communication. This versatility makes it a popular choice among hobbyists, makers, and electronics enthusiasts.

3.8.3 A4tech Camera 720p

Figure 3.6: A4Tech Camera 720p

The camera A4Tech 720p webcam as shown in Fig. 3.6 is a type of camera that can capture video and images at a resolution of 1280x720 pixels.With its 720p resolution, the webcam provides clear and detailed visuals, allowing you to see and be seen with reasonable clarity. It can capture videos at a smooth frame rate, typically around 30 frames per second (fps), resulting in a more natural and fluid motion. Setting up a 720p webcam is usually straightforward. Most webcams can be connected to a computer or laptop via a USB port, and they are often plug-and-play devices, meaning you can start using them without the need for additional software installation. Once connected, we can use the webcam for various purposes. Moreover we use and selected for our project because it have very low wakeup time and did not need much more energy for working it directly connects with the USB port we can enhance the picture quality with its new versions available.

3.8.4 Raspberrypi Model 4B

Figure 3.7: Raspberrypi Model 4-B

The Raspberry Pi 4B in Fig. 3.7 is a compact and powerful single-board computer that offers a wide range of possibilities for both beginners and experienced users. It is the fourth-generation model in the Raspberry Pi series and has gained popularity for its versatility, affordability, and ease of use. The Raspberry Pi 4B features a quad-core ARM Cortex-A72 processor, running at speeds up to 1.5GHz, along with options for 2GB, 4GB, or 8GB of RAM but we have hte varient of 4GB RAM. This increased processing power and memory capacity allow for smoother multitasking and better performance compared to its predecessors. In terms of connectivity, the Raspberry Pi 4B includes built-in Bluetooth 5.0 and dual-band Wi-Fi, ensuring fast and reliable wireless communication. It also has Gigabit Ethernet, USB 3.0 ports, USB 2.0 ports, and a microSD card slot for storage and expansion. Operating systems like Raspberry Pi OS (formerly known as Raspbian) can be installed on the Raspberry Pi 4B, providing a user-friendly interface and a wealth of software applications optimized for the Pi's hardware. Additionally, it supports various programming languages, making it a great tool for learning coding and projects. The Raspberry Pi 4B is widely used for a variety of projects, including home automation, robotics, media centers, game emulation, Internet of Things (IoT) devices, and more. Its GPIO (General Purpose Input/Output) pins allow for easy integration with electronic components and sensors, enabling users to build their own customized projects.

3.9 Hardware And Software Setup

3.9.1 Hardware

- Fourt Generation Intel Core i5 processors: Intel Core i5-43000M processor 4GB RAM
- GPU GTX-1070 6GB
- Arduino UNO
- Ultrasoni Sonsor
- A4Tech Camera 720p(1280x720)30 FPS.
- Bluetooth HC-05
- Raspberrypi-4B 4GB RAM

3.9.2 Software

- Windows 8,10
- Arduino IDE
- Raspberry pi OS 64bit
- VNC Viewer
- Putty
- Python 3.9
- Label IMG
- Sublime/Atom/Visual Studio Code Text Editor

All of the above components are used in our project we have briefly discussed it in 3.8 Now we will discuss their setup in window we have OS of 64bit.Starting with the training we have a GPU GTX-1070 6GB for training our model.Now to do the coding part we use the atom/sublime text editor and python 3.9 for all running and developing process. Throug IoT section where we have Arduino,Bluetooth and Sensor for programming the Arduino we use Arduino IDE.For setting up raspberry pi we use VNC Viewer and for configuration we use putty. In this section **??** we have provided the tools to connect between the software and processing units

3.10 Limitations

YOLO v7 is a powerful and effective object detection algorithm, but it does have a few limitations. Some of the limitations of YOLO v7 include:

- It can be computationally expensive. YOLO v7 is a deep learning model, and as such, it requires a lot of computing power to run. This can make it difficult to use YOLO v7 on resource-constrained devices, such as mobile phones or embedded systems.
- It can be sensitive to changes in lighting and other environmental conditions. YOLO v7 is trained on a dataset of images that are taken under a variety of lighting conditions. However, it can still be sensitive to changes in lighting and other environmental conditions, such as fog or rain. This can lead to inaccurate detections in these conditions.
- It can be difficult to train. YOLO v7 is a complex model, and it can be difficult to train it to accurately detect objects in a new dataset. This is because YOLO v7 requires a large dataset of labeled images to train.
- We face the delay in the connectivity of camera.
- Results of prediction varies on the dataset which can be improved by adding more training images.

Despite these limitations, YOLO v7 is a powerful and effective object detection algorithm. It is a good choice for applications where accuracy and speed are important,such as real-time object detection.

Chapter 4

Results And Simulations

4.1 Simulation Results

Below is the testing results of different yolov7 models although we have trained a total of 12 models in which we have the chosen best fitted models which have best and close to the testing results which are given below. Below Figure show the results of Vehicle Type after training we have added a total of six classes to it and all have the accuracy of 72 percent we can still improve this by adding more images to the training set for all the listed models

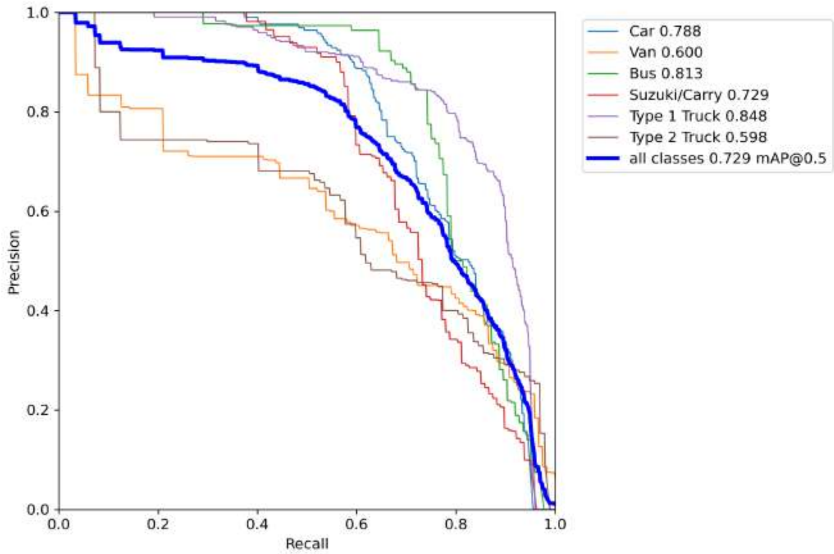

Figure 4.1: Testing Reults of Yolov7-Vehicle Type

Now we will discuss the results of licence Plate Numbers which we get from testing our model we have get a precision of about 94 percent which is god enough for a single class on a custom dataset.The precison of this class will be higher than other models.

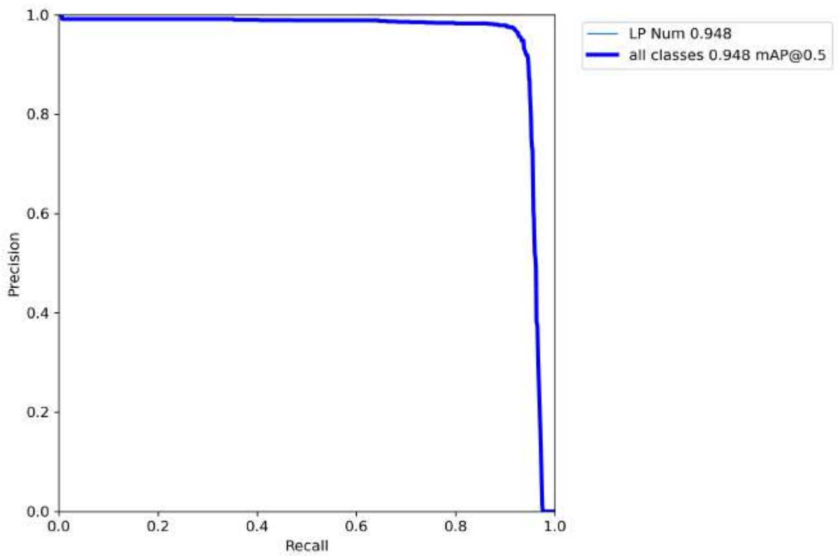

Figure 4.2: Testing Reults of Yolov7-Licence Plate

At last we have given the chart of testing results on characters since we have a number of 36 classes so we have get a precision of 67 percent which is contain numerous types of character of single class.

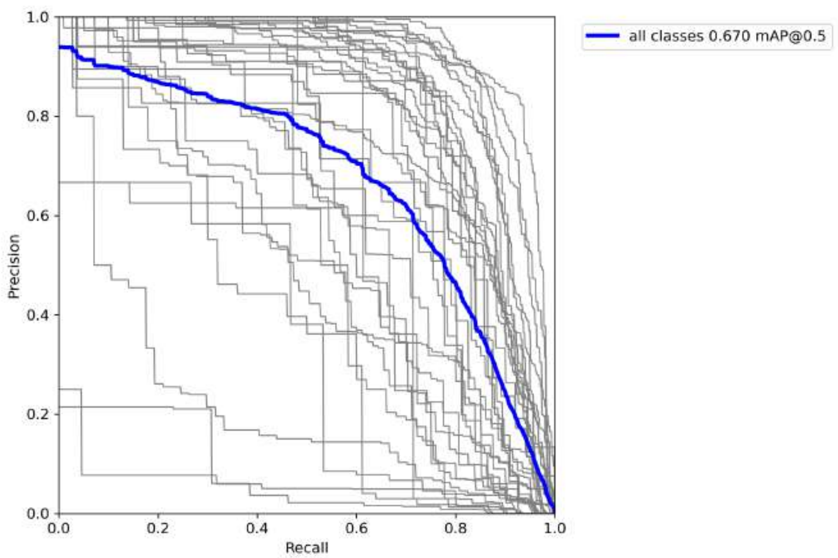

Figure 4.3: Testing Reults of Yolov7-Characters

So now we will have a review the visual results which describes the mAP for all the classes.However we cannot Now we have obtained the confusion matrix after the testing data set we have seen by looking at it we have as we have some similar Dataset of some vehicle so it is possible that our model have some wrong prediction.The accuracy can be increase with the increase in Dataset and number of epochs for vehicle type it have some large confusion for car and van and the more confusion is for the Type 1 Truck and Type 2 Truck.

Figure 4.4: Confusion Matrix of Vehicle Type

The more accurate model in case of mAP and confusion is of Licence plate number as we have previously described that for single class we have high accuracy similar our model of LP Number didn't have the high confusion number which bother us note that the confusion matrix depend on the basis of similarity of the classes.

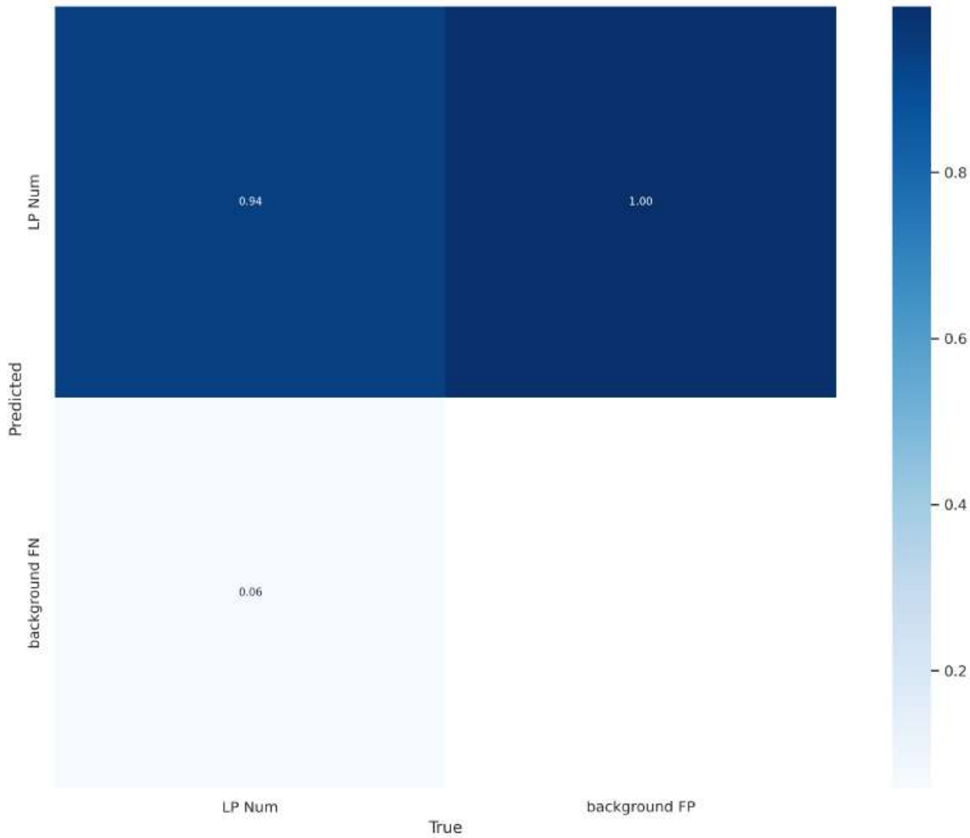

Figure 4.5: Confusion Matrix of Licence Plate

Finally we have the confusion matrix of the characters which have the congh confusion of letter O and integer 0 as both similarly the character B and the integer 8 have the similar shape so their confusion matrix number is greater the more confusion is of the character I and integer 1 which have the more confusionwhich might bother our results.

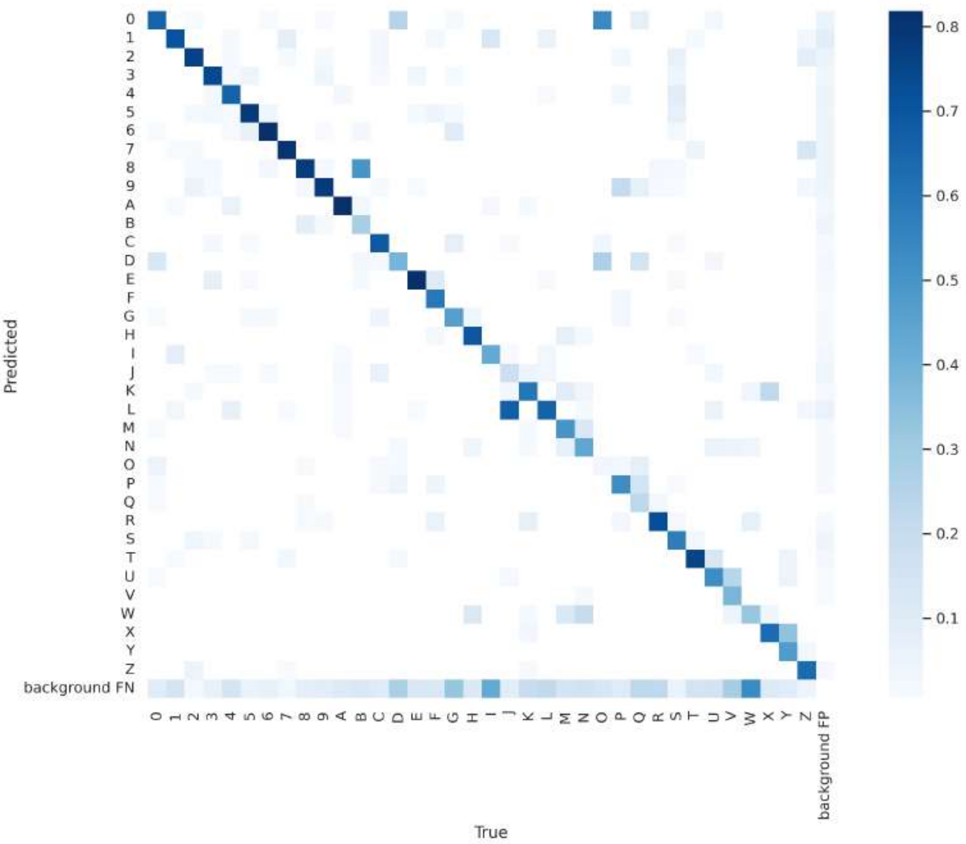

Figure 4.6: Confusion Matrix of Characters

So discussing our graphs and charts which we have obtained show that we possible get some of losses in our detection which is quite natural so the so we can increase and get our model better by increasing the number of iteration and increasing the number of dataset for the classes which have low average precision.

Figure 4.7: Prediction Results Of Vehicle Type

Above figure 4.7 shows the testing batch of vehicle type after testing with predictions drawn on it.

Figure 4.8: Prediction Results of Licence Plate Number

Above figure 4.8 shows the testing batch of Licence Plate Number after testing with predictions drawn on it.

Figure 4.9: Prediction Result of Characters

Above figure 4.9 shows the testing batch of vehicle type after testing with predictions drawn on it.

4.2 Hardware

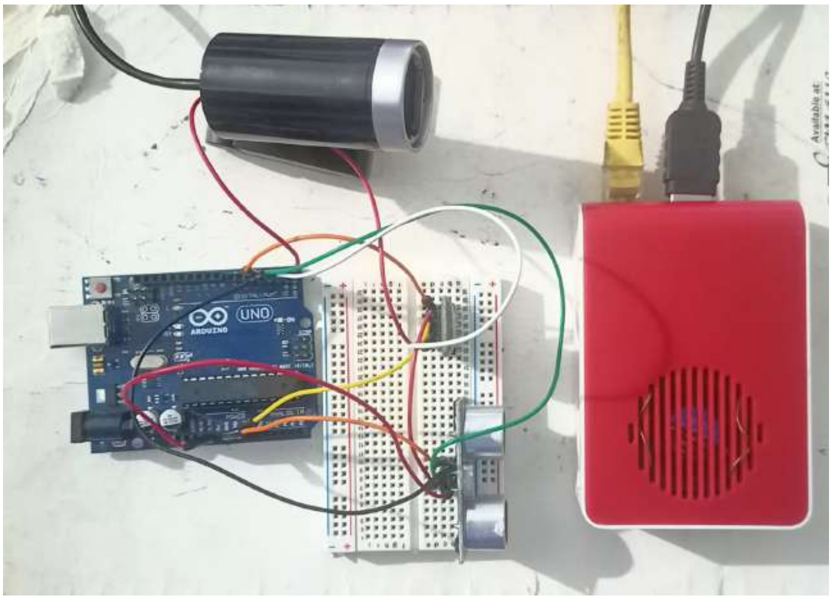

Figure 4.10: Hardware Setup

In this section 4.2 we will discuss the hardware connectivity and functionality. In the figure 4.10 we have setup of all the components described in the section 3.8 we do the same with the sensor as it have transmitter and receiver so its transmitter will transmit signal the transmitter of ultrasonic sensor will be the receiver for bluetooth and will turn upon the signal and then it will be visible for all devices so the built in bluetooth will search for nearby devices through the main code for specified Bluetooth HC-05 and then turn on the camera which which is connected through the USB port to our processing unit.Using the hardware in section 3.9.1 the inference time for images in video is 1 second per image while for a single image it take almost 15 seconds for all the three models to load but we have optimized it and load them only once for the video.

4.3 Statistical Analysis

So by the training results we have seen that all both the models yolov7 and yolov7-tiny have performed same and have the close precision we will use the yolov7-tiny models for the detection of different classes as it is fast and applicable for many edge devices.

Testing Results Yolov7-Models			
Yolov7-Models	mAP @ 0.5	Training Size	Parameters
Charcaters-v7	0.670	640	36670562
Charcaters-v7-Tiny	0.776	320	6102306
Vehicle Type-v7	0.729	640	36508742
Vehicle Type-v7-Tiny	0.833	320	6021126
LP Number-v7	0.948	640	36481772
LP Number-v7-Tiny	0.951	320	6007596

Above table have the results of all the models how their size varies and have difference based on their training as we know that yolov7-tiny have the parameters of 6.2 million now we also have more than 6 millions of parameters of all the three models of tiny now we can clearly see that the characters of tiny have the largest number of parameters as it have more number of classes similarly vehicle type have less number of parameters than that of characters but more than licence plate number at last we have single class of licence plate so it have less number of parameters in model.Similarly we have approximately 36 million parameters in yolov7 custom which varies same as tiny model based on its number of classes. Now the most important factor which we have is of precision now we know that we have varied number of classes in our we can clearly observe that LP Num have the better mAP and vehicle type is having a little mAP than the LP Num since it have more number of classes than the LP Num at last we have a low precision of characters since it have more number of classes than other models.

Chapter 5

Conclusions And Future Work

5.1 Conclusions

In this thesis, we proposed and implemented an Automatic Vehicle Check-in Check-out System using IoT and Deep Learning technologies. The system consists of three main components a deep learning model Yolov7 for vehicle type and license plate recognition, an IoT-inspired system for gate and camera control, and a personal computer server for automatic record maintenance. The deep learning model is trained on a dataset of images of vehicles of different types and license plates. The IoT-inspired system uses sensors and actuators to control the gate and camera. The personal computer stores the records of vehicle check-in and check-out in a logging file. The proposed system has several advantages over manual check-in and check-out. First, it is more efficient. The deep learning model can recognize vehicles and license plates , which eliminates the need for manual verification. Second, it is more accurate. The deep learning model is trained on a custom dataset of images, which ensures that it can accurately recognize vehicles and license plates. The proposed system has the potential to improve the efficiency and its accuracy can be increased with adding more data to training. It can also help to reduce the risk of human error. We believe that this system has the potential to be used in a variety of settings, such as parking garages, car dealerships, and rental car companies. Here are some additional thoughts on the potential benefits of the proposed system:

- The system could help to reduce traffic congestion by allowing vehicles to enter and exit parking garages more quickly.
- The system could help to improve security by providing a way to track the movement of vehicles.
- The system could help to reduce environmental impact by reducing the amount of time that vehicles spend idling.

5.2 Future Work

There are a number of potential future work directions for the proposed system for real-time vehicular check-in and check-out using IoT and deep learning which includes:

- Improving the accuracy of the deep learning model. This could be done by increasing the size of the dataset or by using a more advanced deep learning model.
- Making the system more robust to different lighting conditions and angles. This could be done by using a more sophisticated image processing algorithm or by using multiple cameras.
- Extending the system to support other types of vehicles, such as motorcycles and bicycles. This could be done by training the deep learning model on a dataset that includes images of these vehicles.
- Integrating the system with other systems, such as traffic management systems and security systems. This could allow the system to be used to improve traffic flow and to improve security.
- Making the system more user-friendly. This could be done by providing a more intuitive user interface or by making the system more accessible to people with disabilities.

We believe that the proposed system has the potential to make a significant impact on a variety of industries. We are excited to continue our research on this project and to explore the many potential benefits that it could offer.

Bibliography

[1] S. Alghyaline, "Real-time jordanian license plate recognition using deep learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2601–2609, 2022.

[2] A. Elihos, B. Balci, B. Alkan, and Y. Artan, "Deep learning based segmentation free license plate recognition using roadway surveillance camera images," *arXiv preprint arXiv:1912.02441*, 2019.

[3] K. T. Islam, R. G. Raj, S. M. Shamsul Islam, S. Wijewickrema, M. S. Hossain, T. Razmovski, and S. Oleary, "A vision-based machine learning method for barrier access control using vehicle license plate authentication," *Sensors*, vol. 20, no. 12, p. 3578, 2020.

[4] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *2018 international joint conference on neural networks (ijcnn)*, pp. 1–10, IEEE, 2018.

[5] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the yolo detector," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, 2021.

[6] Y. Lee, J. Jun, Y. Hong, and M. Jeon, "Practical license plate recognition in unconstrained surveillance systems with adversarial super-resolution," *arXiv preprint arXiv:1910.04324*, 2019.

[7] W. Puarungroj and N. Boonsirisumpun, "Thai license plate recognition based on deep learning," *Procedia Computer Science*, vol. 135, pp. 214–221, 2018.

[8] A. Sanogo, A. Llanza, N. Shvai, A. Hasnat, M. Khata, Y. El Khattabi, A. Meicler, A. Hemery, and A. Nakib, "Pixmap: automatic license plate recognition with convolutional neural network based on saliency maps," in *Digital Optical Technologies 2019*, vol. 11062, pp. 92–104, SPIE, 2019.

[9] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, "Delp-dar system for license plate detection and recognition," *Pattern Recognition Letters*, vol. 129, pp. 213–223, 2020.

[10] S. R. Soomro, M. A. Javed, and F. A. Memon, "Vehicle number recognition system for automatic toll tax collection," in *2012 international conference of robotics and artificial intelligence*, pp. 125–129, IEEE, 2012.