Automated Access Control for Restricted Zone

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering

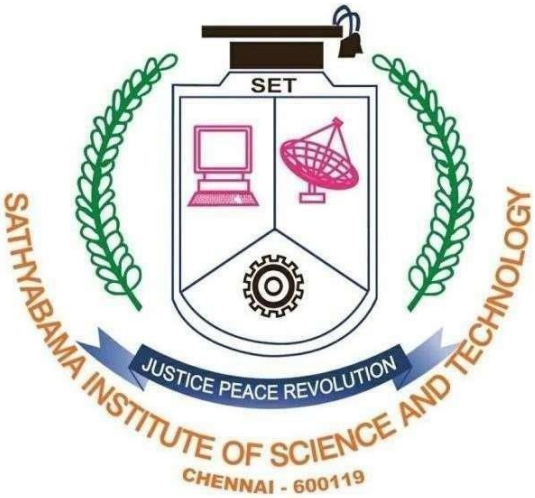
by

N U SURYA KIRAN - 42130311

S M V SHIVA - 42130285

M SIVA - 42130271

P BHANU PRAKASH - 42130331



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A++" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

NOVEMBER - 2024

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

Accredited with Grade "A++" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of N U SURYA KIRAN (42130311), SALADI MOHANA VENKATA SIVA (42130285), MEDIDINNE SIVA (42130271), PAKAM BHANU PRAKASH (42130331) who carried out the project entitled "**Automated Access Control for Restricted Zones"** under our supervision from July 2024 to November 2024.

Internal Guide

MS. S. ANNIE ANGELINE PREETHI, M.E., (PH.D.),

HEAD OF THE DEPARTMENT

Dr. T. RAVI, M.E., Ph.D.,

Submitted for viva voce Examination held on_____.

INTERNAL EXAMINER **EXTERNALEXAMINER**

DECLARATION

We N U SURYA KIRAN (42130311), SALADI MOHANA VENKATA SIVA (42130285), MEDIDINNE SIVA (42130271, PAKAM BHANU PRAKAS (42130331) hereby declare that the design thinking Report entitled "**Automated Access Control for Restricted Zones** "done by us under the guidance is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE(S)

1.

2.

3.

4.

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Professor & Dean, School of Electrical and Electronics and Dr. T. RAVI, M.E., Ph.D., Professor & Head, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our **Internal Guide Ms. S Annie Angeline Preethi M.E., (PH.D.),** or her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Vehicle Number Plate Recognition is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or area around top government offices e.g. Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address, etc. The system is implemented and simulated in Matlab, and it performance is tested on real image. It is observed from the experiment that the developed system successfully detects and recognize the vehicle number plate on real images. Automatic license plate recognition (ALPR) is the extraction of vehicle license plate information from an image or a sequence of images. The extracted information can be used with or without a database in many applications, such as electronic payment systems (toll payment, parking fee payment), and freeway and arterial monitoring systems for traffic surveillance. The ALPR uses either a color, black and white, or infrared camera to take images. The quality of the acquired images is a major factor in the success of the ALPR. ALPR as a real-life application has to quickly and successfully process license plates under different environmental conditions, such as indoors, outdoors, day or night time. It should also be generalized to process license plates from different nations, provinces, or states. These plates usually contain different colors, are written in different languages, and use different fonts; some plates may have a single colour background and others have background images. The license plates can be partially occluded by dirt, lighting, and towing accessories on the car. In this paper, we present a comprehensive review of the state-of-the-art techniques for ALPR. We categorize different ALPR techniques according to the features they used for each stage, and compare them in terms of pros, cons, recognition accuracy, and processing speed.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Vehicle Number Plate Recognition (VNPR) system is an important technique, used in Intelligent Transportation System. VNPR is an advanced machine vision technology used to identify vehicles by their number plates without direct human intervention. It is an important area of research due to its many applications. The development of Intelligent Transportation System provides the data of vehicle numbers which can be used in follow up, analyses and monitoring. VNPR is important in the area of traffic problems, highway toll collection, borders and custom security, premises where high security is needed, like Parliament, Legislative Assembly, and so on. The complexity of automatic number plate recognition work varies throughout the world. For the standard number plate, VNPR system is easier to read and recognize. In India this task becomes much difficult due to variation in plate model. The ANPR work is generally framed into the steps: Number plate extraction, character segmentation and character recognition. From the entire input image, only the

number plate is detected and processed further in the next step of character segmentation. In character segmentation phase each and every character is isolated and segmented. Based on the selection of prominent features of characters, each character is recognized, in the character recognition phase. Extraction of number plate is difficult task, essentially due to: Number plates generally occupy a small portion of whole image; difference in number plate formats, and influence of environmental factors. This step affects the accuracy of character segmentation and recognition work. Different techniques are developed for number plate extraction. The license plates can be partially occluded by dirt, lighting, and towing accessories on the car. In this paper, we present a comprehensive review of the state-of-the-art techniques for ALPR. We categorize different ALPR techniques according to the features they used for each stage,and compare them in terms of pros, cons, recognition accuracy, and processing speed. Future forecasts of ALPR are given at the end. Due to a huge number of vehicles, modern cities need to establish effectively automatic systems for traffic management and scheduling. One of the most useful systems is the VehicleLicense-Plate (VLP) Recognition System.

CHAPTER 2
LITERATURE SURVEY

Sharma. J et al (2018) **"Performance Analysis of Vehicle Number Plate Recognition System Using Template Matching Techniques"** Vehicle number plate recognition (VNPR) system is a digital image processing techniques which is broadly used in vehicle transportation system to identify the vehicle by their number plate. Yet it's a very challenging problem, due to the diversity of plate formats, different scales, and non-uniform illumination conditions during image acquisition. This research mainly focuses on Nepali vehicle number plate recognition system in which the vehicle plate image is received by the digital cameras and the image was then processed to obtain the number plate information. A real image of a vehicle is captured and processed using various algorithms. Morphological operations, and edge detection, smoothing, filtering, techniques for plate localization and characters segmentation for segment character and this segmented character was cut in to block of 70×70 size and calculate the correlation with the template of database using the template matching algorithm normalized cross correlation and phase correlation and compare this result in term of accuracy. The system was tested by 90 patterns under several conditions. It includes experiment of number plate recognition using phase correlation and normalized cross correlation methods. From the study and analysis of test after applying on number of images of database, the normalized cross correlation method was found more accurate to recognize the number plate then phase correlation method and recognition accuracy of normalized cross correlation was 67.98% and phase correlation was 63.46%.

Li Yao et al 2019 "**Research and Application of License Plate Recognition Technology Based on Deep Learning"** There are many types of vehicle license plates in China, including new energy license plates, large truck license plates, government vehicle license plates, and military license plates. The existing commercial license plate recognition system only targets common license plates and does not completely cover the full range of license plates. Therefore, this paper proposes an SSD-based end-to-end license plate recognition system (LPR-SSD). The LPRSSD network architecture consists of upper and lower classification networks: the upper layer network is used for vehicle license detection and classification, and the lower layer network is used for license plate character detection and classification. In order to enhance the

generalization performance of the LPR-SSD network, in addition to the real license plate image captured by the camera, this paper synthesizes 50K simulated license plates for each type of license plate according to the legal document [1]. Experiments show that LPR-SSD achieved a faster convergence speed during training. After the test set verification, the accuracy of license plate location detection and classification reaches 98.3%, and the character recognition accuracy rate reaches 99.1%.

Jethi Shashi Rangana et al (2020) "**Automated License on Plate Recognition**" With the explosive growth in the number of vehicles in use, automated license plate recognition (ALPR) systems are required for a wide range of tasks such as law enforcement, surveillance, and toll booth operations. The operational specifications of these systems are diverse due to the differences in the intended application. For instance, they may need to run on handheld devices or cloud servers, or operate in low light and adverse weather conditions. In order to meet these requirements, a variety of techniques have been developed for license plate recognition. Even though there has been a notable improvement in the current ALPR methods, there is a requirement to be filled in ALPR techniques for a complex environment. Thus, many approaches are sensitive to the changes in illumination and operate mostly in daylight. This study explores the methods and techniques used in ALPR in recent literature. We present a critical and constructive analysis of related studies in the field of ALPR and identify the open challenge faced by researchers and developers. Further, we provide future research directions and recommendations to optimize the current solutions to work under extreme conditions

J M S V Ravi Kumar et al (2021) "**Automatic Vehicle Number Plate Recognition System Using Machine Learning**" The big combination of software things, under different perspectives of our current day globe, has led to the transformation of vehicles into conceptual resources in the field of data technologies [1]. Since the information application is meaningless without any information as the name implies, there is a huge requirement to redesign the vehicle information among the information systems. It is also be achieved by external agents manually [2]or by special smart system which permits detection and recognition of vehicles by the license plates in real-time environments. Between smart equipment, previously mentioned is made of the system of identification of the number plates of the vehicles [3]. This model of vehicle number plate identification and recognition is used to identify the number plates and then make the detection of the plate that is to get the actual text from the picture and by using the

modules of calculation, which use location algorithms [4], segmentation plate and character identification. Common usages are - Car parking usage,Electronic Toll Payment , Variety of vehicles in the road and Areas with less as well as more congestion.

`**Omar Gheni Abdulate ef** et al (2022) "**Vehicle License Plate Detection Using Deep Learning**" This study focuses on using a Deep Convolutional Network trained with data from license plates to automatically categorize and geo locate vehicles (DCNN). Toll collection, accident reconstruction, and the identification of suspicious vehicles are just some real-world applications that use license plate readers. The study recommended using a vehicle classifier based on deep learning to pinpoint the location of license plates and license numbers simultaneously. Bounding quadrilaterals are provided by the classifier instead of bounding rectangles, which provides a more accurate indication for vehicle registration estimation to license plate localization. This task was accomplished using the Python programming language and various deep learning libraries. Since the training of the proposed DCNN model began with a weight that had already undergone a certain number of iterations in a model without a classification head, the total number of training iterations will be close to 10,000 when taking into account the transfer learning component of DCNN. Because of transfer learning, the DCNN model could begin at a good place, making it simpler to enhance functional heads at once. According to the study's characterization of the task at hand vehicle number estimation as well as license plate segmentation and vehicle-the DCNN achieved 98.8% accuracy in classification

Chapter 3

Aim and Scope of the Project

Image Acquisition. The first step is to capture images or video frames of vehicles, typically using cameras placed at strategic locations. These images serve as the primary input for the recognition system. Preprocessing. Preprocessing involves improving the image quality for better recognition. This step includes:Gray-scale Conversion: Convert the image to gray-scale for easier processing. Noise Reduction: Apply filters to reduce noise and enhance the quality of the number plate area . Edge Detection: Use techniques like the Canny Edge Detection to highlight the edges of the number plate. Localization of Number Plate.The vehicle number plate is detected within the image. Various techniques such as morphological operations or contour detection are used to identify rectangular regions that could potentially be the number plate.After localizing the plate segmentation involves separating individual characters on the number plate. This is done using techniques like connected component analysis or thresholding to isolate each character. Character RecognitionOptical Character Recognition (OCR) algorithms are used to identify and convert the segmented characters into readable text. Common algorithms include: Template Matching: Comparing each character with predefined templates. Machine Learning/Deep Learning Models: Utilizing neural networks or support vector machines (SVM) to classify characters .Post-Processing .Post-processing helps refine the results by validating the detected number plate characters. This can include Checking the format of the number plate (as per country/state regulations) .Correcting any misidentified characters using dictionary or rule-based methods. Database Integration. The recognized number plate is compared against a database for further actions such as retrieving vehicle details, issuing fines, or logging entries.

3.1 AIM:

This project aims to develop a system that can automatically detect and recognise license plates from images of vehicles. The main objective of this project is to create a robust and accurate vehicle number plate recognition system that can identify license plate numbers from images taken under varying conditions, including different lighting, angles and distances.

3.2 Scope of the Project:

- Develop an image processing pipeline to enhance the quality of input images.

- Implement license plate detection algorithms to locate number plates within images.
- Employ Optical Character Recognition (ocr) techniques to recognize individual characters on the license plates.
- Assemble all recognized characters to form the complete license plate number.
- Design a user-friendly interface for interacting with the system.

There are seven primary algorithms that the software requires for identifying a license plate:

- Plate localization – responsible for finding and isolating the plate on the picture
- Plate orientation and sizing – compensates for the skew of the plate and adjusts the dimensions to the required size
- Normalization – adjusts the brightness and contrast of the image
- Character segmentation – finds the individual characters on the plates
- Optical character recognition
- Syntactical/Geometrical analysis – check characters and positions against country-specific rules
- The averaging of the recognised value over multiple fields/images to produce a more reliable or confident result, especially given that any single image may contain a reflected light flare, be partially obscured, or possess other obfuscating effects.

CHAPTER 4
PROPOSED METHOD

4.1EXPERIMENTAL MATERIALS AND METHODS

4.1.1ARDUINO

Arduino comprises of both a physical programmable circuit board (commonly known as a microcontroller) and a programming software, or IDE (Integrated Development Environment) that can be run on a PC, used to compose and transfer PC code to thecircuit board. It can be done by using the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Unlike other programmable circuit boards, the Arduino does not require a different equipment (called a software engineer) to upload code to the circuit board, one can essentially utilize a USB link. Also, the Arduino IDE utilizes a rearranged rendition of C++, making it simpler to figure out how to program. In a word, Arduino make the functions of the micro-controller into a more accessible package. The Uno is one of the more prevalent boards in the Arduino family and an extraordinary option for the beginners.

Arduino, an open-source electronics platform, has transformed the landscape of do- it-yourself (DIY) electronics and prototyping. Founded on user-friendly hardware and software, Arduino empowers enthusiasts, students, and professionals to create a wide array of innovative electronic projects. At the core of Arduino lies a microcontroller that can be programmed to perform various tasks, from simple blinking LED lights to complex robotic movements and IoT applications. Arduino, an open-source electronics platform, has transformed the landscape of do- it-yourself (DIY) electronics and prototyping. Founded on user-friendly hardware and software, Arduino empowers enthusiasts, students, and professionals to create a wide array of innovative electronic projects. At the core of Arduino lies a microcontroller that can be programmed to perform various tasks, from simple blinking LED lights to complex robotic movements and IoT applications.

One of the key strengths of Arduino is its versatility, facilitated by a variety of board types tailored to diverse user needs. The Arduino Uno, a flagship model, stands out for its ease of use and adaptability, making it an ideal choice for beginners. Its numerous digital and analog input/output pins, coupled with a USB interface, enable seamless programming and interaction with the physical world. For projects demanding

compactness, the Arduino Nano offers similar functionality in a smaller form factor, suitable for wearable devices and space-constrained applications Arduino Uno is one of the most widely used Arduino boards. It features an ATmega328P microcontroller, digital and analog input/output pins, USB connection for programming, and power jack. Uno is ideal for beginners due to its simplicity and versatility, making it suitable for a wide range of project. Designed for projects requiring a large number of digital and analog inputs/outputs. It utilizes the ATmega2560 microcontroller, providing ample resources for complex applications. Mega boards are commonly used in robotics, 3D printers, and projects involving multiple sensors and actuators. Arduino Uno:

Arduino Uno is one of the most widely used Arduino boards. It features an ATmega328P microcontroller, digital and analog input/output pins, USB connection for programming, and power jack. Uno is ideal for beginners due to its simplicity and versatility, making it suitable for a wide range of projects.

Arduino Nano is a compact board based on the ATmega328P microcontroller. It is similar to Arduino Uno but smaller in size. Nano boards are often used in projects where space is a constraint, such as wearable devices and small-scale electronics projects.

Arduino Mega is designed for projects requiring a large number of digital and analog inputs/outputs. It utilizes the ATmega2560 microcontroller, providing ample resources for complex applications. Mega boards are commonly used in robotics, 3D printers, and projects involving multiple sensors and actuators.

Arduino Due stands out for its 32-bit ARM Cortex-M3 processor, offering significantly higher processing power than other Arduino boards. It is suitable for applications requiring advanced computation capabilities, such as high-performance robotics and data- intensive projects. It utilizes the ATmega2560 microcontroller, providing ample resources for complex applications. Mega boards are commonly used in robotics, 3D printers, and projects involving multiple sensors

One of the most widely used Arduino boards. It features an ATmega328P microcontroller, digital and analog input/output pins, USB connection for programming, and power jack. Uno is ideal for beginners.

4.1.3 Software

Download the Arduino IDE ARDUINO 1.8.10 The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux.

The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for Installation instructions

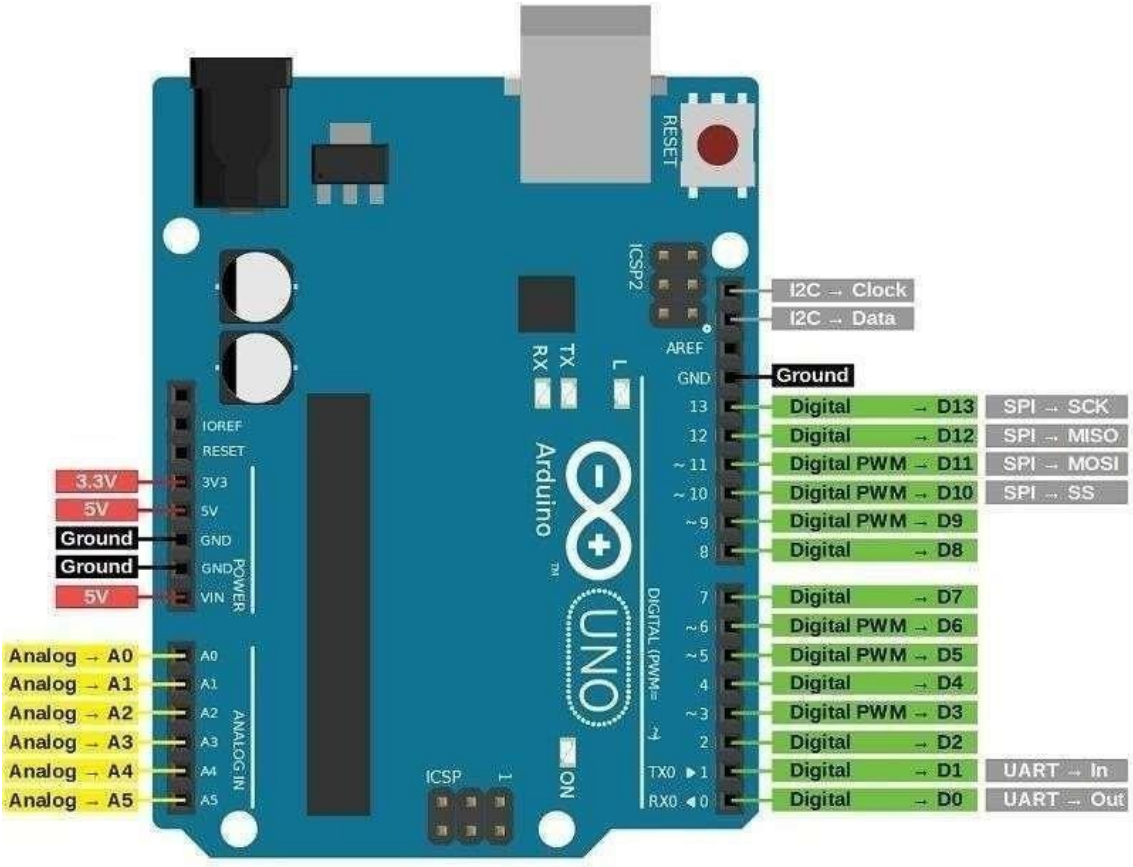


Fig: 4.1 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on wiring), and the Arduino software (IDE), based on processing.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming.As soon as it reached a wider community, the Arduino board started changing to adaptto new needs and challenges, differentiating its offer from simple 8- bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. AllArduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too,is open-source, and it is growing through the contributions of users worldwide.

4.1.4 Servo Motor

The servo motor is a versatile and widely used electromechanical device that provides precise control over angular position. It operates based on feedback control, where the position of the motor's shaft is continuously monitored and adjusted to match the desired position. In Arduino-based projects, servo motors are popular for tasks requiring controlled and specific movements, such as opening and closing doors, rotating objects, or controlling robotic arms.

Servo motors consist of a DC motor, a set of gears, a position sensor (usually a potentiometer), and control circuitry. The control circuitry interprets the PWM (Pulse Width Modulation) signals received from the Arduino board and adjusts the position of the motor shaft accordingly. By varying the duration of the PWM signals, typically ranging from 1 millisecond to 2 milliseconds, the servo motor can rotate to a specific angle within its operational range, usually 0 to 180 degrees.

One of the key advantages of servo motors in Arduino projects is their precise and controlled movement. They can hold a specific position with high accuracy, making them ideal for applications where exact positioning is crucial.

Additionally, servo motors are relatively easy to use, requiring minimal external components for operation. They are also compact and lightweight, making them suitable for applications with limited space constraints.

Servo motors can be powered directly from the Arduino board's power supply or an external power source, depending on the motor's power requirements. Arduino libraries simplify the programming of servo motors, allowing developers to easily control their movement using simple and intuitive code.

Fig:4.2 Servo Motor

4.2 APPLICATIONS:

- Section control, to measure average vehicle speed over longer distances
- Border crossings
- Automobile repossessions
- Petrol stations to log when a motorist drives away without paying for their fuel
- A marketing tool to log patterns of use

Fig:4.3 Number plate

- Traffic management systems, which determine traffic flow using the time it takes vehicles to pass two automatic number plate recognition sites

- Analyses of travel behaviour (route choice, origin-destination etc.) for transport planning purposes
- Drive-through customer recognition, to automatically recognize customers based on their license plate and offer them the items they ordered the last time they used the service
- To assist visitor management systems in recognizing guest vehicles
- Police and auxiliary police
- Car parking companies
- To raise or lower automatic bollards
- Hotels
- Enforcing Move over laws for emergency vehicles

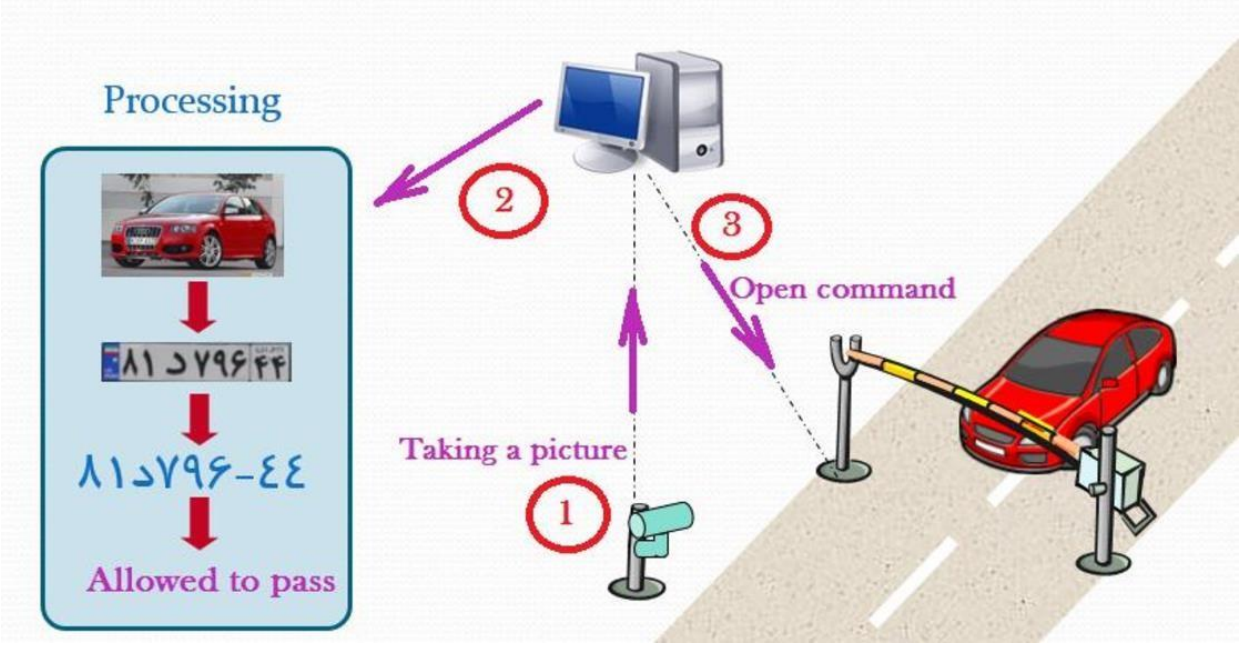


Fig:4.4 Processing

4.3 WORKING PRINCIPLE:

This project will utilize image processing technique to enhance image quality, integrate object detection models to identify license plates, employ OCR models to recognize characters and implement logic to assemble characters into license plate numbers.

4.3.1 STEPS INVOLVED IN LICENCE PLATE RECOGNITION:

License Plate Detection: The first step is to detect the License plate from the car. We will use the contour option in OpenCV to detect for rectangular objects to find the number plate. The accuracy can be improved if we know the exact size, color and approximate location of the number plate. Normally the detection algorithm is trained based on the position of camera and type of number plate used in that particular country. This gets trickier if the image does not even have a car, in this case we will an additional step to detect the car and then the license plate.

Character Segmentation: Once we have detected the License Plate we have to crop it out and save it as a new image. Again this can be done easily using OpenCV.

Character Recognition: Now, the new image that we obtained in the previous step is sure to have some characters (Numbers/Alphabets) written on it. So, we can perform OCR (Optical Character Recognition) on it to detect the number.

4.3.2. BLOCK DIAGRAM:

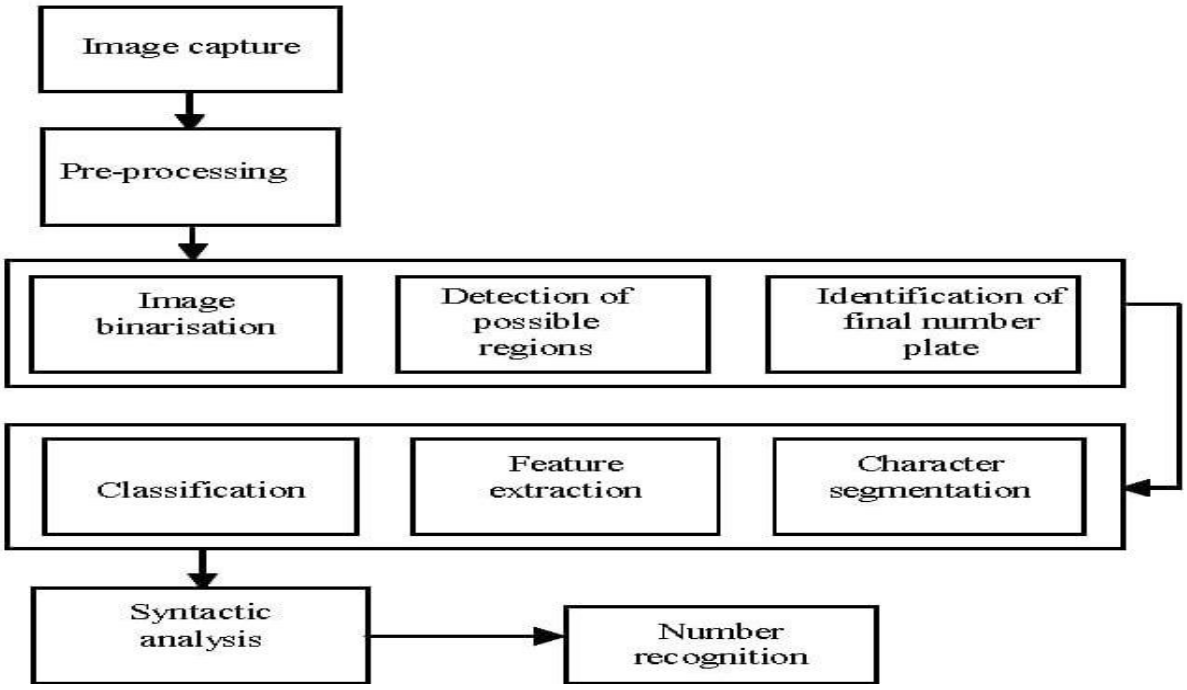


Fig:4.5 Block Diagram

4.3.3. SOFTWARE USED:

Fig:4.6 Python Software

Here we use python programming language to design our software. Python is a highlevel, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. **Python** can be used on a server to create web applications. Python is one of the widely used programming languages for image processing applications. Its amazing libraries and tools helps in achieving the tasks of image processing more efficiently.

4.3.4. IMAGE PROCESSING TOOL:

It stands for Open Source Computer Vision Library. This library consists of around 2000+ optimised algorithms that are useful for computer vision and machine learning. There are several ways you can use opencv in image processing, a few are listed below:

- Converting images from one color space to another i.e. like between BGR and HSV, BGR and gray etc.
- Performing thresholding on images, like, simple thresholding, adaptive thresholding etc.
- Smoothing of images, like, applying custom filters to images and blurring of images.
- Performing morphological operations on images.

- Building image pyramids.
- Extracting foreground from images using GrabCut algorithm.

With this library you can also perform simple image techniques, such as flipping images, extracting features, and analyzing them. Images can be represented by numpy multi-dimensional arrays and so their type is NdArrays. A color image is a numpy array with 3 dimensions. By slicing the multidimensional array the RGB channels can be separated. Below are some of the operations that can be performed using NumPy on the image (image is loaded in a variable named test_img using imread).

- To flip the image in a vertical direction, use np.flipud(test_img).
- To flip the image in a horizontal direction, use np.fliplr(test_img).
- To reverse the image, use test_img[::-1] (the image after storing it as the numpy array is named as <img_name>).
- To add filter to the image you can do this:

Example: np.where(test_img > 150, 255, 0), this says that in this picture if you find anything with 150, then replace it with 255, else 0.

- You can also display the RGB channels separately. It can be done using this code snippet:

To obtain a red channel, do test_img[:,:,0], to obtain a green channel, do test_img[:,:,1] and to obtain a blue channel, do test_img[:,:,2].

4.3.5 Pytesseract

PyTesseract is a Python wrapper for Google's Tesseract-OCR Engine, which is widely used for optical character recognition (OCR). It allows developers to extract text from images, making it useful for tasks like digitizing printed documents, reading text from screenshots, or automating data extraction from images.

Here are some key points about PyTesseract:

- **Text Extraction**: PyTesseract can recognize and extract text from images of various formats like PNG, JPEG, BMP, etc. It converts this extracted text into machine-readable formats, enabling automation and data processing tasks

- **Integration with Image Libraries**: PyTesseract works seamlessly with libraries like PIL (Pillow) or OpenCV, which can be used to preprocess images (e.g., cropping, scaling, or converting to grayscale) to improve OCR accuracy.
- **Multilingual Support**: The Tesseract-OCR engine supports a wide range of languages, which PyTesseract can use to extract text from images in different languages.
- **Open Source**: It's open-source and available for free, which makes it highly accessible for developers working on text recognition or document digitization projects.
- **Practical Applications**: PyTesseract is useful for a variety of applications like automating the process of reading receipts, scanning documents for text, and extracting information from business cards or forms.

4.3.5 PROGRAM:

4.3.5.1 Python code:

```
import cv2
import pytesseract
import serial as s
import time
arduino = s.Serial('COM5', 9600)
time.sleep(1)

# Configure tesseract path if required (for Windows users)
# pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-
OCR\tesseract.exe'

# Start capturing video from the webcam
cap = cv2.VideoCapture(1)

while True:
    ret, frame = cap.read()
    if not ret:
        break
```

```
    # Convert the frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Use pytesseract to extract text from the frame
    text = pytesseract.image_to_string(gray, config='--psm 6')

    # Filter out only alphabets and numbers, ignoring spaces
    filtered_text = ''.join(filter(str.isalnum, text))

    # Display the filtered text on the frame
    cv2.putText(frame, filtered_text, (30, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 255, 0), 2, cv2.LINE_AA)

    # Show the output frame
    cv2.imshow('Camera Feed - Alphabets and Numbers Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('c'):
        print ('value is '+filtered_text)
        try:
            arduino.write((filtered_text + '\n').encode())
        except:
            print('port not opened')
    # Press 'q' to exit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the capture and close any OpenCV windows+
arduino.close()
cap.release()
cv2.destroyAllWindows()
```

4.3.5.2 Arduino code:

```
#include <Servo.h>
Servo myServo;  // Create a servo object
```

```
int servoPin = 9;  // Pin connected to the servo
String command;  // Variable to store the incoming command
int ledPin = 13;
void setup() {
  Serial.begin(9600);  // Start the serial communication
  pinMode(ledPin, OUTPUT);
  myServo.attach(servoPin);  // Attach the servo to the pin
}

void loop() {
  if (Serial.available() > 0) {
    command = Serial.readStringUntil('\n');  // Read the incoming command

    if (command.equals("TN33J1364") || command.equals("IN33J1364")) {
      myServo.write(90);
      digitalWrite(ledPin, HIGH);
      delay(3000);
      myServo.write(0);
      digitalWrite(ledPin, LOW);
      delay(1000);
    }
  }
}
```

The script displays the recognized plate number and attempts to map the first two characters (assumed to be the state code) to the state using the `states` dictionary. It also displays rectangles around the detected plates and prints the recognized plate number on the image.

Fig:4.7 View of detected number

CHAPTER 5
RESULT AND CONCLUSION

5.1 RESULT

They can enhance security, aid law enforcement, reduce manual labor, and VNPR may face challenges in accurately reading license plates under certa conditions, such as poor lighting, dirt, or obstructed plates. Additionally, the integration of ultrasonic sensors for object and person detection proved effective. When someone approached the waste bin, the system activated the door mechanism, allowing for convenient and hygienic waste disposal. In cases where the bin was already full, the red indicator light effectively signaled unavailability, preventing further waste deposition and encouraging responsible waste disposal behavior among users.

The system's performance was further validated through extensive field tests in various urban environments. These tests confirmed the reliability of the sensors, the accuracy of the alert mechanisms, and the overall robustness of the IoT infrastructure. User feedback from these tests indicated high levels of satisfaction, emphasizing the system's user- friendliness and its positive impact on the community's waste disposal practices.

5.2. CONCLUSION

Our experimental study leads to two conclusions:

1. Different applications are better tackled by the solution with compatible settings to guarantee efficiency and performance; and
2. each module and the overall proposed solution are competitive to existing approaches.

CHAPTER-6
REFERENCES

Aniket.A.More & Akshay.V.Jadhav & Sweta Patil, “Surveillance Robot for Defence Environment”, VOLUME 6 I ISSUE 2 I APRIL — JUNE 2019, IJRARInternational Journal of Research and Analytical Reviews.

Chanda P, Mukherjee PK, Modak S, Nath A. Gesture Controlled Robot using Raspberry pie 3 and Android. International Journal. 2016 Jun; 6(6).

Jenifer, T. Maria, et al. "Mobile Robot Temperature Monitoring System Controlled by Android Application via Bluetooth." International Journal on Advanced Computer Theory and Engineering (IJACTE) 2.3 (2013).

Kanza Zafar, Ibrahim M. Hussain, “Rope Climbing Robot with Surveillance Capability”, International Journal of Intelligent Systems and Applications (IJISA, IJISA Vol. 5, No. 9, August 2013.

Naik YM, Deshpande CM, Shah RR, Kulkarni RR. ANDROID CONTROLLED SPY-ROBOT. International Journal of Software and Web Sciences. 2013:54-7.

Pahuja, Ritika, and Narender Kumar. "Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller." Electronics & Communication Engineering, Department, BRCM College of Engineering & Technology, Bahal, India, International Journal of Scientific Engineering and Research (IJSER) www. ijser. in ISSN (Online) (2014): 2347-3878.

Saravana Kumar K, Mannu Nayyar,Reshma M, Biju Joseph” Android controlled robot with image transfer” . International Journal of Advanced Multidisciplinary Research 2(3): (2015): 98–102.