The 7 Habits of Highly Effective People

Be Proactive.

Begin with the end in mind.

Put first things first.

Think Win-Win.

First Understand, then be Understood.

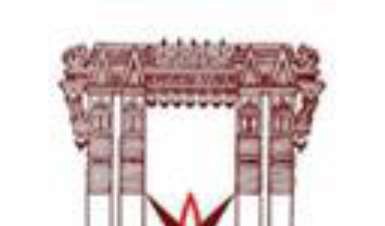
Synergize.

Sharpen Your Saw.

Practicing at

**VIGNANA JYOTHI GROUP OF INSTITUTIONS**

We have followed the above 7 steps during our project work.



|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Swetha Goparaju** |  | **ROLL No:20071A04K9** |
| **Sohan Kuchangari** |  | **ROLL No:20071A04N2** |
| **Manish Vemula** |  | **ROLL No:20071A04Q9** |

**License Plate Detection Algorithm Using PYNQ-Z2 Board**

**A MINI PROJECT REPORT**

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

**Submitted By**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Swetha Goparaju** |  | **ROLL No:20071A04K9** |
| **Sohan Kunchangari** |  | **ROLL No:20071A04N2** |
| **Manish Vemula** |  | **ROLL No:20071A04Q9** |

**UNDER THE SUPERVISION OF**

**B.B.Shabarinath**

**ASSOCIATE PROFESSOR**

**VNRVJIET**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY**

NBA Accredited CE, EEE, ME, ECE, CSE, EIE, IT -B.Tech Programs Approved by AICTE, New Delhi, Affiliated to JNTUH

VIGNANA JYOTHI NAGAR, BACHUPALLY, NIZAMPET (S.O.), HYDERABAD

500090. TELANGANA, INDIA.

#### **ACKNOWLEDGEMENTS**

We Indebted to **Dr. C. D. NAIDU**, Principal, VNRVJIET, for his help and guidance in our work. We consider ourselves fortunate to have obtained his friendly and valuable advice during the course of my research.

Our sincere thanks to **Dr. S RAJENDRA PRASAD**, Professor, Head of the Department, ECE, VNR VJIET for his esteemed guidance and encouragement provided during the course of our project.

We would like to express our sincere thanks to **B.B.Shabarinath sir**, Associate Professor, VNR VJIET for his precious guidance and kind co-operation at every step of this project work.

We thankful to all the staff members of ECE department, VNR VJIET for helping us during this project.

We thankful to all the project committee members of ECE department, VNR VJIET for helping us during this project.

Finally, we are very thankful to our family members and our friends for their great moral support.

|  |  |
| --- | --- |
|  |  |
| **Swetha Goparaju** |  | **ROLL No:20071A04K9** |
| **Sohan Kunchangari** |  | **ROLL No:20071A04N2** |
| **Manish Vemula** |  | **ROLL No:20071A04Q9** |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

ABSTRACT

Numerous computer vision applications, including automated toll collection, traffic monitoring, and vehicle recognition, depend on the accurate localization of license plates. In this project, license plate localization is implemented using the PYNQ-Z2 board, a flexible platform that combines FPGA (Field-Programmable Gate Array) capabilities with a Zynq-7000 SoC (System on Chip). This project focuses on implementing license plate localization using the OpenCV library and Python programming language. The suggested remedy makes use of the hardware acceleration capabilities of the PYNQ-Z2 to boost the effectiveness of license plate recognition. Using OpenCV, Python, and the PYNQ-Z2 board together, a strong foundation for license plate localization is produced.

**INDEX**

|  |  |
| --- | --- |
| DESCRIPTION | Page |
| Abstract 1 | |
|  |  |
|  |  |
| Table of Contents | 2-3 |
|  |  |
|  | |
| Chapter 1: Introduction…………………………………. |  |
| 1.1 Introduction | 4 |
| 1.2 Literature Survey |  |
| 1.2.1 Literature Survey-1 | 5 |
| 1.2.2 Literature Survey-2 | 5 |
| 1.2.3 Literature Survey-3 | 5 |
| 1.2.4 Literature Survey-4 | 6 |
| Chapter 2: Methodology………………………………… |  |
| 2.1 Working | 7 |
| 2.2 Block Diagram | 8 |
| 2.2.1 Description of Block diagram  2.3 Circuit Diagram  2.3. 1 Description of Circuit diagram | 8  9  9 |
| Chapter 3: Process and Requirements…………………. |  |
| 3.1 Software Requirements |  |
| 3.1.1 Jupyter Notebook | 10 |
|  |  |
| 3.1.2 Open CV | 11 |
| 3.2 Components Description |  |
| 3.2.1 Pynq-Z2 Board | 11-12 |
|  |  |
|  |  |
| Chapter 4: Coding…………………………………… |  |
| 4.1 Coding | 13-14 |
| Chapter 5: Advantages…………………………….… |  |
| 5.1 Advantages | 15 |
| Chapter 6: Result and Conclusion…………………… |  |
| 6.1 Result | 16-17 |
| 6.2 Conclusion  Chapter 7: References………………………………... | 17 |
| 7.1 References | 18 |
|  |  |
|  |  |
| LIST OF FIGURES |  |
|  |  |
|  |  |
| DESCRIPTION PAGE  Block Diagram 8  Circuit Diagram 9  Jupyter Notebook 10  Open CV 11  Pynq-Z2 Board 12    Result 16-17 |  |
|  |  |

**CHAPTER 1: INTRODUCTION**

**1.1 Introduction:**

Number plate detection, a crucial aspect of modern traffic management, security, and surveillance systems, plays a pivotal role in identifying vehicles and enforcing traffic regulations. Automating the process of extracting number plates from images or video streams has garnered significant attention due to its potential to enhance efficiency and accuracy in various applications.

PYNQ Z2 BOARD FEATURES: PYNQ-Z2 board integrates USB and Ethernet to connect to internet, HDMI Input/Output, MIC Input, Audio Output, Arduino interface, Raspberry Pi interface, 2 Pmod, user LED, pushbuttons, switches, MicroSD Slot, Power In port for direct power connection, a Jumper for power source selection and another jumper to select Boot Mode. It is intended to be easily extensible with Pmod, Arduino, and peripherals, along with general purpose GPIO pins.

This system consists of an android smartphone to continuously update location of vehicle with the help of drivers mobile. This device is called as tracking device which is continuously accessing its current location and is sending updates to monitoring device. At monitoring device which is an android application is providing the user with the exact location of the vehicles of his interest

The PYNQ-Z2 board stands as an exemplar of innovation in the field of embedded systems. Combining a high-performance Arm processor with an FPGA fabric, the PYNQ-Z2 provides a versatile platform for implementing complex algorithms.Leveraging this unique combination, we present a comprehensive exploration of a number plate detection system using the PYNQ-Z2 board. The increasing need for automated vehicle identification and traffic management has spurred the demand for accurate and real-time number plate detection systems.

**1.2 LITERATURE SURVEY:**

We have studied the literature surveys which gives the following information:

Balaji Rameshkumara, Lokesh Govindasami , Harirajkumar Jagannathan ,Sasirekha Natarajan Department of Electronics and Communication Engineering, Sona College of Technology, Salem, India. This is from the paper AIP Conference Proceedings 2857, 020018 (2023).

The programme recognizes automobile license plates entering and leaving the entry exit ramp is the result of the software presented in this paper. It would also be feasible to focus identification on a few specific frames and minimize computation demands now even more with effective application of tracking methods supplied by the OpenCV library. Another benefit of this technique is that it is not dependent on the nation of origin. The programme can detect license plates from any nation.

Prof. Shivganga Patil, Ashwini Associate Professor, P.G. Scholar Department of Digital Communication and Networking Engineering Sharnbasva University, Kalaburagi, India. This is from the paper of Intenational Journal of Creative Research Thoughts(IJCRT).

The purpose of this search is to demonstrate that free and open supply applied sciences are developed enough to be used in scientific computer environments. Device is able to handle a wide range of lighting conditions and unique types of number plates that are often seen in India. Even if there are certain limitations, it's still a better option than present proprietary solutions.

Swapna Premasiri, Lahiru Wijesinghe, Randika Perera Department of Electrical & Computer Engineering, Sri Lanka Institute of Information Technology. This is from the paper of Swapna Premasiri et al. Int. Journal of Engineering Research and Applications.

In this system, VGA data was read with the VGA clock and histograms of the isolated number plate were taken. As it was being read, the sum along the lines were calculated and the sum was saved in to a RAM. Similarly, VGA Y data was read along with the clock and was saved in to a separate RAM. the functionality of this system is as expected, provided that the system functions within the given assumptions.

Raghav Bhatia, Sarthak Jha Department of Electrical & Computer Engineering, Galgotias college of Engineering and technology, Greater Noida, India. This is from the paper of International Conference On Advance Computing and Innovative Technologies in Engineering(ICACITE).

The authors proposed that a vehicle number recognition plays a important role in many areas. The input image was tested through bilateral filter. The cropped image of the vehicle number goes into a folder in which the text image was converted into string.Tesseract was used to read the output of cropped image and the result was shown in the output terminal of python.

**CHAPTER 2: METHODOLOGY**

**2.1 WORKING:**

**Step-1: Reading the Image from the Same Directory**

This step involves loading an image from the same directory as your notebook or project. A Python library like PIL or OpenCV can be used for this.

**Step-2: Image Processing Operation**

This process entails a number of actions to improve the image for improved number plate recognition:

Colour conversion: If necessary, change the image's colour space. Changing from RGB to grayscale, as an example.

Apply filters or other techniques to the image to decrease noise. Gaussian blur or median blur are common methods.

Equalise Histogram: This method enhances the image's contrast.

**Step-3: Morphological Operation and Thresholding**

In order to process a picture, morphological processes utilise structural components (like kernels). Erosion and dilation processes are frequently employed to reduce noise or enhance forms.

When thresholding, a picture is turned into a binary image in which each pixel is either black or white depending on the threshold value.

**Step-4: Edge Detection and Dilation of Image**

To find edges in the image, edge detection methods (like Canny edge detector) are used. You can use these edges for additional processing.

The morphological process of dilation enlarges the edges of the objects in a picture.

**Step-5: Localizing License Plate using the Contouring Method**

A technique for identifying and analysing item boundaries in an image is contouring. In this step, contours will be used to find probable areas that might have a licence plate.

**Step-6: Showing the Localized Image - License Plate**

You will extract and display the prospective licence plate region individually after localising it.

**Step-7: Original image with the bounding boxes**

On the original image, draw bounding boxes around the detected licence plate(s) to visually emphasise the position.

**2.2 BLOCK DIAGRAM:**

The block diagram of the vehicle tracking system is shown below. This shows overall view of the system. The block that are connected here are the components used in this project.

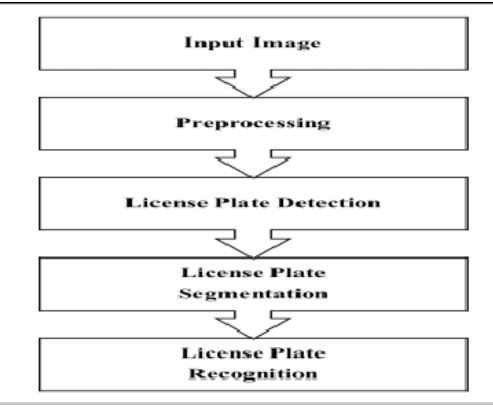


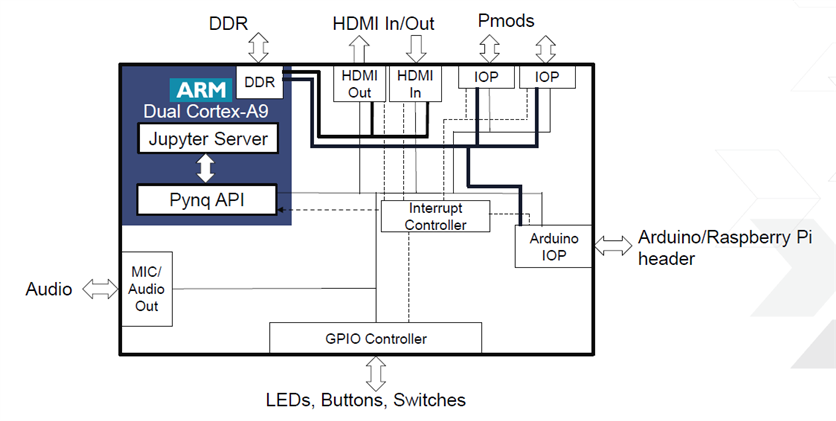
FIG: BLOCK DIAGRAM OF VEHICLE PLATE DETECTION

**2.2.1 DESCRIPTION OF BLOCK DIAGRAM:**

This procedure utilises OpenCV to recognise and separate licence plates from images on a PYNQ-Z2 board. Make use of the pre-installed OpenCV library and the Python environment on the PYNQ-Z2 board.

Bring up a pre-defined sample image. Use methods like noise reduction, colour conversion, and scaling to improve the quality of your images. Identify probable plate edges by using the Canny edge identification technique. To discover contours in the image with edges recognised, use OpenCV's findContours function. To find possible plates, analyse contours based on their aspect ratio, area, etc. Mark off a box around each potential licence plate. With the help of OpenCV and a PYNQ-Z2 board, this method shows how to locate licence plates.

**2.3 CIRCUIT DIAGRAM:**



**2.3.1 DESCRIPTION OF CIRCUIT DIAGRAM:**

The Xilinx Zynq-7000 SoC, which blends FPGA fabric with ARM Cortex-A9 CPUs, is the foundation of the PYNQ-Z2 board. For peripheral connections, it offers USB, Ethernet, HDMI, DDR3 RAM, on-board flash, and audio connectors. The board contains GPIO, Pmod connectors, and an Arduino header for hardware customisation. It contains switches, LEDs, and user buttons for user interactivity.

JTAG and USB-UART connectors facilitate programming and communication. The addition of a microSD card slot and programmable jumpers increases flexibility. Since this board was designed for FPGA development with Python programmability, it can be used for a range of embedded applications.

**CHAPTER 3: PROCESS & REQUIREMENTS**

**3.1 SOFTWARE REQUIREMENTS:**

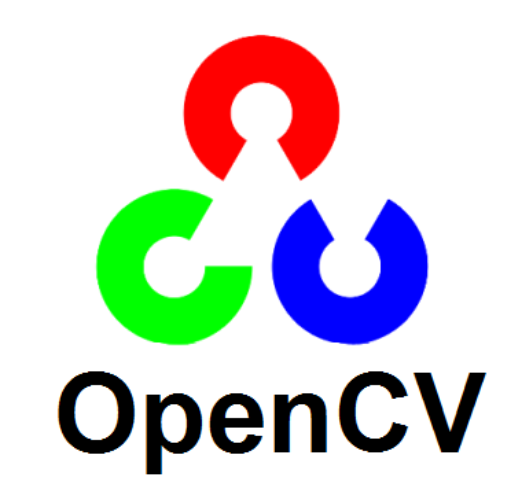
**3.1.1 Jupyter Notebook:**

Multiple programming languages are supported by the interactive computing environment known as Jupyter Notebook. It enables users to build documents with narrative text, live code, and visualisations. Users can independently execute code cells in the highly interactive environment to receive rapid feedback. The seamless integration of code and explanatory prose is made possible by notebooks, which makes them the best format for recording analyses and research. They provide rich outputs including tables, graphs, pictures, and interactive widgets, which improves the capabilities of data visualisation. Version control systems like Git make it simple to share, store, and manage notebooks in a variety of formats. Additionally, they provide plugins and extensions for capability expansion. Jupyter is an effective tool for data analysis and machine learning applications due to its compatibility with well-known data science packages. It allows for the use of LaTeX for mathematical notation.



**3.1.2 Open CV:**

An effective open-source library for computer vision and image processing is called OpenCV, or simply OpenCV. For tasks like image and video analysis, object detection, feature extraction, and more, it offers a wide variety of tools and capabilities. Python, C++, and Java are just a few of the many programming languages that OpenCV supports, making it available to a large developer community. For applications like face detection, object recognition, and motion tracking, it provides a comprehensive selection of pre-trained models and methods. Because of its reliability and effectiveness, OpenCV is a good choice for real-time applications like robots and augmented reality. It is extensively employed in industries including medical imaging, autonomous vehicles, and computer vision research. The community of OpenCV actively participates in its development, ensuring ongoing enhancements and upgrades.



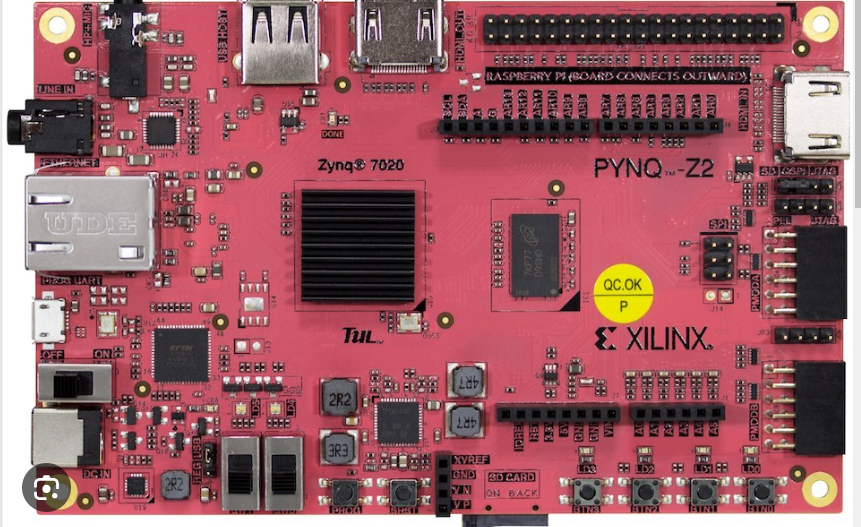
**3.2 COMPONENTS DESCRIPTION:**

* Pynq-Z2 Board
  + 1. **Pynq-Z2 Board**

BOARD PYNQ Z2 FEATURES: The PYNQ-Z2 board includes USB and Ethernet for internet connectivity, HDMI Input/Output, MIC Input, Audio Output, Arduino Interface, Raspberry Pi Interface, 2 Pmod, User LED, Pushbuttons, Switches, MicroSD Slot, Power In port for direct power connection, a Jumper for power source selection and another Jumper to select Boot Mode. Along with general purpose GPIO pins, it is meant to be easily extendable with Pmod, Arduino, and peripherals.

With the support of the driver's mobile device, this system uses an android smartphone to continuously update the location of the car. This item is referred to as a tracking device since it constantly checks its location and updates the monitoring device. An android app on a monitoring equipment gives the user access to the precise location of the vehicles he is interested in.

The PYNQ-Z2 board is a prime example of creativity in the embedded systems industry. The PYNQ-Z2 offers a flexible platform for performing complex algorithms since it combines an FPGA fabric and a high-performance Arm CPU.We offer a thorough investigation of a number plate identification system employing the PYNQ-Z2 board by utilising this special combination. The demand for precise and real-time number plate detection systems has increased due to the growing requirement for automated vehicle identification and traffic management.



**CHAPTER 4: CODING**

**Coding of Project is as follows:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from PIL import Image

import cv2

**Reading the image from the same directory code:**

img = cv2.imread('./car\_1.jpg')

plt.imshow(img)

**Image processing operation:**

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

noise\_removal = cv2.bilateralFilter(img\_gray,9,75,75)

equal\_histogram = cv2.equalizeHist(noise\_removal)

plt.imshow(equal\_histogram)

**Morphological operation and Thresholding:**

kernel = cv2.getStructuringElement(cv2.MORPH\_RECT,(15,5))

morph\_image = cv2.morphologyEx(equal\_histogram,cv2.MORPH\_OPEN,kernel,iterations=15)

sub\_morp\_image = cv2.subtract(equal\_histogram,morph\_image)

ret,thresh\_image = cv2.threshold(sub\_morp\_image,0,255,cv2.THRESH\_OTSU)

plt.imshow(thresh\_image)

**Edge Detection and Dilation of Image:**

canny\_image = cv2.Canny(thresh\_image,250,255)

canny\_image = cv2.convertScaleAbs(canny\_image)

kernel = np.ones((3,3), np.uint8)

dilated\_image = cv2.dilate(canny\_image,kernel,iterations=1)

plt.imshow(dilated\_image)

**Localizing License Plate using the Contouring Method:**

new,contours, hierarchy = cv2.findContours(dilated\_image, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

contours= sorted(contours, key = cv2.contourArea, reverse = True)[:10]

n = 230

print(len(contours))

for c in contours:

peri = 0.03\*cv2.arcLength(c, True)

approx = cv2.approxPolyDP(c, peri, True)

x,y,w,h = cv2.boundingRect(approx)

final = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

if len(approx) == 4:

screenCnt = approx

x1,y1,w1,h1 = cv2.boundingRect(screenCnt)

ratio = w/h

print(ratio)

crop = img[y1:y1+h1, x1:x1+w1]

n = n+1

plt.subplot(n),plt.imshow(crop)

**Showing the Localized Image - License Plate:**

plt.imshow(crop)

**Original image with the bounding boxes:**

plt.imshow(img)

**CHAPTER 5: ADVANTAGES**

1) PYNQ-Z2 leverages Xilinx Zynq SoC, combining ARM Cortex-A9 processors with programmable logic for high flexibility.

2) Python-based development environment and Jupyter notebook interface streamline coding and testing processes.

3) FPGA fabric accessibility allows for custom hardware accelerators and interfaces tailored to specific applications.

4) Low-cost entry point makes PYNQ-Z2 suitable for educational purposes, hobbyists, and small-scale projects.

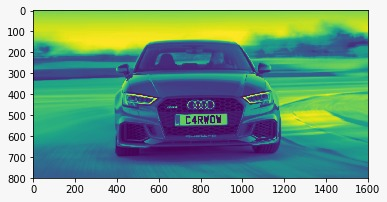
5) Strong open-source community support and a rich set of on-board peripherals enhance development capabilities.

**CHAPTER 6: RESULT & CONCLUSION**

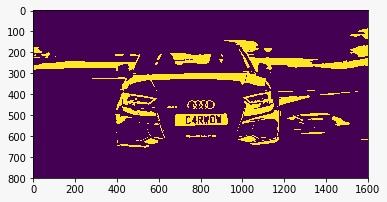
**6.1 Result:**



Original image of the car is taken.



Color conversion, Noise removal and Equalize Histogram Operation are done.



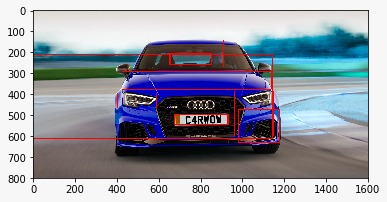
Morphological operation and Thresholding is done.



Localizing License Plate using the Contouring method.



Showing the localized image-License Plate.



Original image with Bounding Boxes.

**6.2 Conclusion:**

This project successfully demonstrated the development and implementation of an efficient and accurate number plate detection system using the PYNQ-Z2 board. This real-time capability positions the system as a valuable tool for scenarios requiring immediate response, such as toll collection and access control.In essence, the successful implementation of this number plate detection system using the PYNQ-Z2 board establishes a solid foundation for advancing smart transportation, security, and surveillance systems.

**CHAPTER 7: REFERENCES**

1. Attendance authentication system using face recognition Sabeenian, R.S., Harirajkumar, J., D’ Cruz, L.Journal of Advanced Research in Dynamical and Control Systems, 2020, 12(4 Special Issue), pp. 1235–1248
2. OpenCV, “OpenCV documentation,” <https://opencv.org/about.html>
3. Sahil Shaikh, Bornika Lahiri, Gopi Bhatt, Nirav Raja, A Novel Approach for Automatic Number Plate Recognition, International conference on Intelligent systems and Signal Processing, Gujarat, India, 2013, 375 – 380