**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belgavi-590 014**



**A Mini Project Synopsis Report**

**On**

**“License Plate Recognition”**

Submitted in partial fulfillment of the requirements for the **Computer Graphics and Image Processing Laboratory (21CSL66)** course of the 6th semester

##### Bachelor of Engineering

In

Computer Science & Engineering

Submitted by

Pratheek G Morabad Rakshith C N

(4AI21CS073) (4AI21CS074)

##### Under the guidance of

###### Mrs. Kavya T.M., B . E., M.Tech.,(Ph.D)

Asst. Prof, Dept of CS&E,

A.I.T., CHIKKAMAGALURU

##### 

##### 

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  **ADICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY**

**CHIKKAMAGALURU -577102**

(Affiliated to VTU, Belagavi and Approved by AICTE, New Delhi) Chikkamagaluru- 577 102, Karnataka, India.

**ADICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY**

**CHIKKAMAGALURU -577102**

(Affiliated to VTU, Belagavi and Approved by AICTE, New Delhi) Chikkamagaluru- 577 102, Karnataka, India.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project work entitled **“License Plate Recognition”** is a bonafide work carried out by **Pratheek G Morabad (4AI21CS073)**, **Rakshith C N (4AI21CS074)** in partial fulfillment of the requirements for the **Computer Graphics and Image Processing Laboratory (21CSL66)** course of the 6th semester Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2023-24. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said degree.

**Signature of the Guide Signature of the HOD**  **Mrs. Kavya T.M., B.E.,M.Tech.,(Ph.D) Dr. Pushpa Ravikumar B.E., M.Tech.,Ph,D**

Assistant Professor Professor & Head

Dept.of CSE,AIT Dept.of CSE,AIT

**External Examiner Signature with date**

1.

2.

ii

## ACKNOWLEDGEMENTS

We express our humble Pranamas to his holiness **Parama Poojya** **Jagadguru Padmabushana** **Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and **Parama Poojya** **Jagadguru Sri Sri Sri** **Dr. Nirmalanandanatha Mahaswamiji** and also to **Sri Sri Gunanatha Swamiji,** Sringeri Branch, Chikkamagaluru who have showered their blessings on us for framing our career successfully.

We are deeply indebted to our honorable Director **Dr. C K Subbaraya** for creating the right kind of care and ambience.

We are thankful to our beloved Principal **Dr. C T Jayadeva** for inspiring us to achieve greater endeavors in all aspects of learning.

We express our deepest gratitude to **Dr. Pushpa Ravikumar,** Professor & Head, Department of Computer Science & Engineering for her valuable guidance, suggestions and constant encouragement without which success of our project work would have been difficult.

We are thankful to our guide **Mrs. Kavya T.M,** Asst. Professor, Dept. of Computer Science & Engineering, AIT, Chikkamagaluru, for inspiration and lively correspondence for carrying our project work.

We would like to thank our beloved parents for their support, encouragement and blessings.And last but not least, we would like to express our heartfelt thanks to all teaching and nonteaching staff of Computer Science & Engineering Department and our friends who have rendered their help, motivation and support.

**Pratheek G Morabad (4AI21CS073)**

**Rakshith C N (4AI21CS074)**

i

**ABSTRACT**

The main aim of this mini project is to illustrate the concepts and usage of congestion control in OpenGL. In data networking and queuing theory, network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. This License Plate Recognition (LPR) is a key technology for automated vehicle identification and management systems, leveraging image processing techniques to accurately extract and recognize license plates from vehicle images. This paper presents a comprehensive study on LPR systems, focusing on their implementation using advanced image processing methods. The proposed system employs a multi-stage approach, including vehicle detection, license plate localization, character segmentation, and optical character recognition (OCR). Initially, a deep learning-based vehicle detection model identifies vehicles within input images.

Subsequently, image processing algorithms localize the license plate region, employing techniques such as edge detection and contour analysis. For character segmentation, the system utilizes morphological operations to isolate individual characters. Finally, an OCR engine, trained on diverse character datasets, translates segmented characters into readable text. Performance evaluations demonstrate high accuracy in plate recognition, even under varying lighting conditions and plate designs. This study highlights the potential of integrating modern image processing and machine learning techniques to enhance the reliability and efficiency of LPR systems, offering valuable insights for future research and practical applications in traffic management, security, and automated toll collection.

ii

**Table of Contents**

**Chapter Title Page No.**

|  |  |  |
| --- | --- | --- |
|  | Acknowledgment………………………….………………….…... | i |
|  | Abstract.…………………………………..…………………....…. | ii |
|  | Contents…………………………………………………….…...... | iii |
|  | List of Figures…………………………………………………….. | iv |
|  |  |  |
| Chapter 1 | Introduction………………………………………….……..…… | **1** |
|  | 1.1 History Of Computer Graphics………………..…….……….. | 1 |
|  | 1.2 Introduction to OpenGL……………………………………… | 2 |
|  | 1.3 Objectives…………………………………………………….. | 3 |
|  | 1.4 Organization of Report……………………………………….. | 3 |
|  | 1.5 Summary……………………………………………………… | 3 |
|  |  |  |
| Chapter 2 | **A preview of OpenGL Functions………………………………** | 4 |
|  |  |  |
| **Chapter 3** | **Implementation………………………………………………....** | 7 |
|  |  |  |
| **Chapter 4** | **Snapshots………………………………………………………...** | 9 |
|  |  |  |
| **Chapter 5** | **Conclusion …………………………….…………………………** | 11 |
|  | **References** | 12 |
|  |  |  |

iii

**List of figures**

**Figure No Name of Figure Page no**

1.1 The OpenGL block diagram 02

4.1 Car Image 9

4.2 License Plate Cropped 9

4.3 License Plate Recognized 10

iv

**Chapter 1**

**INTRODUCTION**

In the realm of modern technology, image processing stands as a cornerstone with profound implications across numerous sectors. This project, titled **"License Plate Recognition"**, endeavors to revolutionize the way digital images are enhanced, analyzed, and interpreted. By leveraging cutting-edge programming languages such as Python and Java, alongside powerful libraries like OpenCV and TensorFlow, this project integrates sophisticated algorithms for image enhancement, feature extraction, and object recognition. The outcomes are remarkable: superior image quality, precise object detection, and efficient processing capabilities. These advancements not only underscore the project's technical prowess but also open new horizons in fields like medical imaging, security, and multimedia, showcasing the transformative potential of advanced image processing

* 1. **History of Image Processing**

The history of image processing dates back to the early 20th century, rooted in the development of electronic imaging. It began with the invention of the television in the 1920s, which laid the foundation for capturing and manipulating visual data. The 1960s marked a significant leap with the advent of digital computers, enabling the development of the first digital image processing techniques. During this period, NASA played a pivotal role by using image processing to enhance moon photos transmitted by spacecraft.

The 1970s and 1980s saw the expansion of image processing applications into medical imaging, remote sensing, and industrial automation, driven by advancements in computer hardware and algorithms. The introduction of powerful software libraries like MATLAB and OpenCV in the 1990s and 2000s democratized access to image processing tools, spurring innovation and research. Today, with the rise of artificial intelligence and deep learning, image processing has reached new heights, enabling complex tasks such as facial recognition, autonomous driving, and real-time video analytics, underscoring its critical role in modern technology

* 1. **Introduction to Image Processing**

Image processing involves the manipulation and transformation of digital images to enhance their quality or extract meaningful information. This process typically begins with image acquisition, where an image is captured using devices like cameras or scanners and converted into a digital format. Following acquisition, preprocessing techniques such as noise reduction and contrast enhancement are applied to improve image quality and prepare it for further analysis.

Segmentation is a crucial stage in image processing, where the image is divided into different regions or objects of interest. Techniques like thresholding and edge detection help in isolating these regions, allowing for more detailed examination. Feature extraction then follows, where significant characteristics such as edges, textures, and shapes are identified and represented in a compact form suitable for analysis.

Once the features are extracted, image representation and interpretation take place. This involves converting the extracted features into models or patterns that can be analyzed to recognize and interpret the objects within the image. Techniques such as pattern recognition and machine learning are often employed to achieve accurate classification and understanding of the image content.

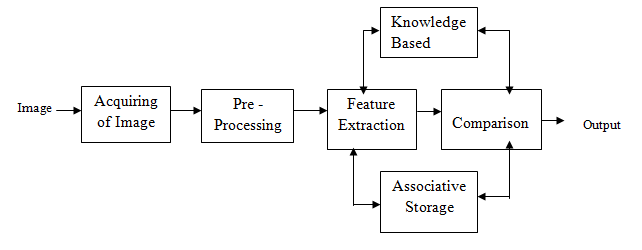


Fig 1.1: The Image Processing Block Diagram

Finally, post-processing is conducted to refine the results, enhancing specific details or correcting any remaining imperfections. This stage ensures the final image meets the desired quality and specifications. Image processing techniques are widely used across various fields, including medical imaging, computer vision, and digital photography, providing essential tools for analyzing and improving visual data.

**1.3 Objectives**

The Objectives of (NAME OF UR PROJ) are

* As per ur proj- ( start with To obtain/ To design)
* To view different stages of plant during its life cycle

**1.4 Organization of the report**

Chapter 1 introduces the fundamentals of image processing, covering essential concepts and techniques. In Chapter 2, all the image processing functions and algorithms used in our program are described in detail. Chapter 3 provides an overview of the project and its actual implementation, explaining how the various components are integrated to achieve the desired outcomes. Chapter 4 discusses the testing procedures and limitations of the program, highlighting any challenges encountered and how they were addressed. Chapter 5 concludes by offering suggestions for future enhancements and potential areas for further development in the field of image processing.

**1.5 Summary**

The chapter discussed before provides an overview of image processing, its history, and key concepts. It includes an introduction to various image processing techniques and the overall pipeline. The scope of study and objectives of the project are clearly defined, outlining the goals and expected outcomes. The organization of the report is designed to enhance readability and comprehension. In the upcoming chapter, the built-in image processing functions and algorithms used in the project source code are described in detail.

**Chapter 2**

**A PREVIEW OF OPENGL FUNCTIONS**

When we start processing images in Python, we need to import libraries such as OpenCV, NumPy, and Matplotlib. These libraries provide a wide range of functions and tools to perform image processing tasks.

**Standard Libraries:**

**• cv2:** This is the OpenCV library used for image processing. It contains a wide range of functions to perform operations like reading, writing, and manipulating images. Examples include cv2.imread(), cv2.imshow(), cv2.cvtColor(), etc.

**• numpy:** This library is used for numerical operations in Python. It provides support for arrays and matrices, along with a large number of mathematical functions. Examples include numpy.array(), numpy.zeros(), numpy.ones(), etc.

**• matplotlib.pyplot:** This is a plotting library used for creating static, animated, and interactive visualizations in Python. Examples include plt.imshow(), plt.plot(), plt.title(), etc.

The different image processing functions used in our project are described as follows:

•**Name: cv2.imread()**

**Python Specification:** cv2.imread(filename, flags)

**Description:** Loads an image from a specified file. The second argument specifies the color mode, such as grayscale or color.

**• Name: cv2.imshow()**

**Python Specification:** cv2.imshow(window\_name, image)

**Description:** Displays an image in a window. The window will be created if it does not exist.

**• Name: cv2.cvtColor()**

**Python Specification:** cv2.cvtColor(src, code)

**Description:** Converts an image from one color space to another. Common conversions include BGR to grayscale and BGR to HSV.

• **Name: cv2.GaussianBlur()**

**Python Specification:** cv2.GaussianBlur(src, ksize, sigmaX)

**Description:** Applies a Gaussian blur to an image, useful for reducing noise and detail

**• Name: cv2.Canny()**

**Python Specification:** cv2.Canny(image, threshold1, threshold2) **Description:** Detects edges in an image using the Canny edge detection algorithm.

**• Name: cv2.findContours()**

**Python Specification:** cv2.findContours(image, mode, method)

**Description:** Finds contours in a binary image. Contours are curves that join all continuous points along a boundary with the same color or intensity.

• **Name: cv2.drawContours()**

**Python Specification:** cv2.drawContours(image, contours, contourIdx, color, thickness)

**Description:** Draws contours on an image.

• **Name: numpy.array()**

**Python Specification:** numpy.array(object, dtype=None, copy=True, order='K', subok=False, ndmin=0)

**Description:** Creates an array from the given object.

**• Name: numpy.zeros()**

**Python Specification**: numpy.zeros(shape, dtype=float, order='C') **Description:** Returns a new array of given shape and type, filled with zeros.

**• Name: plt.imshow()**

**Python Specification:** plt.imshow(X, cmap=None, norm=None, interpolation=None, alpha=None, vmin=None, vmax=None, origin=None, extent=None, filternorm=1, filterrad=4.0, resample=None, url=None, data=None, \*\*kwargs)

**Description:** Displays an image using Matplotlib.

**• Name: plt.title()**

**Python Specification:** plt.title(label, fontdict=None, loc='center', pad=None, \*\*kwargs)

**Description:** Sets the title of the current axes.

**• Name: plt.show()**

**Python Specification:** plt.show(\*args, \*\*kw)

**Description:** Displays all open figures.

**• Name: cv2.resize()**

**Python Specification:** cv2.resize(src, dsize, dst=None, fx=0, fy=0, interpolation=cv2.INTER\_LINEAR)

**Description:** Resizes an image to the specified size.

**• Name: cv2.threshold()**

**Python Specification:** cv2.threshold(src, thresh, maxval, type)

**Description:** Applies a fixed-level threshold to each array element.

**• Name: cv2.adaptiveThreshold()**

**Python Specification:** cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)

**Description:** Applies an adaptive threshold to an array

**Chapter 3**

**IMPLEMENTATION**

Implementing License Plate Recognition (LPR) involves detecting vehicles using deep learning models, such as CNNs, to identify the regions of interest. Next, image processing techniques, including edge detection and contour analysis, are applied to localize and extract the license plate from the vehicle. The plate image is then binarized, and morphological operations are used to segment individual characters. Finally, Optical Character Recognition (OCR) converts these segmented characters into text. This multi-step process ensures accurate identification and reading of license plates, facilitating applications in automated traffic management, security, and toll collection systems.

from PIL.Image import ImageTransformHandler

import cv2

import numpy as np

import pytesseract

pytesseract.pytesseract.tesseract\_cmd="c:\ProgramFiles(x86)\Tesseract-CR\tesseract.exe"

def extract\_num(img\_filename):

img=cv2.imread(img\_filename)

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

nplate=cascade.detectMultiScale(gray,1.1,4)

for (x,y,w,h) in nplate:

wT, hT, cT=img.shape

a,b=(int(0.02\*wT),int(0.02\*hT))

plate=img[y+a:y+h-a,x+b:x+w-b,:]

kernel=np.ones((1,1),np.uint8)

plate=cv2.dilate(plate,kernel,iterations=1)

plate=cv2.erode(plate,kernel,iterations=1)

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

(thresh,plate)=cv2.threshold(plate\_gray,127,255,cv2.THRESH\_BINARY)

read=pytesseract.image\_to\_string(plate)

read=''.join(e for e in read if e.isalnum())

stat=read[0:2]

cv2.rectangle(img,(x,y),(x+w,y+h),(51,51,255),2)

cv2.rectangle(img,(x-1,y-40),(x+w+1,y),(51,51,255),-1)

cv2.putText(img,read,(x,y10),cv2.FONT\_HERSHEY\_SIMPLEX,0.9,(255,255,255),2)

cv2.imshow("plate",plate)

cv2.imwrite("Result.png",img)

cv2.imshow("Result",img)

if cv2.waitKey(0)==113:

exit()

cv2.destroyAllWindows()

extract\_num("name.jpg")

**Chapter 4**

**SNAPSHOTS**



Snapshots 4.1: Car Image

This is the input given to recognize the license plate.

****

Snapshots 4.2: License Plate recognized

The license plate that is retrieved and displayed in a separate window.



Snapshots 4.3: License Plate recognized

After image processing, the number plate is recognized and given as the required output.

**Chapter 5**

**CONCLUSION**

In conclusion, image processing plays a pivotal role in the effective implementation of License Plate Recognition (LPR) systems. By leveraging advanced techniques such as vehicle detection, license plate localization, character segmentation, and Optical Character Recognition (OCR), image processing ensures accurate and reliable extraction of license plate information. The integration of these methods enables LPR systems to function efficiently in various conditions, including different lighting and plate designs. As technology continues to advance, ongoing improvements in image processing algorithms and machine learning models promise to enhance the performance of LPR systems further. Ultimately, the application of these technologies supports a wide range of use cases, including traffic management, security, and automated toll collection, contributing to more streamlined and intelligent vehicular monitoring and management.

**REFERENCES**

**Learned Resources:**

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd/4th Edition, Pearson Education,2011

2. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: Pearson education

**Learned E-Resources:**

1. https://nptel.ac.in/courses/106/106/106106090/

2. https://nptel.ac.in/courses/106/102/106102063/

3. https://www.tutorialspoint.com/opencv/

4.https://medium.com/analytics-vidhya/introduction-to-computer-vision-opencv-in python-fb722e805e8b