WEB APPLICATION BASED ON OPTICAL CHARACTER RECOGNITION

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Falguni Bharadwaj (191034)

Sagrika Gargya (191002)

Vaishnavi Singh (191041)

UNDER THE GUIDANCE OF

Dr. Nafis Uddin Khan

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2023

TABLE OF CONTENTS

CAPTION **PAGE NO.**

DECLARATION

We hereby declare that the work reported in the B.Tech Project Report entitled **“Web Application Based on Optical Character Recognition”** submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of **Dr. Nafis Uddin Khan.** We have not submitted this work elsewhere for any other degree or diploma.

Falguni Bharadwaj	Sagrika Gargya	Vaishnavi Singh
191034	191002	191041

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

08/05/2023

Dr. Nafis Uddin Khan

Date: 05-05-2023

Project Coordinator

ACKNOWLEDGEMENT

Long-term efforts can be successful with the advice and support of a few wise people us. We would like to take this opportunity to express our gratitude to everyone who has a lend us a hand during this project. We owe thanks to Dr. Rajiv Kumar, Head of the Department of Electronics and Communication at JUIT, for his valuable support and guidance during the implementation of the project. We would like to express our sincere gratitude to the project managers, Dr. Shweta Pandit and Dr. Emjee Puthooran, professors of electronics and communication engineering at JUIT, who provided simulated discussions, analysis of issues related to project work, and a guide to the entire project. The project meeting was very informative. Also, we extend our gratitude to our project supervisor, Dr. Nafis Uddin Khan, for his encouragement, tireless support, and trust in us. We express our gratitude for their valuable guidance throughout the project. We would also like to thank all the staff in the ECE department for their valuable advice. We would also like to thank our support staff for providing resources as needed.

LIST OF ABBREVIATIONS

1. OCR – OPTICAL CHARACTER RECOGNITION
2. ANPR/ALPR – AUTOMATIC NUMBER/LISCENCE PLATE RECOGNITION
3. SLR – SYSTEMATIC LITERATURE REVIEW
4. OPENCV – OPEN-SOURCE COMPUTER VISION LIBRARY
5. OS – OPERATING SYSTEM
6. RAM – RANDOM ACCESS MEMORY
7. LOG – LAPLACIAN OF GAUSSIAN
8. DOG – DIFFERENCE OF GAUSSIANS
9. DOH – DETERMINANT OF HESSIAN
10. OSL – OPEN-SOURCE LICENSE
11. PDF – PORTABLE DOCUMENT FORMAT
12. RGB – RED GREEN BLUE
13. PYPI – PYTHON PACKAGE INDEX
14. DNA – DEOXYRIBONUCLEIC ACID
15. GPU – GRAPHICS PROCESSING UNIT
16. CUDA – COMPUTE UNIFIED DEVICE ARCHITECTURE FOR
17. PEP – POST EXPOSURE PROPHYLAXIS
18. OSI – OPEN SYSTEM INTERCONNECTION
19. API – APPLICATION PROGRAMMING INTERFACE
20. OPENCL – OPEN COMPUTING LANGUAGE

21. SCIPY – SCIENTIFIC PYTHON

22. GUI-GRAPHICAL USER INTERFACE

23. MFC – MICROSOFT FOUNDATION CLASS

24. OOP – OBJECT ORIENTED PROGRAMMING

LIST OF FIGURES

ABSTRACT

The science of image processing known as optical character recognition allows us to convert different kinds of documents or photos into editable, searchable, and analyzable data. The need to save information from printed or handwritten documents or photographs to a computer disc is great since it allows for later use of the material for different purposes. It might be possible to scan and store the printed documents as picture files, which would make it simple to store the information from them on a computer system. To read or query text or other information from these image files and use it again, though, could be extremely difficult. Therefore, a method to automatically retrieve and preserve data, especially text, from image files is needed. Optical character recognition is a branch of research aimed at developing a computer system capable of automatically extracting and analyzing text from images. Any type of text or text-containing document can be identified, modified, or converted using optical character recognition (OCR) into an editable digital format for further and more in-depth processing. OCR enables a machine to automatically recognize text in such documents. Optical character recognition facilitates and covers the area of automatic number plate recognition. It is a security system that employs concepts from image processing. Numerous tasks are performed by the number plate recognition system, including allowing vehicles to enter toll roads by automatically identifying their number plate and issuing a pay slip before opening the road for that particular vehicle. The parking authority also employs this method to allow a car to park in their spot. This technique begins by taking a picture of the license plate, which is then processed to determine the plate's identity and read each character for precise recognition. The most important stage is OCR, when the number plate image's letterings are converted into texts that can be decoded later. A complete working model and network flow for optical character recognition and automatic number plate recognition are shown in this project report. The ANPR system's concept is based on the localization of plates, and its accuracy (result) is set at 75–85%.

CHAPTER – 1
INTRODUCTION

We compel the development of machines that can discern patterns out of habit and natural behavior. All of the following topics are of interest: automated optical character recognition, face and speech recognition, fingerprint and DNA sequence identification, and many others. It should be obvious that precise and trustworthy pattern identification would greatly aid machine learning. Optical character recognition is a study area that is being actively pursued in the quest to develop a computer system that can automatically extract and interpret text from photos. Nowadays, there is a huge demand to store data from the available sources in printed or handwritten form to a computer storage disc so that it can be used again. Therefore, a technique to automatically extract and save data, particularly text, from image files is needed. But this is not an easy piece of work. However, if accomplished, it can be very beneficial to practically all fields of research in image processing.

1.1 PROBLEM STATEMENT

The objective is to employ optical character recognition (OCR) and experiment with it to obtain the best results. In order to identify license plates automatically, we first develop a mechanism for doing so. We will increase the system's accuracy and expand its use to real-time operations and more Optical Character Recognition fields by doing extensive research and experimentation.

1.2 MOTIVATION

We ought to work on an optical character recognition system, and especially an automatic number plate identification system, given its wide applicability and research. Multiple benefits exist with such a system. It has numerous uses in the corporate, academic, security, automated, medical, and other industries. It simplifies the task of storing vast amounts of data. Data access is made easier by it. Data can now be updated and searched.

CHAPTER – 2
LITERATURE REVIEW

In order to turn printed text into editable text [1], this project used optical character recognition (OCR) technology. In the first half of this work, the history, architecture, and experiment results using various image kinds and the free OCR program Tesseract are all explained. Using data from vehicle license plates, this study contrasts this technology with Transym OCR, another commercial OCR program. Tesseract and Transym have been used to attempt to deduce the car's number from the number plate. A number of factors have been used to compare the two programs.

This paper reviews prior research on character identification [2] in handwritten manuscripts and makes suggestions for further research. All research publications published between the years 2000 and 2019 that address handwritten OCR (and issues that are closely related) have been gathered, collated, and reviewed by the authors of this Systematic Literature Review (SLR). Frequently used electronic databases were used in accordance with a predetermined review methodology. Keywords, forward reference searching, and backward reference searching were all used to find every article on the topic.

This paper reviews prior research on recognizing characters [3] in handwritten manuscripts and makes recommendations for future research. The research publications on handwritten OCR (and topics that are closely related) that have been published between the years 2000 and 2019 have all been gathered, collated, and reviewed by the authors of this Systematic Literature Review (SLR). Frequently used electronic databases have been used in accordance with a pre-established review methodology. Keywords, forward reference searching, and backward reference searching were all used to find every article on the subject.

Utilizing salient features like shape, texture, and color, this study [4] uses a feature prominent method to obtain the car number plate. In order to identify vertical and horizontal lines (HIS), the authors utilized the Hough transform (HT) to examine RGB images of rectangular car number plates. Splitting may be seen at the license plates' terminus. On a Pentium-IV 2.26 GHz system with 1 GB of RAM, this method is carried out in MATLAB.

In this study [5], the identification of a vehicle's license plate is accomplished by pre-processing, segmentation, and verification. Vertical edges are mapped in the pre-processing step using the global Thresholding. According to the authors' two assumptions, which are that the plates are horizontally oriented, the foreground of the characters and the background of the plates have incredibly different intensities. The rank-filter has MxN pixels that are horizontally orientated, and the image is placed next to it. Later, the plate's vertical edges are found by using vertical projection. Direc's delta function and the Random Transform (RT) function are both used to adjust the skew location of a number plate. A succession of morphological erosions with structural components aligned horizontally are used to find the limits of the horizontal. The last step involves a two-stage verification process.

This work makes use of the fuzzy-based method [6]. Four approaches are used to identify the region of the license plate. The input image's noise is removed in the first stage. The following phase of locating a likely zone makes use of edge detection. The final stage involves deleting any erroneous rectangle segments based on the size, histogram, and other data. In the last step, the candidate region for a license plate is obtained using geometric rectification. The authors employed a fuzzy-based strategy with numerous phases to more precisely extract licence plates because these procedures require some additional processing.

According to the vehicle's license plate and the length of time it has been parked in the lot, the suggested proposal [7] recommends an automated system for collecting parking tolls. In order to build an Automatic Number Plate Recognition System (ANPR), this work addresses a challenge from the fields of artificial intelligence, computer vision (image processing), and neural networks. This problem uses mathematical concepts and algorithms, which assures that there are many different ways to carry out the stages for the product. Any camera that is capable of taking high-quality pictures is used to acquire the image. In this article, the focus is on the localization of the license plate using the contours tracing approach, as well as edge detection and edge sharpening using the Canny edge detection algorithm.

Using footage processing and OCR technology, this system [8] recognizes the image of a vehicle's license plate from video, and various techniques are used to retrieve the number. The technology is suitable at college entrances and other places with rigorous security. Whenever a vehicle traverses by the system, video is recorded, converted into images, then text, and then stored in the database.

This paper [9] provides a complete evaluation of current methodology and advancements in Automatic-Number-Plate-Recognition (ANPR) systems through a performance comparison of several real-time tested and simulated algorithms, including those using computer vision (CV). An excellent way to employ ANPR is to combine it with GPS, Android operating systems, RFID systems, and other analogous technologies. In the CV industry, deep learning techniques are frequently employed for greater detection rates. This research aims to increase the state of knowledge in ITS (ANPR) built on CV algorithms by highlighting significant prior work, reviewing and presenting an overview of extraction, segmentation, and recognition approaches, and making recommendations on potential future trends in this area.

CHAPTER – 3
PROPOSED WORK AND LIBRARIES USED

3.1 SYSTEM PROPOSED

1. The system can use a picture as an input and output the extracted text as an overlay on the image.
2. It applies techniques for pre-processing to the input image, such as filtering, sharpening, masking, grey scalability, and conversion to RGB, among others.
3. It uses the car number plate input image as input, locates the plate's position, and outputs the plate number.

3.2 OPENCV

1. The creation of cross-platform, real-time computer vision applications is made possible by this library.
2. The major focus areas of tools for object and face detection are image processing, video capture, and analysis.
3. Open CV is currently accessible on a wide number of operating systems, including Windows, Linux, OS X, Android, and iOS, and it supports a wide range of programming languages, including C++, Python, and Java.
4. For high-speed GPU tasks, interfaces based on CUDA and OpenCL are also being developed. The Open CV Python API is known as Open CV-Python.
5. It combines the best features of Python and the Open CV C++ API.

3.3 NUMPY

Large, multi-dimensional arrays and matrices as well as a considerable number of high-level mathematical operations are supported by the NumPy library for the Python programming language. Despite not being designed with numerical computing in mind, the scientific and engineering community quickly became interested in the Python programming language. In 1995, a special group of enthusiasts called matrix-sig was established in order to give an array computing environment. Python creator and maintainer Guido van Rossum was one of its contributors. To make array computing simpler, he enhanced the language's grammar, particularly the syntax for indexing.

3.4 MATPLOT

Using a general-purpose GUI toolkit like Tkinter, WXPython, Qt, or GTK +, diagrams can be incorporated into our applications. Matplot is a plot package for the Python programming language and its NumPy numerical math extension. In SciPy, Matplotlib is utilised. John D. Hunter's Matplotlib software has a thriving developer community and a BSD-like licencing structure. Before John Hunter passed away in August 2012, Thomas Caswell and Michael Droettboom shared management of the matplotlib project.

3.5 PYTORCH

Python and the Torch library are the foundation of the open-source PyTorch machine learning (ML) platform. Deep neural networks are constructed using Torch, an open-source machine learning tool based on the Lua programming language. One of the most frequently used deep learning research platforms is this one. A faster transition from research prototyping to implementation is what the framework intends to achieve. The PyTorch framework is capable of supporting more than 200 distinct kinds of mathematical operations. Because building models for artificial neural networks is so much easier with PyTorch, its use is still expanding. PyTorch is used most frequently by data scientists who work on artificial intelligence (AI) research and applications. PyTorch is released under a BSD licence that has been modified.

3.6 EASYOCR

Users can execute optical character recognition on projects using a library called EasyOCR. OCR analysis converts the digital picture input, which might be printed or handwritten, into a machine-readable digital text format. Once the character blocks have been broken down into their component parts and compared to a character dictionary for the analysis of locating text, words, or character blocks, the character blocks are once again dissected into their component parts. This is done after using OCR to break the digital image into smaller bits. Python is a computer language that offers a setting in which this issue can be resolved. Even though we have a sizable library, we can import it and use it for OCR tasks. The design of algorithms, computing, analysis, and other tasks are only a few of the many uses for Python. This offers an immediate fix to our issue and a clear remedy.

3.7 Python

1. Simple to grasp Python features an easy-to-understand structure, clear syntax, and few excessive keywords. Students are able to pick up the language more quickly as a result.

2. The syntax used in Python programming is clear and easy to understand.

3. It is simple to maintain Python source code.

4. Common library procedures UNIX, Windows, and Macintosh platforms can all execute Python's general-purpose library, making it very portable.

5. Fourth Relationship Mode For interaction testing and subtitle mistake correction, Python has interaction modes.

6. Portable On all computer platforms, Python uses the same user interface.

7. The Python interpreter can take simple, extensible modules. Installing or altering these components will allow system designers to improve the effectiveness of their tools.

8. All necessary business information is available in the Python Ways of Meeting.

9. User interfaces for applications like Windows MFC, Macintosh MFC, and Unix X Window are easier to design and install when using GUI Python programming.

10. Python build and support will be useful for big, scalable projects, even though it is not necessary for the shell description.

3.8 Image Processing approaches:

(A) Image Filtering

The different types of filters in image processing are:

(i) Median Filter

(ii) Mean Filter

(iii) Gaussian Filter

(iv) Gaussian Blur Filter

(v) Box Filter

(B) Image Binarization

(C) Image Sharpening

(D) Masking

(E) Edge Detection

The different filters/techniques used for edge detection are:

(i) Prewitt Operator

(ii) Sobel Operator

(iii) Laplacian Operator

(iv) Canny Edge Detection

(v) Canny-Dirichlet Operator

(vi) Roberts Cross Operator

(F) Contour Detection

CHAPTER – 4
METHODOLOGY

4.1 Working of Optical Character Recognition

This problem is difficult to resolve because there are numerous font versions and character writing techniques. The image must be preprocessed prior to choosing an OCR approach so that the image can be read.

4.1.1 Pre-processing

OCR software often pre-process images to boost the chances of recognition.

Techniques include:

1. De-skewing

2. De-speckle

3. Binarization

4. Line removal

5. Layout analysis or zoning

6. Line and word detection

7. Script recognition

8. Character isolation or segmentation

9.Normalization

4.1.2 Feature Extraction

For extracting features in OCR, there are primarily two methods:

1. The feature detection algorithm used in the first method analyses the lines and strokes of a character to recognize it.
2. Pattern recognition employs the complete character to determine how it functions in the second method.

The search for white pixel rows with black pixels in between can be used to identify a line of text. A character’s beginning and end can be distinguished in a similar manner.

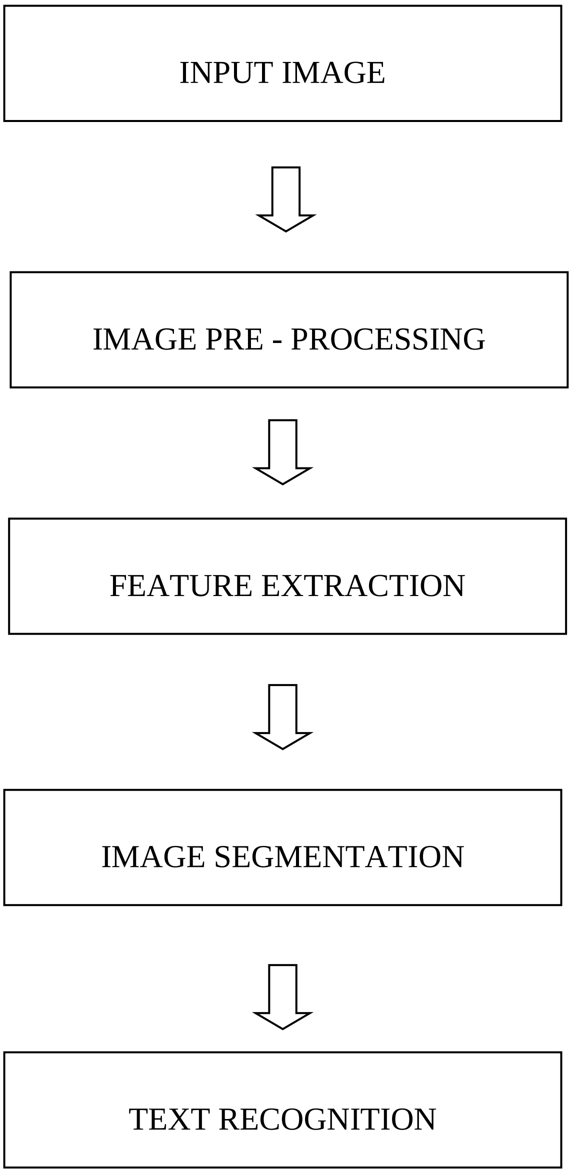
4.1.3 Post processing

By limiting the output (by employing a list of terms allowed in a document) with a vocabulary, OCR accuracy can be increased. A highly specialized lexicon for a certain industry, for example, or a glossary of all English-language expressions are two examples of what this could be. This method may not work as well if the text contains proper nouns or other non-lexical words. The good news is that accuracy can be improved by using free online OCR libraries. Character segmentation is handled by the Tesseract library using its dictionary. One illustration of how more sophisticated OCR algorithms preserve the original page structure is a PDF that includes both the original picture pages and a searchable text image. The output stream could be one character or one string.

4.1.4 Error correction

For error correction, techniques like near neighbor analysis can be used.

4.1.5 Flowchart



4.2 Building the Automatic License Plate Recognition System

Based on different techniques, the majority of number plate identification algorithms can be divided into many classes. The following components must be taken into account in order identify a vehicle number plate:

(1) Plate size: A plate in a picture of a car could be of different sizes.

(2) Plate location: A plate could be positioned anywhere inside the vehicle.

(3) Depending on the type of vehicle, the color of the plate backdrop can change. For example, the backdrop of a government vehicle's number plate may differ from that of other public vehicles.

(4) Screw: A plate screw can be seen of as a character.

4.2.1 Methodological Process

- **(A) Image Acquisition** – It is obtaining an image to be given as input to the system. It can be an acquired dataset of images or a self-made real collection of images.
- **(B) Image Reading** – The original image to be given as input is read using OpenCV into python. It is done for the visualization of the original image for further process and comparison.
- **(C) Image Preprocessing** – It involves various steps to modify and format the original image so as to make it fit to be worked upon. The different techniques for preprocessing the images are:

(i) Image binarization – Image binarization is the process of converting images to black and white. In this process, a certain threshold is used to classify certain pixels as white and certain pixels as black. The main problem, however, is how to select a suitable threshold value for a particular image. Automatic thresholding is the process of choosing a threshold using an algorithm instead of by hand.

(iii) **Image Filtering** – Image filtering is the process of applying various filters to an image for the purpose of removal of noise, blurring of an image, sharpening of an image, or simply enhancing it for further processing. The filter used in the project is the Laplacian filter used for sharpening of image which makes it easier to detect edges.

(D) Edge detection – Edge detection is the primary method for feature extraction or feature detection. The output of edge detection methods often looks like an object border with connected curves. Since this method may result in boundaries of objects with disconnected curves, using it on complex photos becomes quite difficult. Canny, Canny-Dirichlet, Differential, Sobel, Prewitt, and Roberts Cross are just a few of the edge detection algorithms and operators used.

(E) Blob Detection / Contour Detection – Blob detection is used to locate regions or spots that contrast significantly from their surroundings in terms of brightness or color. This method's main objective is to identify complimentary regions that are missed by corner or edge detection techniques. In addition to maximally stable extremal regions, Difference of Gaussians (DoG), Determinant of Hessian (DoH), Laplacian of Gaussian (LoG), and Principal Curvature-Based Region Detector are often used blob detectors.

(F) Segmentation – Image segmentation techniques can be used to retrieve a license plate. There are several different image segmentation techniques listed in diverse books. The majority of techniques employ picture binarization to turn color photos into grayscale ones. Segmentation can be:

(i) Pixel Based Segmentation
(ii) Region Based Segmentation
(iii) Threshold Based Segmentation
(iv) Cluster Based Segmentation
(v) Watershed Segmentation

4.2.2 WORKING WITH PYTHON IN OPENCV – JUPYTER NOTEBOOK

1. Installing and importing dependencies – Pytorch.
2. Reading the image in OpenCV.
3. Gray scaling and smoothing the image.
4. Applying the filter for sharpening.
5. Finding edges for localization.
6. Reading the text using EasyOCR.
7. Rendering the result – as text overlayed on the image.

4.2.3 Flow Diagram

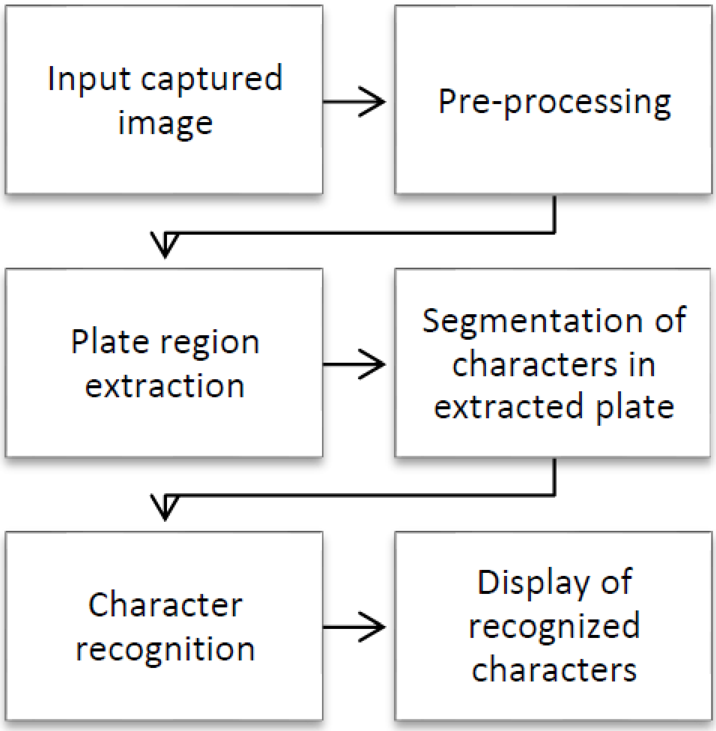


Figure 1: Flow Diagram for ANPR

CHAPTER – 5
EXPERIMENTAL ANALYSIS

5.1 CODE IMPLEMENTATION: IMAGE AS INPUT

Part 1 – OCR

1. We first install and import our libraries and dependencies
2. Next, we read our image in OpenCV using Python
3. Then we read the text in the image and highlight it with a bounding rectangle.
4. And, finally overlay the extracted text over the image.

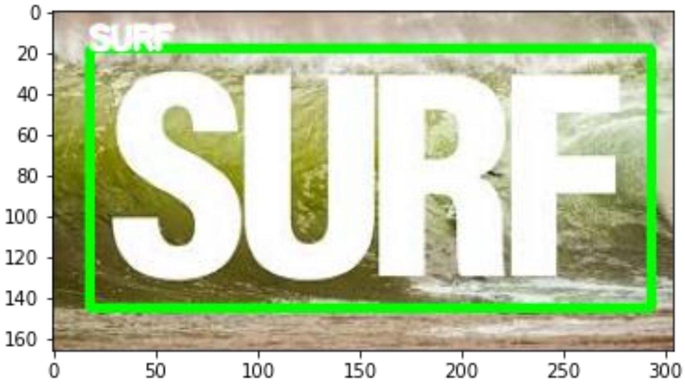


Figure 2: Single word

Handling Multiple Lines

1. To handle more than one lines we use the coordinates of the bounding rectangle wisely.
2. Then we establish a tuple to store the words.

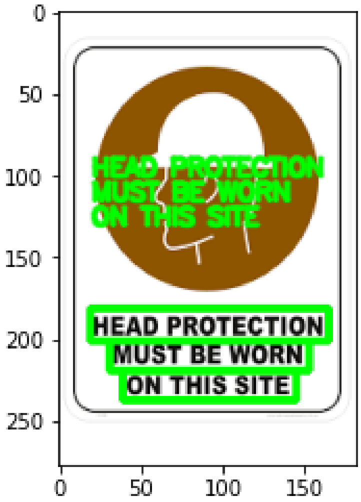


Figure 3: Multiple Lines

Part 2 – ANPR

1. First, we read the image in OpenCV using python.
2. Then we grayscale the RGB image to make it fit for pre-processing.

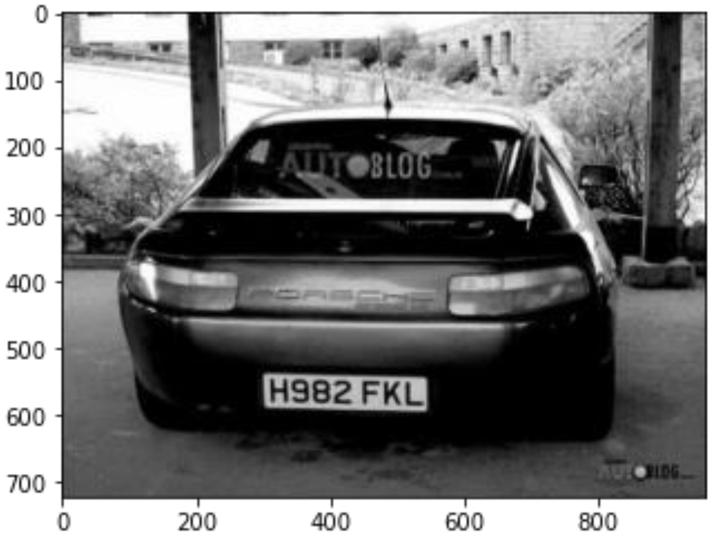


Figure 4: Gray scale Image

3. Next, we apply pre-processing techniques depending upon our need, to make the image clearer and fit for detection of edges.

Sharpening Kernal

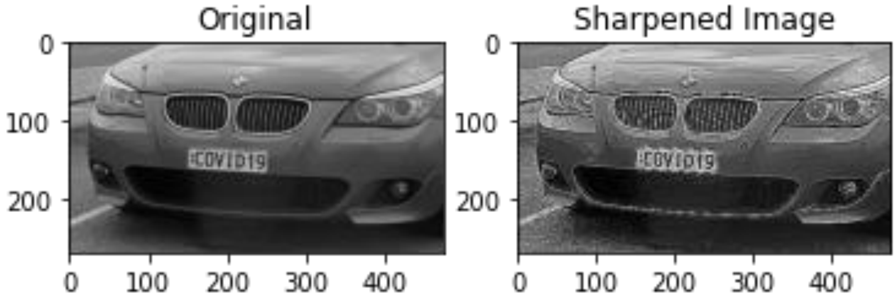


Figure 5: Original Vs Sharpened Images

4. Now, we perform edge detection and contour detection which will aid us in localization of number plate in the image.

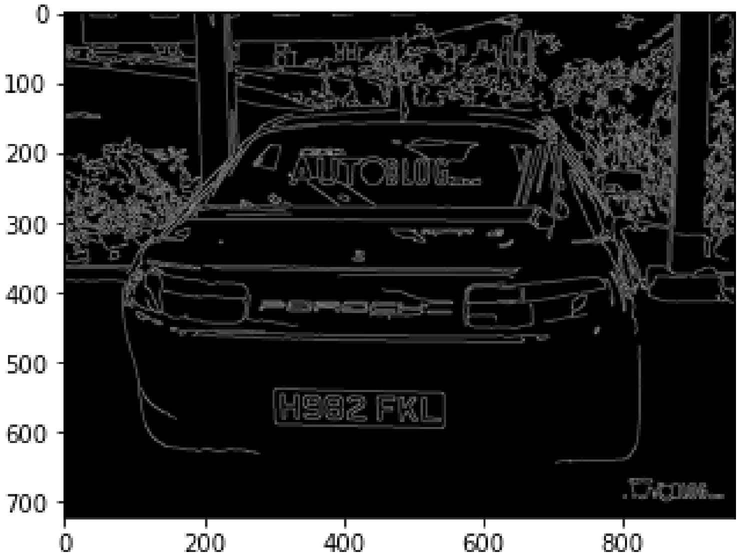


Figure 6: Contour Extraction

5. Next, we apply a mask using the complementary AND operator to isolate the plate area in the image.

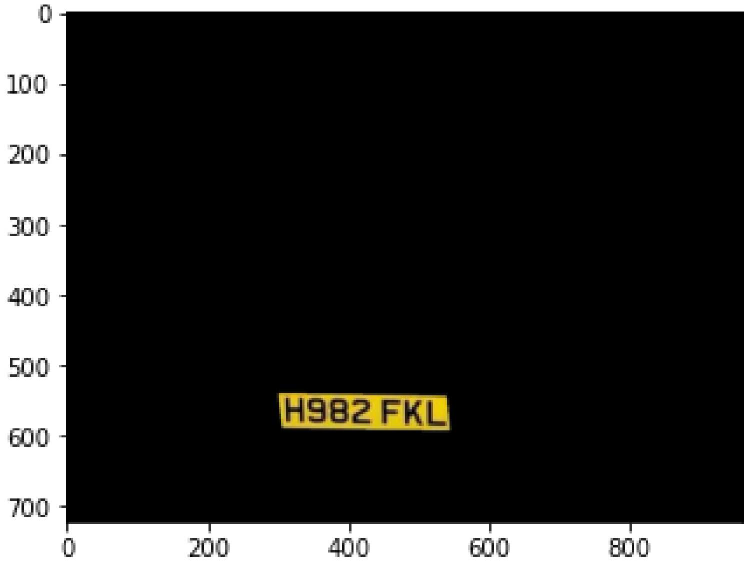


Figure 7: Applying Mask

6. Finally, we extract the text on the number plate using the EasyOCR library and display it as overlayed on the image.

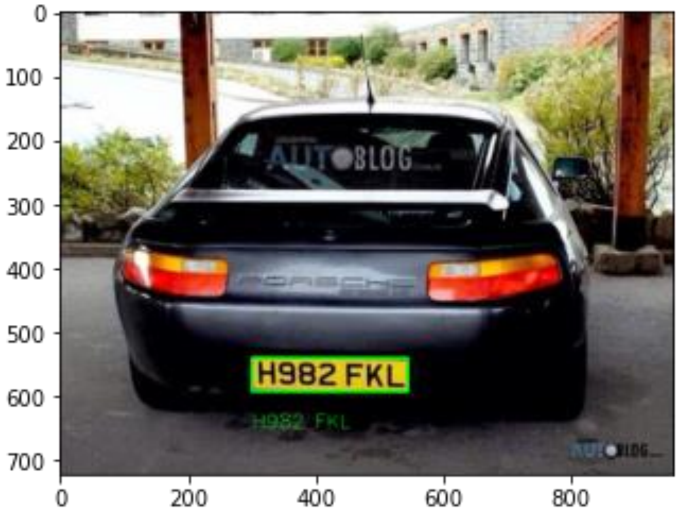


Figure 8: Number Plate Detection

5.2 RESULTS: VISUALISING THE OUTPUT

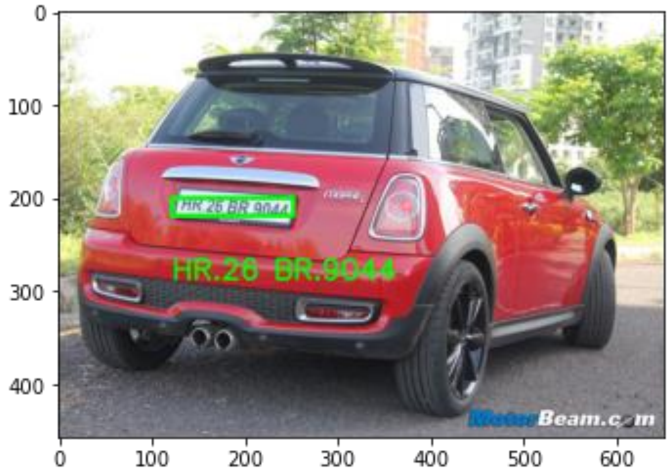


Figure 9: Result 1

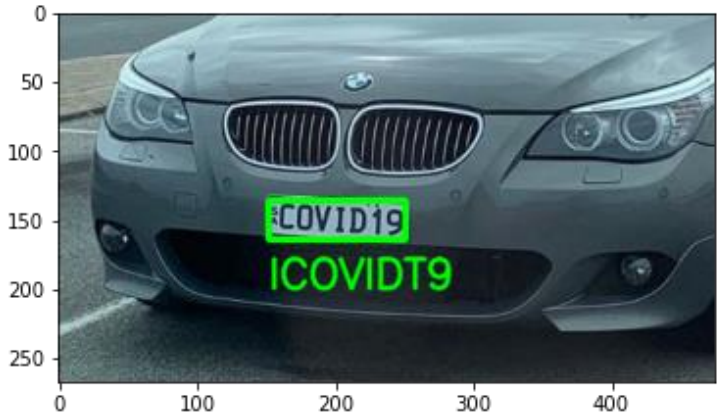


Figure 10: Result 2

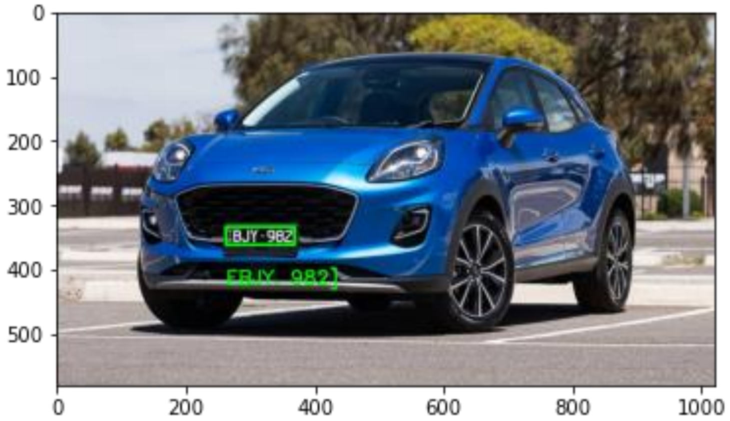


Figure 11: Result 3

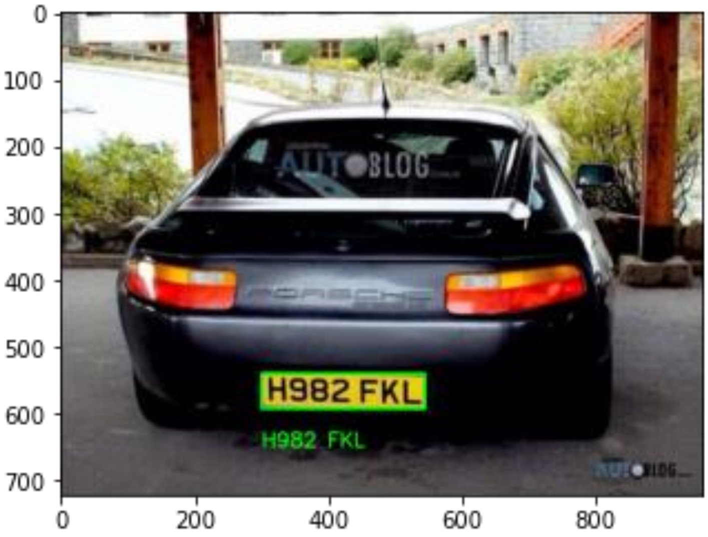


Figure 12: Result 4

5.3 FUNCTIONAL REQUIREMENTS:

The computation result's primary objective is to give the user the processing result. Additionally, it serves as a permanent record of the outcomes for later reference. There are many different kinds of results in general.

- The term "external results" refers to results that are exported outside of the company.

- Business results are only used by the computer department; internal results are those that are viewed and used by the main user and computer within the company.

- the result of a user interface that permits direct communication between users and the system.

- Using simple surveys, discover user preferences, skill levels, and business needs.

5.4 NON-FUNCTIONAL REQUIREMENTS:

It utilizes hardware that is readily available. On a computer with an Intel I5 and 8 GB of RAM, the entire process was carried out. The procedure is broken down into two phases: the training phase, which takes around 15 minutes, and the testing phase, which only requires a short amount of time to provide predictions and gauge accuracy.

5.5 HARDWARE REQUIREMENTS

1. Architecture: 32-bit or 64-bit
2. RAM: 4 GB
3. Storage: 500 GB
4. CPU: 2 GHz or faster

5.5 SOFTWARE REQUIREMENTS

1. Python 3.10 in Jupyter Notebook is used for data pre-processing, model training and prediction.
2. Operating System: Windows 7 and above or Linux based OS or MAC OS.
3. Coding Language - Python.

CHAPTER – 6
CONCLUSION

There have been many algorithms, methods, and approaches proposed for optical character recognition, however the studies' accuracy has been subpar. In this work, we have proposed how to arrange and apply particular approaches, algorithms, and techniques. This well-planned effort is expected to clarify the pertinent ideas and perhaps encourage other advancements in the area. We first discussed the OCR procedure, then we went into great detail regarding the major stages, architecture, and methodology of OCR, and last we demonstrated how these could be applied for automatic license plate recognition.

Due to the large number of stages involved, it is clear that ANPR is a difficult system. Additionally, because each phase depends on the phase before it, 100% overall accuracy is now unattainable. Numerous factors, such as illumination, car shadows, uneven character sizes, different font and backdrop colors, etc., have an impact on the efficiency of ANPR. Some systems can only work under certain conditions, and they might not produce reliable results in difficult situations. The second segment covered the Automatic Number Plate Recognition system, which employs methods from Optical Character Recognition and Image Processing.

Overall, we have looked at and analyzed the methods for automatic and optical character recognition that have already been developed. To recognize and extract characters, we learned how to use the OpenCV and EasyOCR libraries. We created a technique for the identification and detection of license plates from moving vehicles. In addition, we'll focus on the system's shortcomings and strive to minimize errors as much as we can.

REFERENCES

[1] “Optical Character Recognition by open-source OCR Tool Tesseract: A Case Study”, October 2012, International Journal of Computer Applications 55(10):50-56 - Chirag Patel, Atul Patel, Dharmendra Patel.

[2] “Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)”, IEEE Access (Vol. 8) - Jamshed Memon; Maira Sami; Rizwan Ahmed Khan; Mueen Uddin.

[3] “A License Plate-Recognition Algorithm for Intelligent Transportation System Applications” pp. 377-392, 2006 - Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Vassili Loumos, and Eleftherios Kayafas.

[4] “Automatic License-Plate Location and Recognition Based on Feature Saliance”, IEEE Transactions on Vehicular Technology, vol. 58, no. 7, pp. 3781-3785, 2009 - Zhen-Xue Chen, Cheng-Yun Liu, Fa-Liang Chang, and Guo-You Wang.

[5] “Towards a Multinational Car License Plate Recognition system”, Machine Vision and Applications, Springer-Verlag, pp. 173-183, 2006 - Vladimir Shapiro and Georgi Gluhchev Dimo Dimov.

[6] “Fuzzy-based algorithm for color recognition of license plates”, Pattern Recognition Letters, vol. 29, no. 7, pp. 1007-1020, May 2008 - Feng Wang et al.

[7] “Automatic Number Plate Recognition System: Machine Learning Approach (2021)” - Mrs. J. V. Bagade, M Sukanya Kamble, Kushal Pardeshi.

[8] “Automatic Number Plate Recognition Using Tensorflow and EasyOCR (2022)” - Dr. Vishwanath Burkpalli, Abhishek Joshi, Abhishek B Warad.

[9] “Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms (2021)” - Lubna, Naveed Mufti, Syed Afaq Ali Shah.