Number Plate Recognition using Optical Character Recognition (OCR)

Ruchita V. Patel[1] Mr. Saket J. Swarndeep[2]
[1]PG Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1, 2]L. J. Institute of Engineering & Technology (GTU), Ahmadabad, Gujarat, India

Abstract— In this modern era, everything is done digitally. We can witness the currently our traffic signals have a camera built-in, that captures traffic rule violators. My research base on this method. I proposed a system to detect number plate and then transfer it to text to offer many possible features to the owner and cops. I used image processing to detect number plates and then detected number plate pass-through from the OCR. For OCR I used Tesseract OCR. After We get the number plate in text string format. We can offer nearest gas station, nearest emergency helpline, safety on the highway, may reduce car theft cases, Owner details to cops.

Keywords: OCR, Image Processing, Canny, Sobel, Tesseract, Contours, Edge Detection

I. INTRODUCTION

Vehicle Number Plate Recognition using Optical Character Recognition (OCR) Using Canny and Tesseract. Number plate recognition is designed to automatically identify the number plate and automatically recognize it from the Number plate database. number plate recognition has two major parts:

(1) Vehicle number plate extraction,
(2) Optical Character Recognition (OCR).

Number plate recognition is used when the vehicle number plate is read from the database. These images go through the image pre-processing techniques like RGB to gray scale, image resize Edge detection, and so on. The segmental characters area unit normalized Associate in and passed to an OCR algorithmic rule (Tesseract). At last, the optical character info is regenerate into encoded text.

A. Methods & Algorithms

1) Canny Edge Detector:

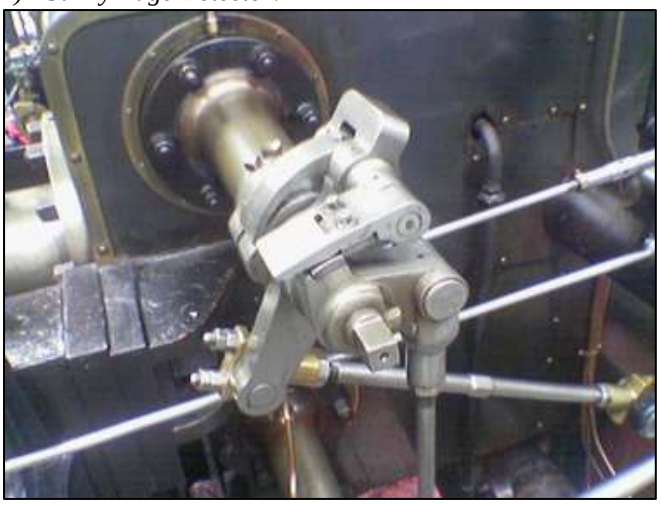

Fig. 1: Image before Edge detection [10]

The Canny edge detector is an operator that detects an outsized vary of edges in pictures using a multi-stage algorithmic program. Canny edge detection may be a technique for extracting helpful structural data from many vision artifacts, whereas reducing the number of information to be processed drastically. It's been used in a variety of computer vision systems. Canny has discovered that the criteria for applying edge detection to a variety of vision systems are remarkably similar. As a result, an edge detection solution that meets these criteria can be used in a variety of scenarios. Canny give us the best result in Digits and Alphabets.

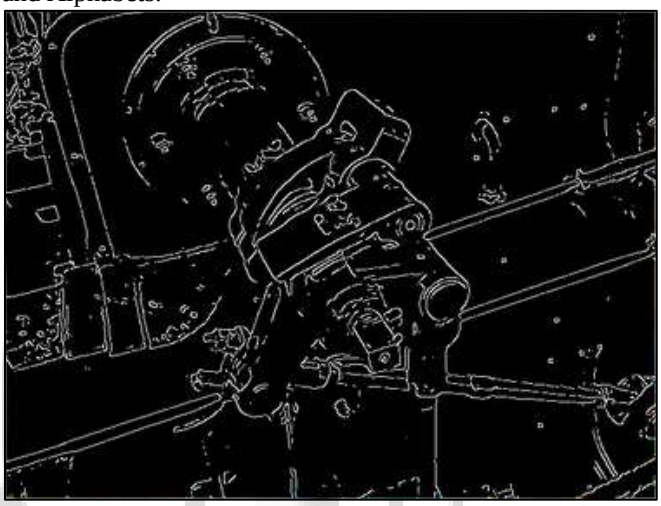

Fig. 2: Image with Canny, after edge detected [10]

2) Sobel Edge Detector:

The Sobel operator, also known as the Sobel–Feldman operator or Sobel filter, is a tool used in image processing and computer vision to emphasize edges in images. For edge detection, the Sobel filter is used. It helps users to analyze the image at every pixel within the image. The result depicts how instant or smoothly the image transfers for each pixel, and therefore how likely the pixels are at the borders.

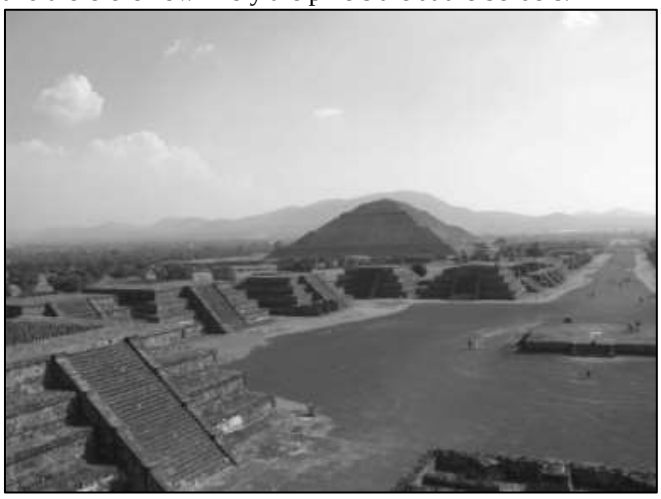

Fig. 3: Image before Sobel edge detection [7]

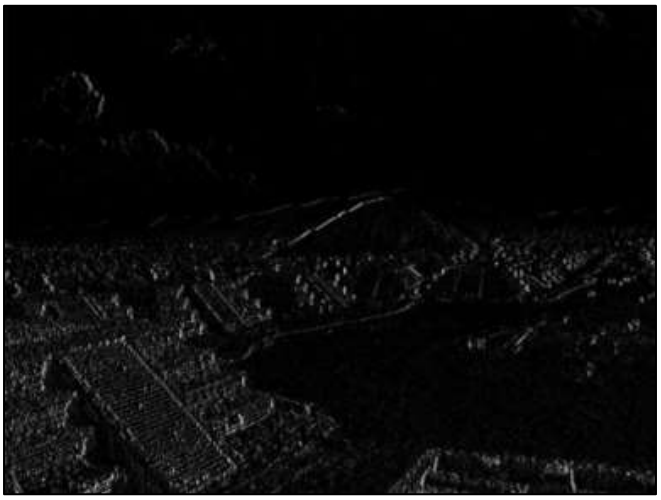

Fig. 4: Image with Sobel edge detection [7]

3) Tesseract OCR (Optical Character Recognition):

Tesseract is an engine for OCR (Optical Character Recognition) for various Operating Systems. Tesseract is an OCR engine with support for Unicode and the ability to recognize more than 100 languages out of the box. It can be trained to recognize other languages. Tesseract examines the text lines to see if they have a set pitch. Tesseract chops the words into characters using the pitch where it finds fixed-pitch text and disables the chopper and associator on these words for the word recognition stage.

4) Engine mode (--OEM):

Tesseract has several engine modes with different performance and speed. Tesseract 4 has introduced an additional LSTM neural net mode, which often works best. Unfortunately, there's no LSTM support on the Android fork yet.

5) OCR

OCR stands for Optical Character Recognition, and it is a technology that addresses the issue of identifying a wide range of characters. Both handwritten and typed characters can be identified and translated into a digital data format that can be read by machines. Consider any serial number or code made up of numbers and letters that need to be encoded. Such codes can be converted to digital outputs using OCR. Several different methods are used in the technology. To put it another way, the image is stored, the characters are removed, and the image is finally remembered.

Working of OCR:

(1) Differentiate word Contours associated with an image. OpenCV contours, Image cropping.
(2) Differentiate letter Contours associated with word Contour Image. OpenCV contours dilation, Image cropping.
(3) Preprocess letter images according to trained OCR input. Keras Framework in Detecting, PIL library in image processing
(4) Consolidate predictions associated OCR model to text. PIL library in Image Processing, Python in consolidation.
(5) Output.

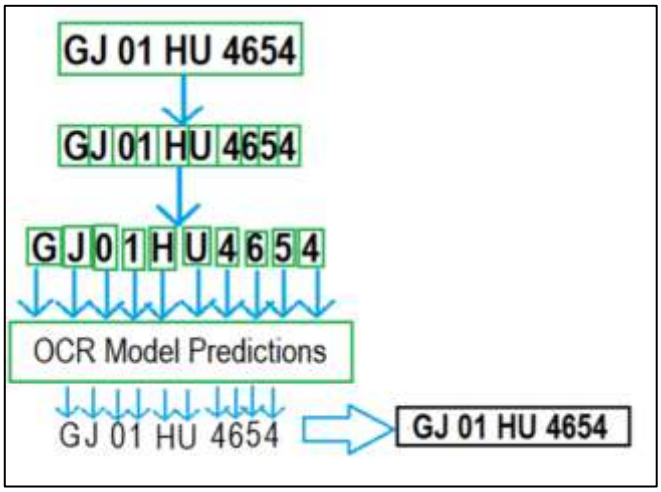

Fig. 5: Working of OCR

B. Objectives

The research's objective is to extract a number plate from the vehicle, transfer that number plate to my database, and also using that number plate to get the number plate in string format. In the future, we can also use the method to provide many possible services to the users.

C. Scope of work

There are many projects already working on this topic. But many of them have many limitations due to their used techniques and methods. However, my research help to archive better result to them, moreover in my project also have some limitations.

D. Bilateral Filter

Bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for an image. It replaces the intensity of each pixel with every nearby pixel.

Fig. 6: Left original Vs. Image with Bilateral filter

E. Smoothing/Blurring:

Smoothing is used to reduce noise or to produce a less pixelated image. Most smoothing methods are based on low-pass filters, but you can also smooth an image using an average or median value of a group of pixels (a kernel) that moves through the image.

II. LITERATURE SURVEY

A vehicle number plate recognition (VNPR) system is implemented for Indian vehicles. For this purpose, they propose a region-of-interest - based filtering method used to detect the candidate regions of the number plate (NP) occurrence. In the proposed filtering method, candidate regions are located in the NP image by detecting vertical edges, removing long edges and stationary regions. Finally, the NP region is segmented from the candidate regions before passing it to the optical character recognition (OCR) system to recognize characters and digits present in the number plate. The novelty of the proposed VNPR system lies in exploring the ROI-based filtering method, which improves the proposed VNPR system's overall performance. The proposed system has been tested using various NP images of vehicles extracted from real-life video sequences that vary along with light, scale, and orientation dimensions. The experimental results demonstrate the robustness of the proposed method. Readability improvement of NP text using image processing techniques and to develop an approach for deciding the best frame among consecutive frames to get the best possible result.

The license plate recognition (LPR) system is an essential system in our life. LPR is an image processing and character recognition system used to recognize any car from the others. An automatic license plate recognition system for the three different Iraqi car license plates was proposed in this paper. Differentiating between the three styles was done depending on the plate size. An optical character recognition (OCR) is used with a correlation approach and template matching for plate recognition by segmenting each number, character, and word into sub-images. The software used is MATLAB R2014a. The algorithm is successfully constructed with a sample of images correctly identified. Many research and creations of intelligent transportation programmers' have grabbed more attention in recent years. An automated, fast, accurate, and robust vehicle plate recognition system has become a need for traffic control and law enforcement of traffic regulations; the solution is ANPR. This paper is dedicated to an improved OCR-based license plate recognition technique using a neural network-trained dataset of object features. A blended algorithm for the recognition of a license plate is proposed. It is compared with existing methods to improve accuracy. The whole system can be categorized under three major modules, namely License Plate Localization, Plate Character Segmentation, and Plate Character Recognition. The system is simulated on 300 national and international motor vehicle LP images, and the results obtained justify the primary requirement.

Countries have different specifications for License Plates (LPs). Therefore, developing one Automatic license plate recognition (ALPR) system that works well for all LPs types is difficult. This paper aims to develop an accurate ALPR for Jordanian LPs. Two-stage Convolutional Neural Networks (CNNs) are used in the proposed approach. The CNNs are based on the YOLO3 framework. The sizes of LPs' characters are tiny compared with the frame size. Therefore, the YOLO3 network architecture is modified to an external network to detect small objects. The proposed approach uses temporal information from different frames to remove false predictions. A set of arrays data structures is used to track the vehicles' LPs and eliminate incorrect ones. To my knowledge, the proposed approach represents the first end-to-end Jordanian ALPR that processes video streams in real-time. To my knowledge, there is no dataset for Jordanian license plates. Therefore, this paper proposes a new dataset called JALPR dataset. The dataset is available online and includes many real videos for moving vehicles in Jordan. Two well-known commercial software packages are used for comparisons. The experimental results in real videos from YouTube show that the proposed approach is very efficient in recognizing the Jordanian license plates and achieved 87% recognition accuracy. In contrast, commercial systems have recognition accuracies that are less than 81%.

A vehicle's plate number is a unique identity by which an individual vehicle can be identified. A vehicle plate recognition system helps to capture a vehicle plate number, extract the numbers on the plate and check the car owner's details. As the number of car owners in a country increases, they are identifying and charging unlawful vehicles on the road has been tedious for law enforcement agents. In this paper, we present an automatic vehicle plate recognition system using Raspberry pi. A Camera was incorporated to help capture the plate number images. It is interfaced to a Raspberry pi processor for authentication. Using the Open Computer Vision (Open CV) and Optical Character Recognition (OCR), the system can extract numbers from the captured plate image and completely automate the license plate recognition. The experimental results from several testing in different locations and conditions show that the system performed better than most of the baseline studies

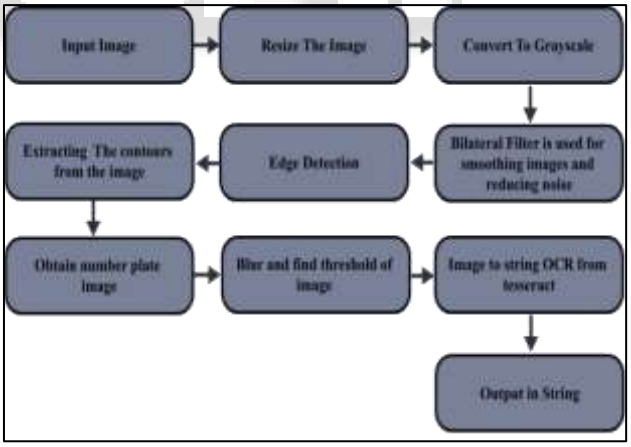

Fig. 7: Proposed flow of my research

III. PROBLEM STATEMENT

In this study, the image of the number plate extract from the vehicle and saved into the database, and then to get the number in text from that image send it to Tesseract OCR.

IV. PROPOSED SYSTEM

In my study one needs to load an image from the database, then image-preprocessing happens on the loaded image like resizing, RGB to gray, noise reduction, Canny edge detection, find/draw contours. After all these steps

extracted number plate is stored in a database. Then Operation performed on the number plate like the threshold of images, then again find contours on the number plate and then it goes to Tesseract OCR and that gives text string as Output.

Fig. 8: Original image

Fig. 9: Image Resize

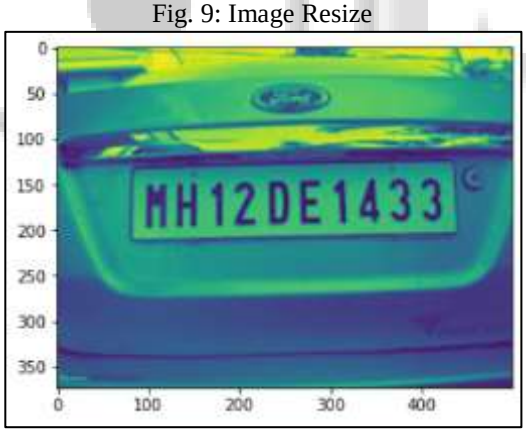

Fig. 10: RGB to Gray

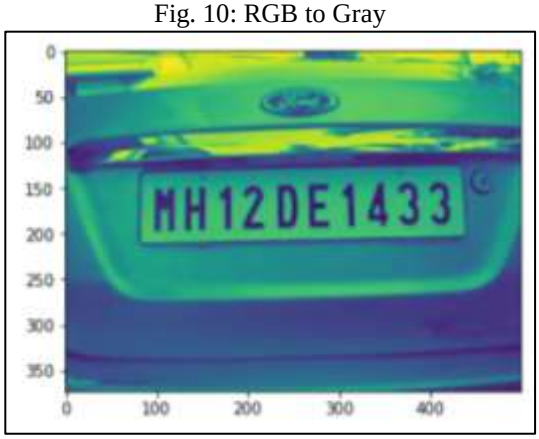

Fig. 11: Bilateral Filter

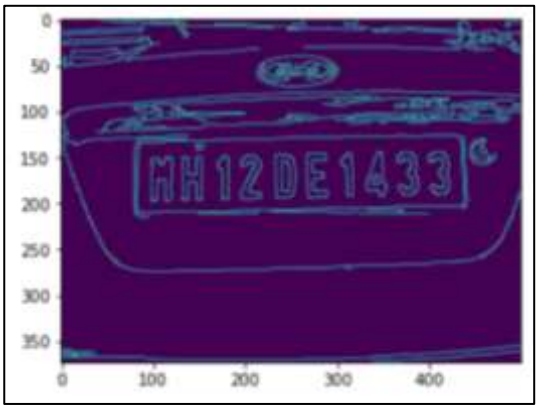

Fig. 12: Canny Edge detection

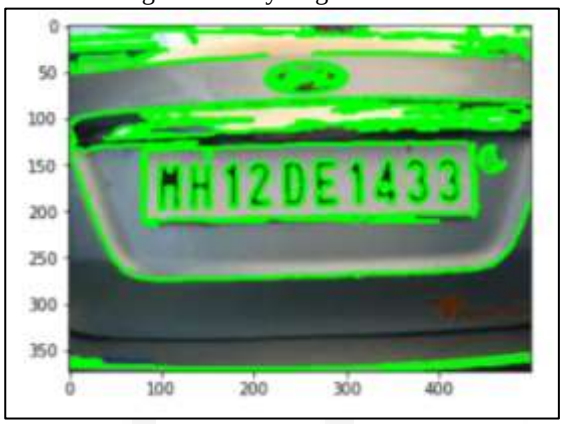

Fig. 13: Find & Draw Contours

Fig. 14: Number plates in different stages

V. RESULT ANALYSIS

A. Edge detection:

Base paper and Existing system: The base paper uses a Sobel edge detector that is very useful for identifying geographical shapes. Where I used the Canny edge detector which is very useful to detect numbers. Sobel needs high-definition images because it analyzed images in very detail.
Image pre-processing: In my base paper, they can use an image that has multiple vehicle number plates where in my system I need to use images that have only one vehicle number plate to get a perfect answer. The base paper

system works in real-time where my system is not working in real-time it uses the database.

B. Contours:

Each system uses contours where base paper only finds contours and my system is finding contours and draws that on my image to get a detailed answer.

C. Output:

Each system gives different outputs like my system gives text string as output where base paper only gives the image.

D. Accuracy:

When I compare 50 images, I get 48 right answers with my methods and algorithms, where in the base paper system it only gives images so both papers have different approaches and methods so it is hard to find accuracy for both papers. In multiple vehicle number plates and in real-time base work better, however, it only gives images as output no string or any other form of output. And when it's about the database and single-vehicle number plate existing system is working better.

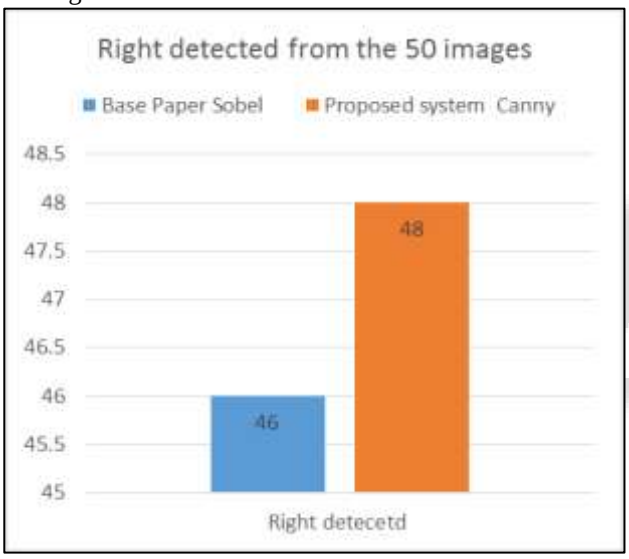

Fig. 15: Right detected Answers from the 50 images

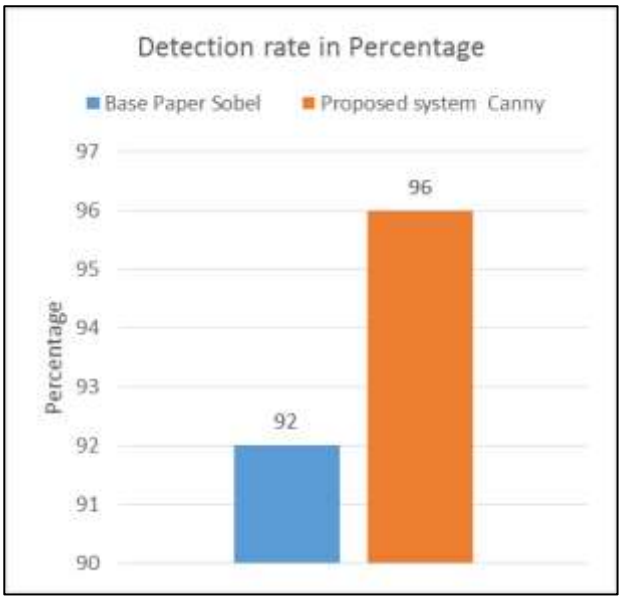

Fig. 16: Accuracy of the papers

The result from 50 images				
	Methods & Algorithms	**Right Detected**	**Accuracy**	**Comments**
Base paper	Sobel Contours Gaussian blur Morphology	46	92	Use multicar images and Sobel is best to detect geographical images
Proposed System	Canny edge detection Bilateral filter, Threshold Contours Tesseract OCR	48	96	Use single-car image and Canny is best for detect numbers and alphabets, also get an answer in text form

Fig. 17: Accuracy table

VI. Conclusion

After implementing both the base paper and my paper, I found that my system has performed better in many fields, as my system works on the database so everything is clean and in a linear manner. Where base paper system works on the real-time scenario so there are many factors which affect that system like lights, exposers, weather, traffic and so on. My system uses OCR so it gives me an output that shows number plate numbers in text form. Where base paper only shows images, which didn't make any difference. There are some limitations in my paper papers but I am looking forward to enhancing them.

References

[1] Agbeyangi, A. O., Alashiri, O. A., & Otunuga, A. E. (2020). Automatic Identification of Vehicle Plate Number using Raspberry Pi. 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 1–4.

[2] Alghyaline, S. (2020). Real-time Jordanian license plate recognition using deep learning. Journal of King Saud University - Computer and Information Sciences, 1–4.

[3] Ghosh, R., Thakre S., & Kumar, P.(2018). A vehicle number plate recognition system using region-of-interest based filtering method. 2018 Conference on Information and Communication Technology (CICT), 1–4.

[4] Kakani, B. V., Gandhi, D., & Jani, S. (2017). Improved OCR based automatic vehicle number plate recognition using features trained neural network. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1–4.

[5] Omran, S. S. & Jarallah, J. A. (2017). Iraqi Car License Plate Recognition Using OCR. Cihan University-Erbil Scientific Journal, 2017(Special-1), 1–13.

[6] Canuma, P. (2020, August 28). Image Pre-processing Towards Data Science. Medium. https://towardsdata-science.com/image-pre-processing-c1aec0be3edf

[7] The University of Auckland. (2021). Edge Detection. Edge Detection, 1(1), 1–8. https://www.cs.auckland.ac.nz/compsci373s1c/PatricesLectures/Edge%20detection-Sobel_2up.pdf

[8] Projects (2021). Open source. Google. https://open-source.google/projects/tesseract
[9] Wikipedia contributors. (2021a, April 11). Canny edge detector. Wikipedia. https://en.wikipedia.org/wiki/Canny_edge_detector
[10] Wikipedia contributors. (2021b, April 28). Tesseract (software). Wikipedia. https://en.wikipedia.org/wiki/Tesseract_(software)
[11] Patel, R., & Swarndeep, S. (2020). Literature Survey on Vehicle Recognition for Augmented Reality using Android Smart Phones. Literature Survey on Vehicle Recognition for Augmented Reality Using Android Smart Phones, 7(12), 1–4. https://www.irjet.net/archives/V7/i12/IRJET-V7I12280.pdf

