# Vehicle Number Plate Detection And Recognition

Sai Vishnu Ramakrishnan
41 Larkhall Street
NL, A1B2C6

***Abstract*-Vehicle number plate is a unique identifier for every individual vehicle which is used for various purposes. The number plate detection is used in toll tac collections, traffic chalans and multilevel parking areas. Identifying vehicle's number plate automatically and extracting the number in text format finds wide areas of applications in this modern era. This project is designed in such a way with a system which clicks the picture of vehicle with the help of cctv cameras and detects the number plate data. The main advantage of this project model and implementation is that there is no need of waiting in a queue for any of the task. This idea could also be implemented in the traffic system and multiple parking areas which could benefit a lot using it.**

**Index Terms— vehicle number plate detection, edge detection algorithm, python, optical character recognition.**

## I. Introduction

In this modern automation system, the automatic recognition techniques in identifying a vehicle's number plate are important research area. The intelligent transportation system has been becoming most significantly read for quite a few years. Identifying vehicles number plate has become the most necessary due to its several applications.

The configuration of authorized plates often differs in many countries by the techniques which are used for automatic detection such as, detection of the number plate, segmentation, and recognition of the characters. The first main step is to capture the image of a car and identify the vehicle's number plate by using image processing techniques such as grayscale, and by applying bilateral filter. Once the image pre-processing steps are done the next step would be to detect and segregate the number plate for the entire image. In-order to achieve this the Sobel edge detection, canny edge detection, and the contour detection is used to extract the vehicle number plate from the entire input image. The final step after the number plate identification from the cropped number plate image would be to extract the characters by using the optical character recognition technique and the final output is produced in the form of string data.

The three key automation recognition techniques which are the most vital errand is to recognize with extraordinarily influence. And for the accuracy of recognition, among various techniques the edge-based strategies are well known and mostly widely used one. After the detection of the number plate, the character division process is carried out. The caught characters are divided by height and width values. For this process the projection technique is used since it is a highly effective method for character division.

## II. System analysis and design

This section briefly describes about the requirements in terms of hardware and software. It also lists out the other requirements which are used for the implementation of this project to achieve the highly efficient number plate data detection and extraction.

### Requirement Analysis

The below mentioned details includes three different aspects of requirements in terms of hardware, software, and others if any.

### *A. Hardware Requirements:*

A system with good configuration which should be capable of running a python application is required. The below mentioned requirements are significant hardware requirements to run the application developed:

- Windows/linux operating system.
- Processor.
- Memory.

### *B. Software Requirements:*

The project is implemented using the python language. The program is designed in such a way that it can run in any python compiler. The IDLE python complier has been used to develop, execute, and test the developed application.

- Language: Python.
- Compiler: IDLE Python compiler/ Any python compiler.

### *C. Other Requirements:*

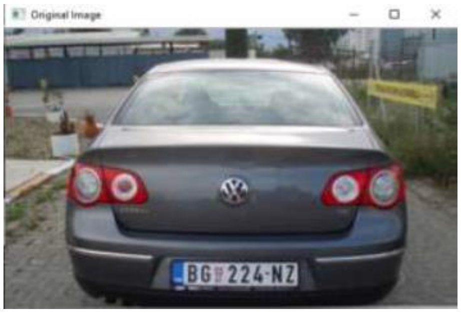
The below mentioned requirements are the additional requirements required for the efficient number plate detection and data extraction.

- Dataset which consists of images of vehicles for which the number plate must be recognised, and characters must be extracted for the detected number plate data into readable format such as string which is nothing but a stream of characters.
- The vehicles pictures which are captured from different angles are used as the data set in the process of detection and conversion of the vehicle number plate date in terms of stream of characters such as string.

## III. Methodology

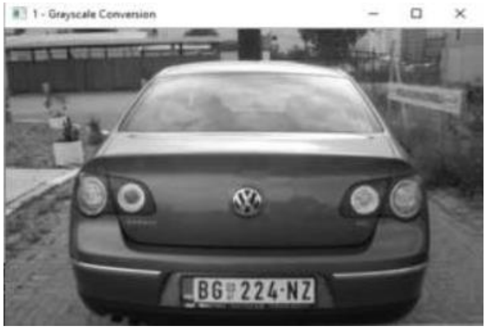
### a) Original Input Image

This is the original image which is given as the input to the detection system implemented in this project to retrieve the licence plate number.



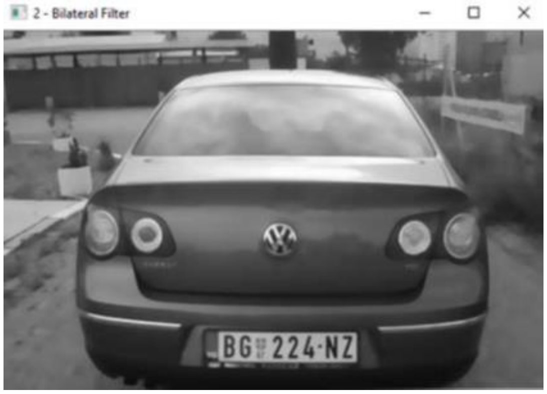
### b) Grayscale

Grayscale is the initial pre-processing step which is applied on the input image to convert the shading spaces to the shades of dim. It shifts between complete dark and complete white. Post this process, the grayscale image contains only brightness information.



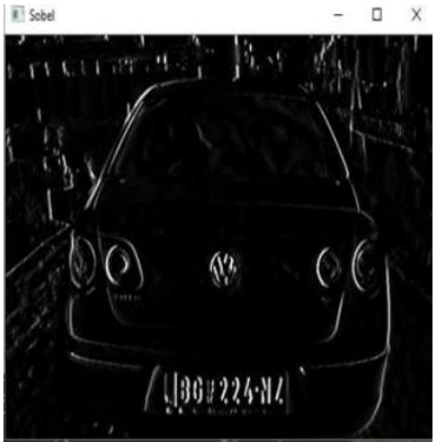
### c) Bilateral Filter -

This process applies bilateral filtering for smoothening the images and for reducing the noise, while preserving the edges. These convolutions often result in a loss of information, since they blur out everything, irrespective of it being noise or an edge.



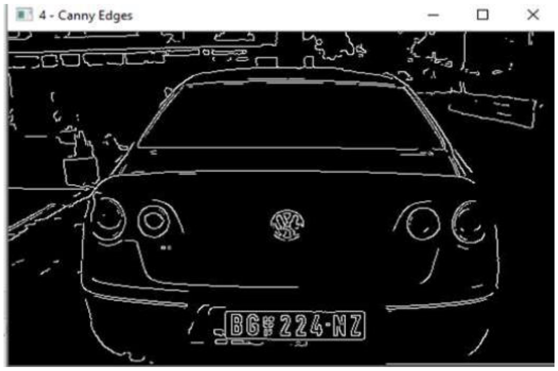
### d) Sobel Edge detection -

In this step, the SOBEL operator is applied to calculate the gradient of image intensity at each pixel within the image. It finds the direction of the largest increase from light to dark and the rate of change in that direction. This is an inclination put together strategy based with respect to first-arrange subordinates. It ascertains the main subsidiaries of the picture independently for the X and Y axes.



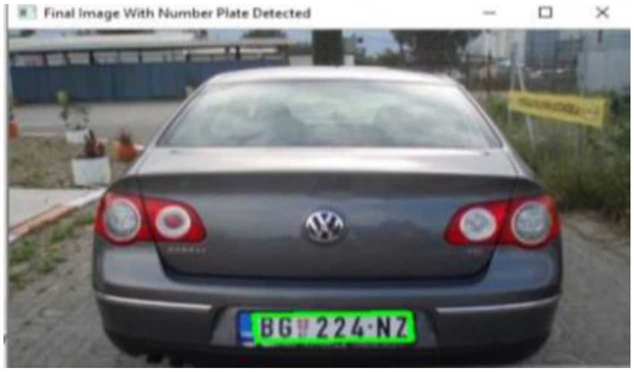
### e) Canny Edge Detection

In this step the wide range of edges in an image is detected, and it is composed of five steps which includes noise reduction, gradient calculation, non-maximum suppression, double threshold, and edge tracking by hysteresis.



### f) Contour detection -

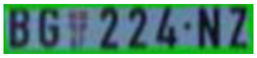
This step is used to identify the geometrical shapes in images, and this can be quite useful for simplifying problems that involve classification or object detection. Using the contour, the number plate from the image is recognized, cropped, and saved into a new image.



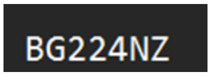
### g) Character recognition

The final stage is the optical character recognition in which the cropped image of the number plate is converted into a string of characters which is the result of this model implementation.

Cropped image of the vehicle number plate is as shown below



Final output of this model in terms of stream of characters or string is as shown below.



```
BG224NZ
```

## IV. Flow Chart

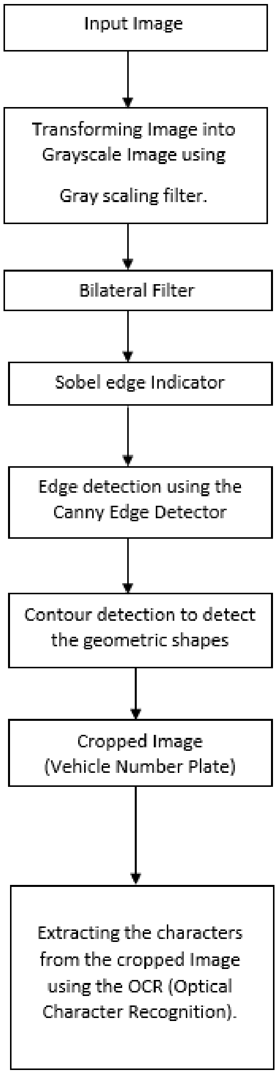
This section shows and explains the flow of the proposed model's implementation. The very initial step of this model would be to read the input image of the vehicle whose number plate is to be extracted, from the specified directory. The next step would be to convert the input image read into a gray scale image which is then followed by the application of bilateral filter on the generated gray scale image by the previous step.

Once the bilateral filter has been applied on the generated gray scale image, the edge detection is implemented using sobel and canny edge detection techniques. This is then followed by the contour detection to detect the geometric shapes.

As the shapes are recognized in the contour detection process, the number plate is detected as well during this step. This results in the detection of the cropped image of the number plate.

The last step post retrieving cropped image of the number plate is to extract the data regarding the number plate of the vehicle as a stream of characters in the form of string. This is achieved using Optical character recognition technique. Thus, the number plate data is extracted in terms of stream of characters as a string from the input image of the vehicle using this model's implementation.

The below mentioned figure depicts the flow of the entire model's implementation for number plate data detection and extraction.


Input Image
Transforming Image into Grayscale Image using
Gray scaling filter.
Bilateral Filter
Sobel edge Indicator
Edge detection using the Canny Edge Detector
Contour detection to detect the geometric shapes
Cropped Image (Vehicle Number Plate)
Extracting the characters from the cropped Image using the OCR (Optical Character Recognition).

## V. Implementation

This section includes the python code used for the implementation of the model for number plate detection. This code picks up the input dataset as the image from a specified directory and implements the steps described in the previous section one after the another. The final output of the below code implementation would be a string which provides the number plate data.

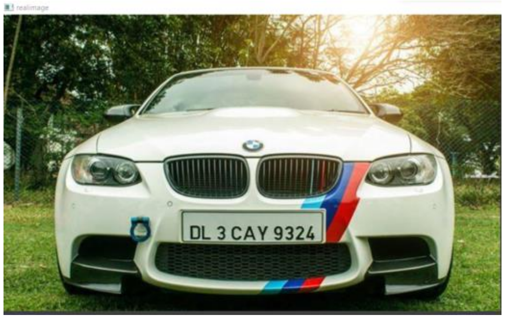
```
import imutils
import pytesseract
import numpy as np
import cv2

pytesseract.pytesseract.tesseract_cmd= r'C://Program Files//Tesseract-OCR//tesseract.exe'
```

```
img = cv2.imread('./bmw.jpg')

cv2.imshow('realimage', img)
cv2.waitKey(0)

#ORIGINAL IMAGE TO GREYSCALE

grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('grayimage', grayscale)
cv2.waitKey(0)

# BILATERAL FILTER

gray = cv2.bilateralFilter(grayscale, 11, 17, 17)
cv2.imshow('2-bilateral Filter', gray)
cv2.waitKey(0)

# EDGE DETECTION USING CANNY

edgedetect = cv2.Canny(gray, 100, 200)
cv2.imshow('edged', edgedetect)
cv2.waitKey(0)

#CONTOUR DETECTION

contour, new = cv2.findContours(edgedetect.copy(),
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
img1 = img.copy()
cv2.drawContours(img, contour, 3, (0, 255, 0), 3)

cv2.waitKey(0)

#TOP 30 CONTOURS

contour = sorted(contour, key = cv2.contourArea, reverse =
True)[:30]
numberplateCnt = None
img2 = img.copy()
cv2.drawContours(img2, contour, -1, (0, 255, 0), 3)
cv2.imshow('contoursimg', img2)
cv2.waitKey(0)
count = 0
index = 7
for c in contour:
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.02 *peri, True)
    if len(approx) == 4:
        NumberPlateCnt = approx
        x, y, w, h = cv2.boundingRect(c)
        new_img = img[y:y + h, x:x + w]
        cv2.imwrite('croppedImage' + str(index) + '.jpg',
new_img)
        index+=1
        break
image = cv2.drawContours(img, [NumberPlateCnt], -1, (0,
255, 0), 3)
cv2.imshow('imageslast', image)
cv2.waitKey(0)

#CROPPED IMAGE

cropped_img_loc = './croppedImage7.jpg'
cv2.imshow('a', cv2.imread(cropped_img_loc))

#EXTRACTING CHARACTERS USING OCR

text = pytesseract.image_to_string(cropped_img_loc,
lang='eng')
print('number is= ', text)
cv2.waitKey(0)
```

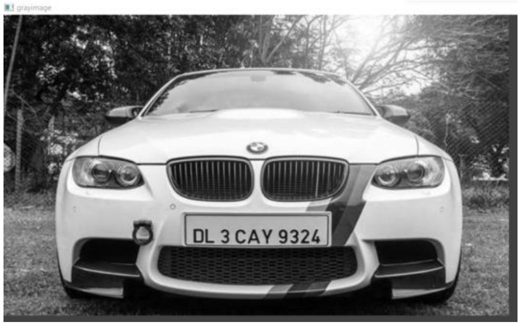
## VI. Output

This section includes the output images and string data of the number plate produced by the model’s implementation.

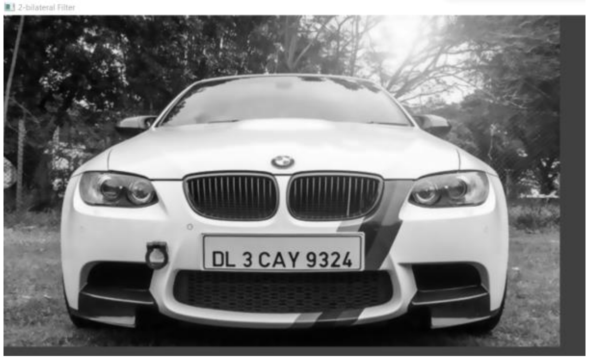
### A. *ORGINAL IMAGE*



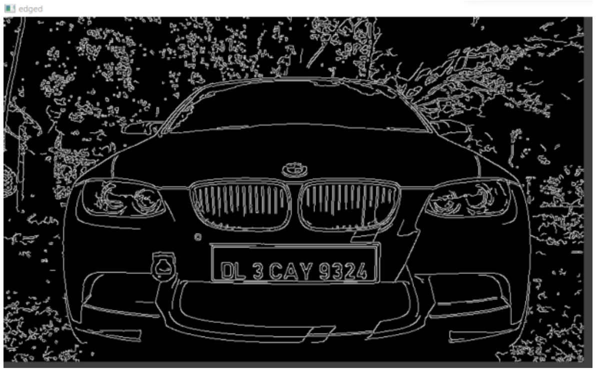
### B. *GRAYSCALE IMAGE*



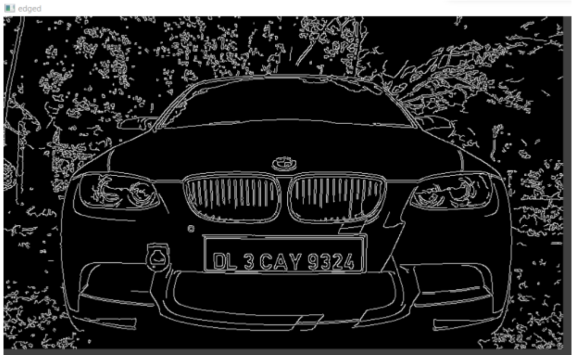
### C. BILATERAL IMAGE



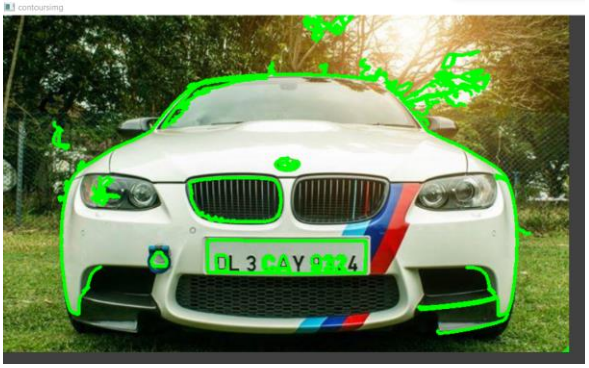
### D. SOBEL EDGE DETECTION



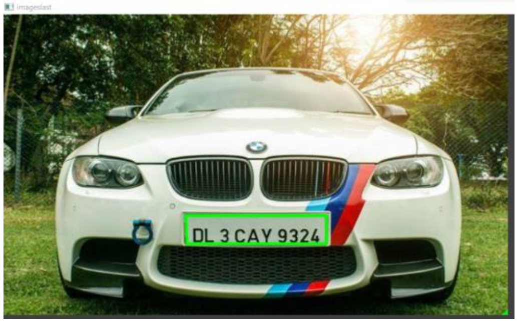
### E. CANNY EDGE DETECTION



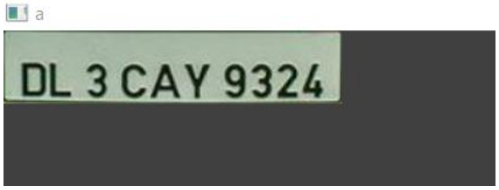
### F. CONTOUR DETECTION



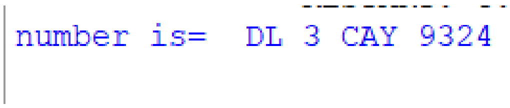
### G. NUMBER PLATE RECOGNITION



### H. OPTICAL CHARACTER RECOGNITION



### I. FINAL OUTPUT IN TEXT FORMAT



```
number is=  DL 3 CAY 9324
```

## VII. REFERENCES AND SIMILAR WORK

The major image processing techniques in this project is referred from the "**Vehicle Number Plate Detection Using Python**" paper for getting the understating of the image processing methodologies and have implemented the OCR phase in addition to the processes as mentioned in **[1]**. The following papers are additionally referred to get the better understanding of the image processing techniques. Layout coordinating has been utilized before in acknowledgment of digits and letters; this paper additionally utilizes the idea of format coordinating with approach in picture handling which is created to remove the vehicle number from the number plate **[2]**. To design and make incredible picture taking care of techniques and computations to limit the license plate in the got picture, to segregate the characters from that number plate, and to recognize each character of the part by using the Open PC Vision Library. This has been executed in K-NN computation and python programming language **[3]**. The k-NN calculation helped in distinctive characters and fuss and separating them into different classes **[4]**. This paper presents an elective strategy for executing ALPR frameworks utilizing Free Programming including Python and the Open PC Vision Library. **[5]**. Pre-processing is the arrangement of calculations applied to the picture to upgrade the quality. The region in the picture that intrigues us is the tag and should be confined from the clamour [6].

## VIII. CONCLUSION

The number plate data of the vehicle is a unique identifier which is used for many critical analysis and purposes. The number plate detection and extraction are efficiently performed in this model and implanted with good performance metrics. This model utilizes various techniques to extract important feature from the input image of the vehicle and ends up with the retrieval of the number plate data as a stream of characters in the form of string data type. The data retrieved could then be utilized for several applications in wide variety of fields and analysis.

## REFERENCES

[1] International Journal of Interdisciplinary Innovative Research &Development **ISSN: 2456-236X** Vol. **05** Issue **02 | 2021** Vehicle Number Plate Detection Using Python.

[2] International Journal of Computer Applications (0975 – 8887) Volume 69– No.9, May 2013 21 Automatic

Number Plate Recognition Systems (ANPR): A Survey.

[3] Lee J.W., Kweon I.S:“Automatic number-plate recognition: neural network approach.” IEEE Vehicle Navigation and Information Systems Conference, vol.3.1. no.12, pp. 99-101 (1998).

[4] T.Pratheeba, “Morphology Based Text Detection and Extraction from Complex Video Scene," International Journal of Engineering and Technology Vol.2 (3), 200-206, 2010.

[5] Pratiksha Jain, Neha Chopra, Vaishali Gupta” Automatic License Plate Recognition using OpenCV” International Journal of Computer Applications Technology and Research Volume 3– Issue 12, 756 - 761, 2014, ISSN:- 2319–8656.