# Image Processing

Synopsis Submitted n Partial Fulfilment Of

The Requirements for The Degree Of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

**Car's Number Plate Detection Using Image Processing**

Of

WEST BENGAL UNIVERSITY OF TECHNOLOGY

By

BANDANA BANERJEE, Roll no - 10901619101
SHAONI BOSE, Roll no - 10900319031
INDRAJIT DAS, Roll no - 10900319093
KRISHNENDU BAG, Roll no - 10900319087

Under the guidance of

PROF. NILADRI SHEKHAR MISHRA

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**NETAJI SUBHASH ENGINEERING COLLEGE**
**TECHNO CITY, GARIA, KOLKATA – 700 152**

<<<2022-23>>

<u>**CERTIFICATE**</u>

This is to certify that this project report titled **Image Processing** submitted in partial fulfilment of requirements for award of the degree Bachelor of Technology (B. Tech) in **Car's Number Plate Detection Using Image Processing** of West Bengal University of Technology is a faithful record of the original work carried out by,

**BANDANA BANERJEE**, **Roll no**. 10901619101, **Regd. No.** 031184 OF 2019-20
**SHAONI BOSE**, **Roll no**. 10900319031, **Regd. No.** 03788 OF 2019-20
**INDRAJIT DAS**, **Roll no**. 10900319093, **Regd. No.** 037774 OF 2019-20
**KRISHNENDU BAG**, **Roll no**. 10900319087, **Regd. No.** 037823 OF 2019-20

under my guidance and supervision.

It is further certified that it contains no material, which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except the assistance drawn from other sources, for which due acknowledgement has been made.

**Date:** ______________ ________________________________

**Guide's signature**

PROF. NILADRI SHEKHAR MISHRA

__________________
Head of the Department

Electronics & Communications Engineering,
NETAJI SUBHASH ENGINEERING COLLEGE,
TECHNO CITY, GARIA, KOLKATA – 700 152

## CERTIFICATE OF APPROVAL

**We hereby approve this synopsis of the project**

**Car's Number Plate Detection Using Image Processing**

carried out by

**BANDANA BANERJEE**, **Roll no**.10901619101, **Regd. No.** 031184 OF 2019-20

**SHAONI BOSE**, **Roll no**. 10900319031, **Regd. No.** 03788 OF 2019-20

**INDRAJIT DAS**, **Roll no**. 10900319093, **Regd. No.** 037774 OF 2019-20

**KRISHNENDU BAG**, **Roll no**. 10900319087, **Regd. No.** 037823 OF 2019-20

under the guidance of

PROF. NILADRI SHEKHAR MISHRA

of Netaji Subhash Engineering College, Kolkata in partial fulfilment of requirements for award of the degree Bachelor of Technology (B. Tech) in << Program name >> of West Bengal University of Technology.

**Date:** ______________

**Examiners' signatures:**

1. ________________________________
2. ________________________________
3. ________________________________

# AUTOMATIC NUMBER PLATE DETECTION SYSTEM USING IMAGE PROCESSING

Image processing allows us to transform and manipulate thousands of images at a time and extract useful insights from them. In our project, we are processing the car’s number plates and retrieving numbers of cars and finding whether the car number is available in our dataset or not.

## ABSTRACT

Number Plate recognition, also called License Plate realization or recognition using image processing methods is a potential research area in smart cities and the Internet of Things. An exponential increase in the number of vehicles necessitates the use of automated systems to maintain vehicle information for various purposes. In the proposed algorithm an efficient method for recognition of Indian vehicle number plates has been devised. We are able to deal with noisy, low illuminated, cross angled, non-standard font number plates. This work employs several image processing techniques such as, morphological transformation, Gaussian smoothing, Gaussian thresholding and Sobel edge detection method in the pre-processing stage, after which number plate segmentation, contours are applied by border following and contours are filtered based on character dimensions and spatial localization. Finally, we apply Optical Character Recognition (OCR) to recognize the extracted characters. The detected texts are stored in the database, further which they are sorted and made available for searching. The project has its own drawbacks and limitations as we are not using higher machine learning or deep learning algorithms but it works efficiently for an average use case.

**CONTENTS**

# LITERATURE REVIEW

## 1.1 INTRODUCTION:

Automatic number plate recognition systems (ANPR) provide a means to overcome the drawbacks and deficiency of successful surveillance of the CCTV cameras. The information extracted from the license plates is mainly used for traffic monitoring, access control, parking, motorway road tolling, and border control, making car logs for parking systems, journey time measurement for toll booth etc. by the law enforcement agencies. The recognition problem is generally sub-divided into 5 parts:

(1) Image acquisition i.e. capturing the image of the license plate
(2) Pre-processing the image i.e. normalization, adjusting the brightness, skewness and contrast of the image
(3) Localizing the license plate
(4) Character segmentation i.e.locating and identifying the individual symbol images on the plate,
(5) Optical character recognition.

## 1.2 APPLICATIONS:

Automatic Number Plate Recognition (ANPR) has a wide range of applications since the license number is the primary, most widely accepted, human readable, mandatory identifier of motor vehicles. Some of the applications are:

1. Housing societies / Apartments: ANPR can be used in housing societies to let in resident's vehicles inside by storing the number plate details in the database. This can hence reduce the man-force near the security gate. Our project works efficiently in this case provided the society or apartment is not very large.

2. Parking: Ticketless parking fee management, parking access automation, vehicle location guidance, parking fee charging, car theft prevention, "lost ticket" fraud, fraud by changing tickets are some areas where ANPR is helpful .

3. Access Control: License plate recognition brings automation of vehicle access control management, providing increased security, car pool management for logistics, security guide assistance, event logging, event management, keeping access diary, possibilities for analysis and data mining. It is very effective in Border and Law controls too.

4. Motorway Road Tolling: Tolls are a common way of funding the improvements of highways, motorways, roads and bridges. Efficient road tolling reduces fraud related to non-payment, makes charging effective, reduces required manpower to process events of exceptions and these can be implemented using ANPR.

5. Journey Time Measurement: Data collected by license plate recognition systems can be used in many ways after processing, feeding back information to road users to increase traffic security, helping efficient law enforcement, optimizing traffic routes and reducing costs and time.

## METHODOLOGY

The proposed methodology consists of four major phases**: pre-processing, detection, recognition and searching** as shown in figure below.

### Step 1: Image pre-processing

**Step 1.1: Noise reduction:** The objective of Gaussian filtering/ Gaussian smoothing is to reduce noise and detail. This will serve well for further image processing steps. For an Image, mathematically, gaussian filter can be expressed as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)}$$

The input image is made to convolve with this 2-D 'G' matrix to obtain a smoothened image. In OpenCV, Gaussian smoothing can be applied using the following function:cv2.GaussianBlur(image,(5,5),0) , where (5,5) refers to the filter size and '0' indicates the model to find the value of standard deviation (sigma) itself.

**Step 1.2: RGB to Grayscale conversion:** Converting RGB image to grayscale saves a lot of time since we have to perform convolution of the image with sobel filter over only one 2D matrix rather than RGB image having 3 channels and making it complicated and also, in case of image edge detection we are focussed on observing the intensity change and it is easier to analyse it in a gray-scaled image.

**Step 1.3: Edge detection using sobel method:** Sobel edge detection works by calculating 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. The following terms are used to perform edge detection using Sobel method:

Run filter on image

$$\frac{\delta f}{\delta x} = S_x f \qquad \frac{\delta f}{\delta y} = S_y f$$

Image gradient

$$\nabla f = fx, fy$$

**Step 1.4: Under-sampling:** The Number plate detection and recognition algorithm are supposed to work at a steady and consistent frame rate. Unsurprisingly, for high-resolution images, image processing algorithms tend to work slow.

**Step 1.5: Morphological transformation:** Top-hat and Black-hat filters are part of Morphological transformations. The Top-hat operation is used to enhance bright objects of interest in a relatively dark background, while the black-hat operation (also known as bottom-hat) is used to enhance dark objects of interest in a relatively bright background.

**Step 2: Number plate detection**

**Step 2.1: Apply Countors:** Contour Tracing, also called as Border following is the algorithm used for generating Contours. A contour is a link of equal intensity points along the boundary.

**Step 2.2: Filter Contours and extract region of interest:** For small regions, especially sharp edges and noise outliers, contours are applied. A human eye can easily figure out that such contours are unnecessary, but this must be incorporated into a program. Initially, Bounding boxes were applied to each contour. Then, for each contour, the following factors were considered such as minimum contour area, minimum contour width and height, minimum and maximum possible aspect ratios. This resulted in the filtering of most of the unnecessary contours, propelling us near to our objective, ie, Detect number plate. (Step 2)

**Step 3: Number plate recognition**

**Step 3.1: Number plate de-skewing:** Skew is the amount of rotation necessary to return an image to horizontal and vertical alignment. Deskewing is a process whereby a skew is removed by rotating an image by the same amount as its skew but in the opposite direction. This results in a horizontally and vertically aligned image where the text runs across the page rather than at an angle.

**Step 3.2: Pre-process region of interest:** It is possible that two or more contours may completely overlap with each other, as in the case with the number 'zero'. The inner contour, if detected in the contour process, may lie completely inside its outer contour. Due to this phenomenon, both contours may get recognized as separate characters during the recognition process.

**Step 3.3: Number plate text recognition:** Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images. We have used this tool finally to obtain the text present in the filtered, de-skewed contour. (Step 3)

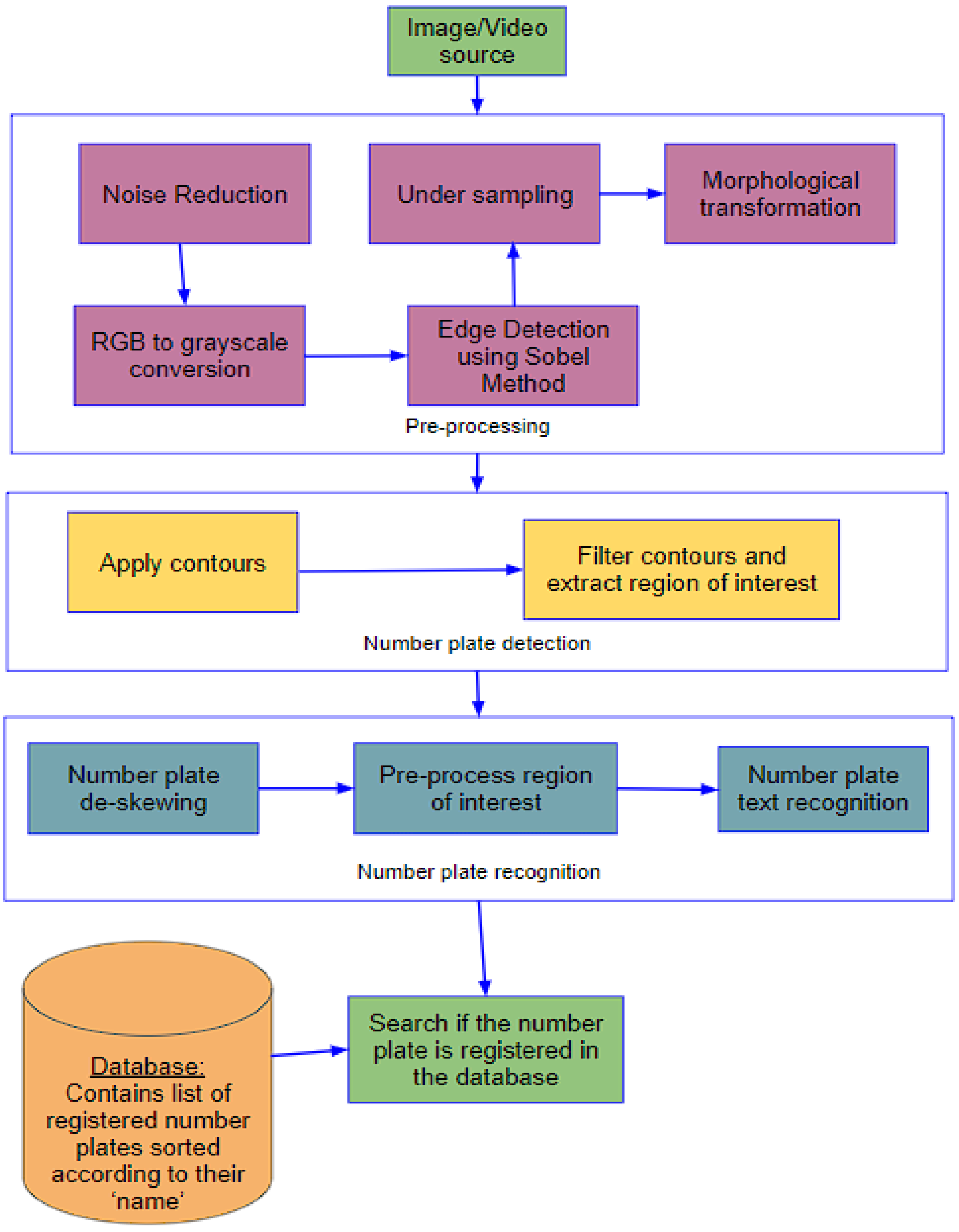
**Step 4: Searching unknown image**

**Step 4.1: Create database:** Using Step-1,2 and 3, register all the vehicles in the dataset and store them in a database after removing other special characters.

**Step 4.2: Sorting:** To make the final stage of searching more efficient, we are performing sorting operations on the detected texts. This is done using a quick sort algorithm. Quick sort is a divide and conquer algorithm. It is not stable and does in-place sorting.

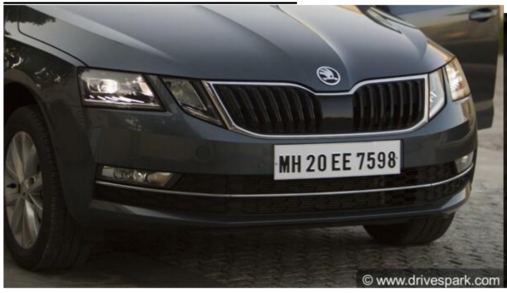
**Step 4.3: Searching:** Pass a new image and follow steps 1,2,3. Obtain the new vehicle's registration number and check if it is present in the database using Binary search method. Binary search is another simple divide and conquer algorithm that is performed on a sorted array/list. It works better than linear search in case of more images in the dataset.

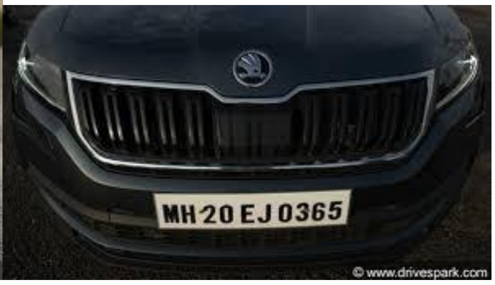
## FLOWCHART:


Image/Video source
Noise Reduction
Under sampling
Morphological transformation
RGB to grayscale conversion
Edge Detection using Sobel Method
Pre-processing
Apply contours
Filter contours and extract region of interest
Number plate detection
Number plate de-skewing
Pre-process region of interest
Number plate text recognition
Number plate recognition
Database: Contains list of registered number plates sorted according to their 'name'
Search if the number plate is registered in the database

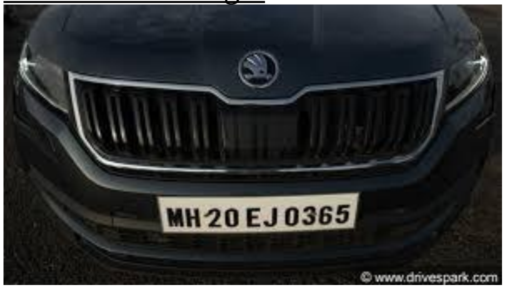
## DATA ANALYSIS

Dataset for Resource :





Data for searching :



Program :

```
import sys
import glob
import os
import glob
import numpy as np
import cv2
from PIL import Image
import pytesseract
import re

#Detecting numberplate
def number_plate_detection(img):
    def clean2_plate(plate):
        gray_img = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)

        _, thresh = cv2.threshold(gray_img, 110, 255, cv2.THRESH_BINARY)
        if cv2.waitKey(0) & 0xff == ord('q'):
            pass
        num_contours,hierarchy = cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)

        if num_contours:
            contour_area = [cv2.contourArea(c) for c in num_contours]
            max_cntr_index = np.argmax(contour_area)
        max_cnt = num_contours[max_cntr_index]
        max_cntArea = contour_area[max_cntr_index]
        x,y,w,h = cv2.boundingRect(max_cnt)

        if not ratioCheck(max_cntArea,w,h):
            return plate,None

        final_img = thresh[y:y+h, x:x+w]
        return final_img,[x,y,w,h]
    else:
        return plate,None

def ratioCheck(area, width, height):
    ratio = float(width) / float(height)
    if ratio < 1:
        ratio = 1 / ratio
    if (area < 1063.62 or area > 73862.5) or (ratio < 3 or ratio > 6):
        return False
    return True

def isMaxWhite(plate):
    avg = np.mean(plate)
    if(avg>=115):
        return True
    else:
        return False

def ratio_and_rotation(rect):
    (x, y), (width, height), rect_angle = rect

    if(width>height):
        angle = -rect_angle
    else:
        angle = 90 + rect_angle

    if angle>15:
        return False

    if height == 0 or width == 0:
        return False

    area = height*width
    if not ratioCheck(area,width,height):
        return False
    else:
        return True

img2 = cv2.GaussianBlur(img, (5,5), 0)
```

```
    img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

    img2 = cv2.Sobel(img2,cv2.CV_8U,1,0,ksize=3)
    _,img2 = cv2.threshold(img2,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

    element = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(17, 3))
    morph_img_threshold = img2.copy()
    cv2.morphologyEx(src=img2, op=cv2.MORPH_CLOSE, kernel=element,
dst=morph_img_threshold)
    num_contours, hierarchy=
cv2.findContours(morph_img_threshold,mode=cv2.RETR_EXTERNAL,method=cv2.CHAI
N_APPROX_NONE)
    cv2.drawContours(img2, num_contours, -1, (0,255,0), 1)


    for i,cnt in enumerate(num_contours):
        min_rect = cv2.minAreaRect(cnt)
        if ratio_and_rotation(min_rect):
            x,y,w,h = cv2.boundingRect(cnt)
            plate_img = img[y:y+h,x:x+w]
            if(isMaxWhite(plate_img)):
                clean_plate, rect = clean2_plate(plate_img)
                if rect:
                    fg=0
                    x1,y1,w1,h1 = rect
                    x,y,w,h = x+x1,y+y1,w1,h1
                    plate_im = Image.fromarray(clean_plate)
                    text = pytesseract.image_to_string(plate_im, lang='eng')
                    return text

#Quick sort
def partition(arr,low,high):
    i = ( low-1 )
    pivot = arr[high]

    for j in range(low , high):
        if   arr[j] < pivot:
            i = i+1
            arr[i],arr[j] = arr[j],arr[i]
    arr[i+1],arr[high] = arr[high],arr[i+1]
    return ( i+1 )

def quickSort(arr,low,high):
    if low < high:
        pi = partition(arr,low,high)

        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)
    return arr

#Binary search
def binarySearch (arr, l, r, x):

    if r >= l:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binarySearch(arr, l, mid-1, x)
        else:
            return binarySearch(arr, mid + 1, r, x)
    else:
        return -1

print("HELLO!!")    print("Welcome to the Number Plate Detection System.\n")

array=[]

dir = os.path.dirname(_file_)

for img in glob.glob(dir+"\OneDrive\Documents\Project_NumberPlate-
Detection\Dataset/*.jpeg") :
    img=cv2.imread(img)

    img2 = cv2.resize(img, (600, 600))
    cv2.imshow("Image of car ",img2)
    cv2.waitKey(1000)
    cv2.destroyAllWindows()

    number_plate=number_plate_detection(img)

    res2 = str("".join(re.split("[^a-zA-Z0-9]*", number_plate)))
    res2=res2.upper()
    print(res2)

    array.append(res2)

#Sorting
array=14uicksort(array,0,len(array)-1)
print ("\n\n")
print("The Vehicle numbers registered are:-")
c=1
for i in array:

    print("Car No",c,"-",i)
    c+=1
```

```
print ("\n\n")

#Searching
for img in glob.glob(dir+"\OneDrive\Documents\Project_NumberPlate-Detection\Search-
image/*.jpeg") :
    img=cv2.imread(img)

    number_plate=number_plate_detection(img)
    res2 = str("".join(re.split("[^a-zA-Z0-9]*", number_plate)))

print("The car number to search is:- ",res2)


result = binarySearch(array,0,len(array)-1,res2)
if result != -1:
        print ("\n\nThe Vehicle is allowed to visit." )
else:
    print ("\n\nThe Vehicle is  not allowed to visit.")
```

## FUTURE WORK

The proposed ALPR system has many shortcomings such as inaccurate results if the image is not of proper texture for example, if the image contains blurred or tilted license plates the results are inaccurate. So, the existing algorithm could be modified to produce better results. Furthermore, the character recognition has limitations such as number of characters which varies from region to region thus there is a great need of a universal algorithm for the same.

.