LICENCE PLATE RECOGNITION USING OPENCV

PROJECT REPORT

Submitted in fulfilment for the JComponent of

Digital Image Processing (SWE1010)

CAL Course
in
M.Tech. – Software Engineering
by

BAGYALAKSHMI R 16MIS0050

Under the guidance of
Dr. Hemalatha S
SITE School

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

School of Information Technology and Engineering
Winter Semester 2018-19

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	ABSTRACT	3
2	PROBLEM STATEMENT	4
3	SYSTEM DESIGN	5
4	SOFTWARE REQUIREMENT SPECIFICATIONS	6
5	IMPLEMENTATION DETAILS	8
5.1	SOURCE CODE	18
6	RESULTS	21
6.1	SCREENSHOTS	22
7	LITERATURE REVIEW	24
8	REFERENCES	27

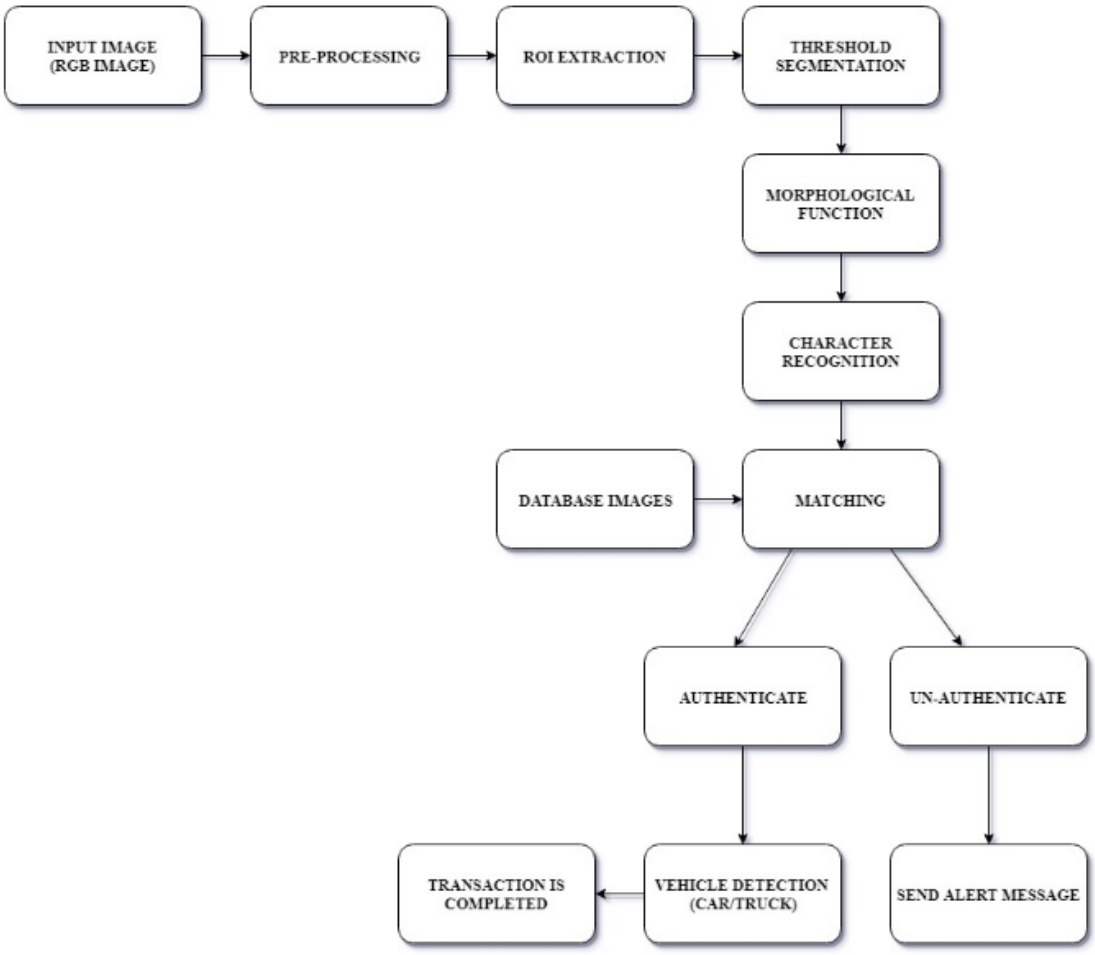
ABSTRACT

Automatic license plate recognition (ALPR) is the extraction of vehicle license plate information from an image. The system model uses already captured images for this recognition process. First the recognition system starts with character identification based on number plate extraction, Splitting characters and template matching. ALPR as a real life application has to quickly and successfully process license plates under different environmental conditions, such as day time. It plays an important role in numerous real-life applications, such as automatic toll collection, traffic law enforcement, parking lot access control, and road traffic monitoring. The system uses different templates for identifying the characters from input image. After character recognition, an identified group of characters will be compared with database number plates for authentication. The proposed model has low complexity and less time consuming in terms of number plate segmentation and character recognition. This can improve the system performance and make the system more efficient by taking relevant sample at the same time compared their advantages and disadvantages, which provide the basis for license plate recognition.

PROBLEM STATEMENT

The tasks of managing and using cars well, cracking theft and robbery of motor vehicles, as well as maintaining the normal order of urban transport have become increasingly heavy. Currently, it has become an important issue for the public security department to tom static management into dynamic change management and to tumor manual management into automation. There are urgent needs to employ Intelligent Transportation System (ITS) so as to make effective management. ITS can perform efficient and reliable management to ambient vehicles under various circumstances. As one of the core technologies of ITS, Vehicle Feature Recognition Technology is an important link to police enforcement system, automated highway toll collection system, Urban Traffic Surveillance System and Intelligent Parking Management System, etc. Thus employing image processing technology to recognize the vehicle license plate number of various kinds of vehicles is not only an important issue for information process technology, but also a research issue which is of great importance in modem transportation management.

SYSTEM DESIGN



SOFTWARE REQUIREMENT SPECIFICATIONS

A. Number Plate Extraction

The captured image is in capital RGB format. It is converted into grayscale image and into binary image.

B. Character Segmentation

The character segmentation part further segments the character individually from the extracted number plate. From input image the first process will be to crop out the number plate characters from starting to the ending point leaving all the extra wide spaces from top to below and from right to left as it is. Characters are equally fit in the plate region. For easy comparison of the input character with the character in the data base the result is normalized into the character set as the size of the images in the database.

C. Optical Character Recognition

The optical character recognition is a recognition method in which the input is an image and the output is string of character. OCR is a process which separates the different characters from each other taken from an image. Template matching is one of the approaches of OCR. The cropped image is compared with the template data stored in database. OCR automatically identifies and recognizes the characters without any indirect input. The characters on the number plate have uniform fonts then the OCR for number plate recognition is less complex as compared to other methods.

D. Template Matching

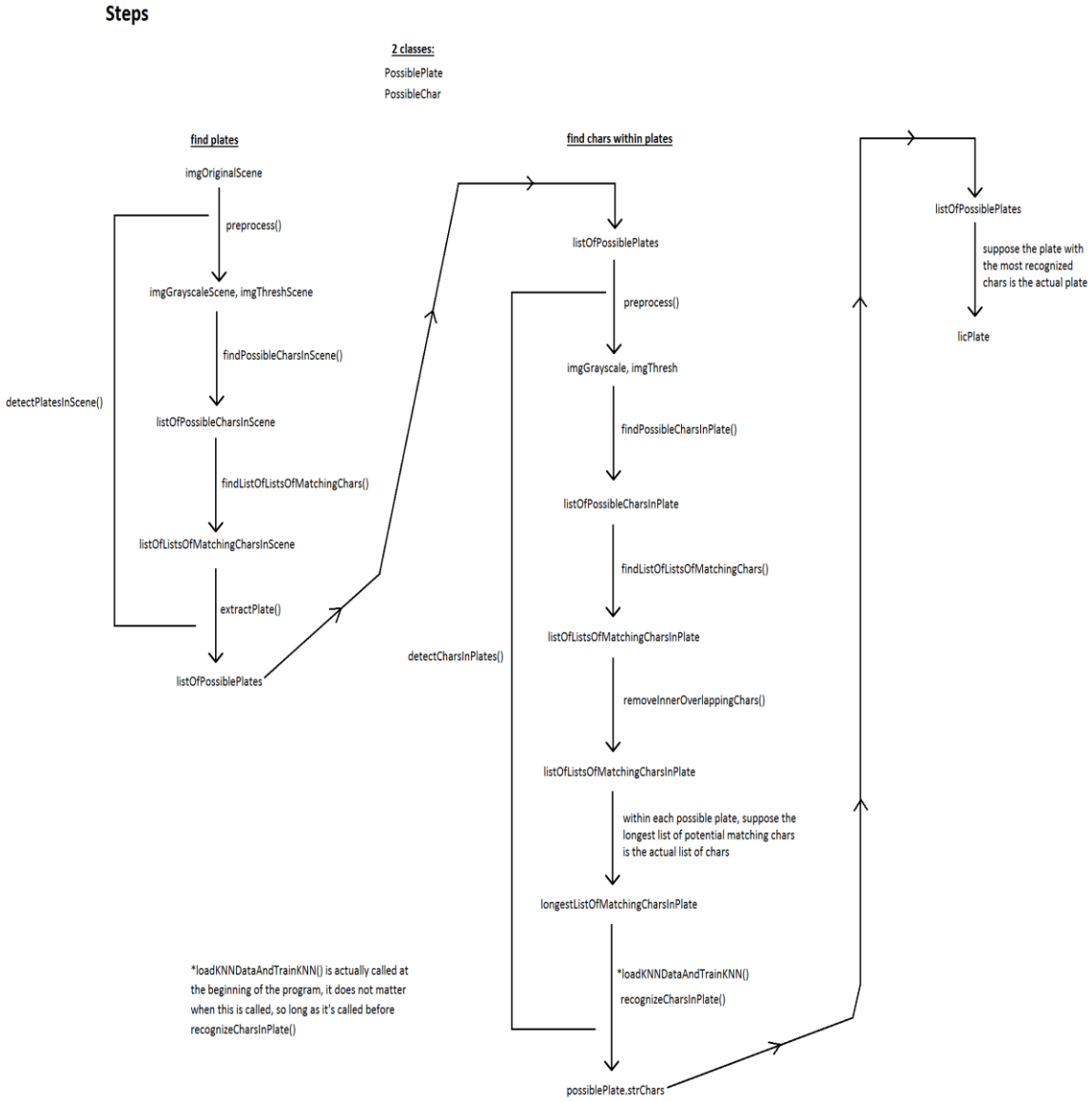
Template matching affects the accuracy of Automatic number plate recognition.

E. PROPOSED ALGORITHM

In this project the processes we have going through are as such.
The first step is capturing the image approximately 1 meter from the number plate with camera. The purpose is to get a clear image without distortion. The second step is cropping the number plate from captured image. The cropped image is the input for the character recognition. The third step is character recognition. The OCR technique is used to recognize the character.

- Supported Operating Systems- Windows
- Programming Languages- Python, OpenCV
- Image Input- Still image from memory or file (BMP, JPEG,JPEG2000)
- Type of Plate- Recognition is country and font independent.Any Latin, Arabic, Chinese, Korean and Cyrillic characters can be recognized. State/country region recognition (optional)
- System Requirements Intel PIII. 1 GHz or higher CPU (128MB RAM)

IMPLEMENTATION



STEPS EXPLANATION

imgOriginalScene

preprocess()

imgGrayscaleScene, imgThreshScene


California
MCLRN F1


California
MCLRN F1

findPossibleCharsInScene()

all contours

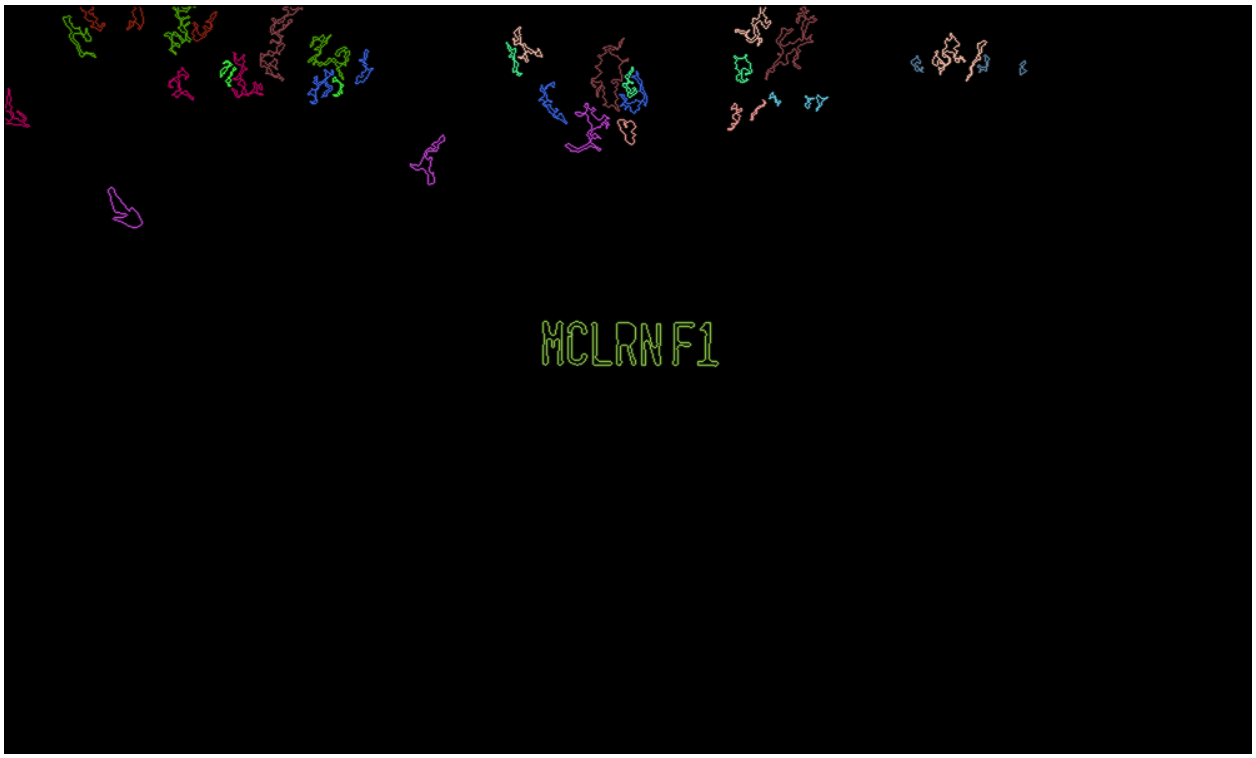
(2362 w/MCLRN F1 image)

vectorOfPossibleCharsInScene

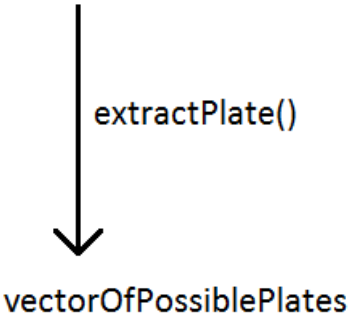
(131 w/MCLRN F1 image)

findVectorOfVectorsOfMatchingChars()

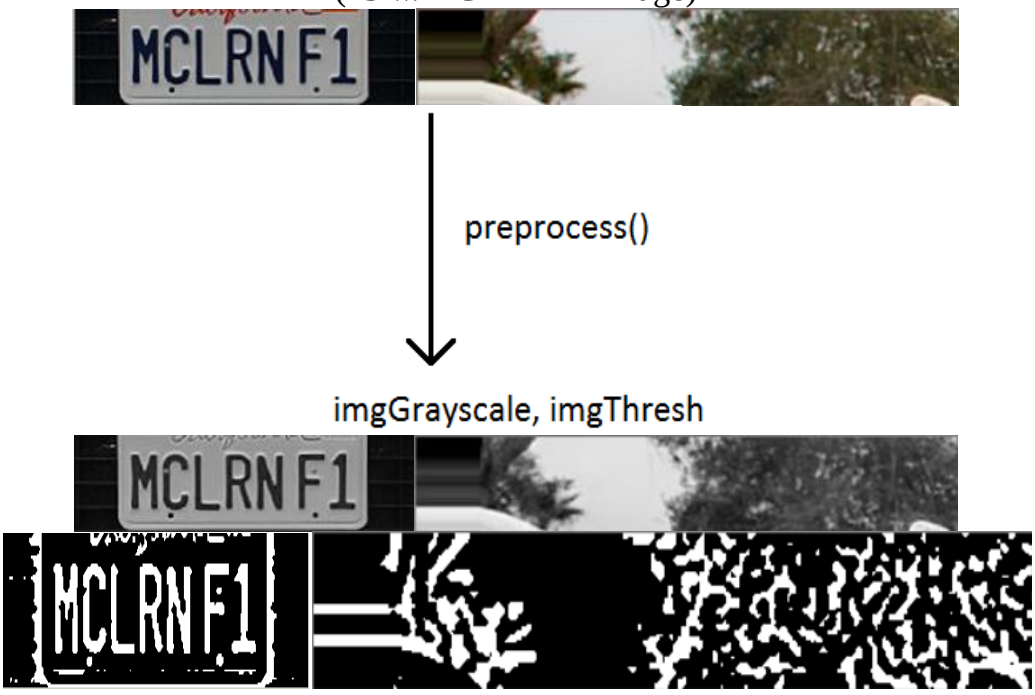
vectorOfVectorsOfMatchingCharsInScene



(13 w/MCLRN F1 image)

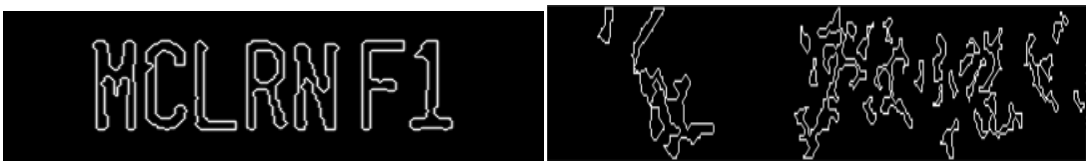


(13 w/MCLRN F1 image)



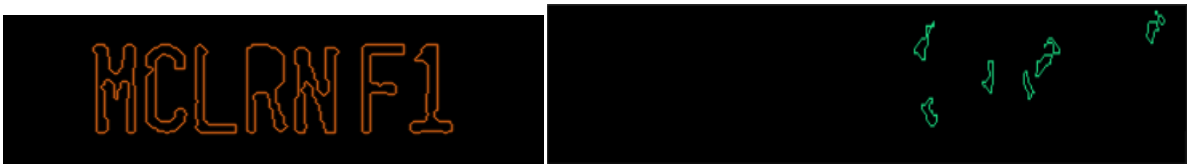
findPossibleCharsInPlate()

vectorOfPossibleCharsInPlate



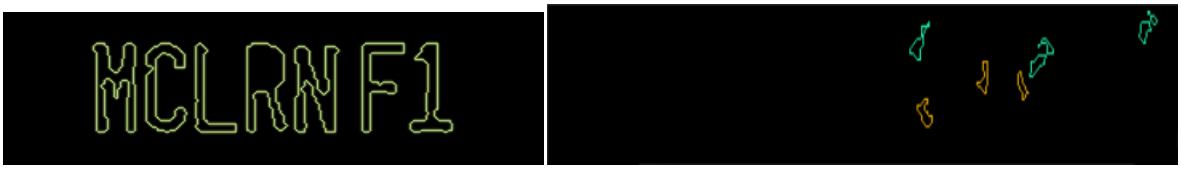
findVectorOfVectorsOfMatchingChars()

vectorOfVectorsOfMatchingCharsInPlate



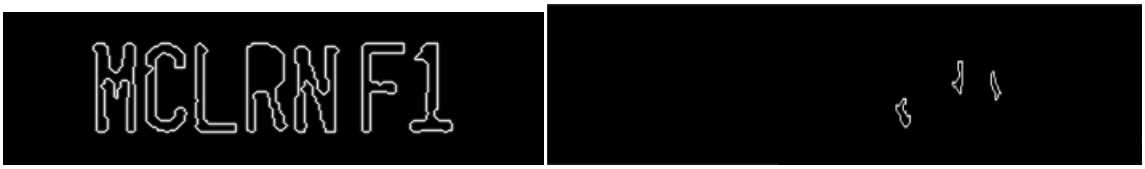
removeInnerOverlappingChars()

vectorOfVectorsOfMatchingCharsInPlate



within each possible plate, suppose the longest list of potential matching chars is the actual list of chars

longestVectorOfMatchingCharsInPlate



recognizeCharsInPlate()

chars found in plate number 0 = MCLRNF1, chars found in plate number 5 = I1I,

possiblePlate.strChars

suppose the plate with
the most recognized
chars is the actual plate


```
Run  Main
C:\Python27\python.exe C:/Users/cdahms/Do

13 possible plates found

license plate read from image = MCLRNF1

-----------------------------------------
```

SOURCE CODE

```
# Main.py

import cv2
import numpy as np
import os

import DetectChars
import DetectPlates
import PossiblePlate

# module level variables
###########################################################################
SCALAR_BLACK = (0.0, 0.0, 0.0)
SCALAR_WHITE = (255.0, 255.0, 255.0)
SCALAR_YELLOW = (0.0, 255.0, 255.0)
SCALAR_GREEN = (0.0, 255.0, 0.0)
SCALAR_RED = (0.0, 0.0, 255.0)

showSteps = False

###########################################################################################
##############
def main():

    blnKNNTrainingSuccessful = DetectChars.loadKNNDataAndTrainKNN()         # attempt KNN
training

    if blnKNNTrainingSuccessful == False:                               # if KNN training was not successful
        print("\nerror: KNN traning was not successful\n")  # show error message
        return                                                  # and exit program
    # end if

    imgOriginalScene  = cv2.imread("LicPlateImages/12.png")               # open image

    if imgOriginalScene is None:                            # if image was not read successfully
        print("\nerror: image not read from file \n\n")  # print error message to std out
        os.system("pause")                                  # pause so user can see error message
        return                                              # and exit program
    # end if

    listOfPossiblePlates = DetectPlates.detectPlatesInScene(imgOriginalScene)          # detect plates

    listOfPossiblePlates = DetectChars.detectCharsInPlates(listOfPossiblePlates)         # detect chars in
plates

    cv2.imshow("imgOriginalScene", imgOriginalScene)          # show scene image
```

```
    if len(listOfPossiblePlates) == 0:                      # if no plates were found
        print("\nno license plates were detected\n")  # inform user no plates were found
    else:                                                   # else
            # if we get in here list of possible plates has at leat one plate

            # sort the list of possible plates in DESCENDING order (most number of chars to least number of chars)
        listOfPossiblePlates.sort(key = lambda possiblePlate: len(possiblePlate.strChars), reverse = True)

            # suppose the plate with the most recognized chars (the first plate in sorted by string length descending order) is the actual plate
        licPlate = listOfPossiblePlates[0]

        cv2.imshow("imgPlate", licPlate.imgPlate)           # show crop of plate and threshold of plate
        cv2.imshow("imgThresh", licPlate.imgThresh)

        if len(licPlate.strChars) == 0:                     # if no chars were found in the plate
            print("\nno characters were detected\n\n")  # show message
            return                                  # and exit program
        # end if

        drawRedRectangleAroundPlate(imgOriginalScene, licPlate)             # draw red rectangle around plate

        print("\nlicense plate read from image = " + licPlate.strChars + "\n")  # write license plate text to std out
        print("----------------------------------------")

        writeLicensePlateCharsOnImage(imgOriginalScene, licPlate)           # write license plate text on the image

        cv2.imshow("imgOriginalScene", imgOriginalScene)            # re-show scene image

        cv2.imwrite("imgOriginalScene.png", imgOriginalScene)           # write image out to file

    # end if else

    cv2.waitKey(0)                                      # hold windows open until user presses a key

    return
# end main

###################################################################################################
def drawRedRectangleAroundPlate(imgOriginalScene, licPlate):

    p2fRectPoints = cv2.boxPoints(licPlate.rrLocationOfPlateInScene)            # get 4 vertices of rotated rect
```

```
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[0]), tuple(p2fRectPoints[1]), SCALAR_RED, 2)
# draw 4 red lines
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[1]), tuple(p2fRectPoints[2]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[2]), tuple(p2fRectPoints[3]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[3]), tuple(p2fRectPoints[0]), SCALAR_RED, 2)
# end function

###################################################################################################
def writeLicensePlateCharsOnImage(imgOriginalScene, licPlate):
    ptCenterOfTextAreaX = 0                         # this will be the center of the area the text will be written
to
    ptCenterOfTextAreaY = 0

    ptLowerLeftTextOriginX = 0                      # this will be the bottom left of the area that the text will
be written to
    ptLowerLeftTextOriginY = 0

    sceneHeight, sceneWidth, sceneNumChannels = imgOriginalScene.shape
    plateHeight, plateWidth, plateNumChannels = licPlate.imgPlate.shape

    intFontFace = cv2.FONT_HERSHEY_SIMPLEX                  # choose a plain jane font
    fltFontScale = float(plateHeight) / 30.0                # base font scale on height of plate area
    intFontThickness = int(round(fltFontScale * 1.5))       # base font thickness on font scale

    textSize, baseline = cv2.getTextSize(licPlate.strChars, intFontFace, fltFontScale, intFontThickness)
# call getTextSize

        # unpack roatated rect into center point, width and height, and angle
    ( (intPlateCenterX, intPlateCenterY), (intPlateWidth, intPlateHeight), fltCorrectionAngleInDeg ) =
licPlate.rrLocationOfPlateInScene

    intPlateCenterX = int(intPlateCenterX)          # make sure center is an integer
    intPlateCenterY = int(intPlateCenterY)

    ptCenterOfTextAreaX = int(intPlateCenterX)      # the horizontal location of the text area is the same
as the plate

    if intPlateCenterY < (sceneHeight * 0.75):                                  # if the license plate is in the
upper 3/4 of the image
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) + int(round(plateHeight * 1.6))      # write the
chars in below the plate
    else:                                                                       # else if the license plate is in the lower 1/4
of the image
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) - int(round(plateHeight * 1.6))      # write the
chars in above the plate
    # end if
```

```
    textSizeWidth, textSizeHeight = textSize                # unpack text size width and height

    ptLowerLeftTextOriginX = int(ptCenterOfTextAreaX - (textSizeWidth / 2))           # calculate the
lower left origin of the text area
    ptLowerLeftTextOriginY = int(ptCenterOfTextAreaY + (textSizeHeight / 2))          # based on the text
area center, width, and height

        # write the text on the image
    cv2.putText(imgOriginalScene, licPlate.strChars, (ptLowerLeftTextOriginX,
ptLowerLeftTextOriginY), intFontFace, fltFontScale, SCALAR_YELLOW, intFontThickness)
# end function

#########################################################################################
##############
if __name__ == "__main__":
    main()
```

RESULT

The objective of this project was to study and resolve algorithmic and mathematical aspects of the automatic number plate recognition systems, such as problematic of machine vision, pattern recognition, OCR and neural networks. The problematic has been divided into several chapters, according to a logical sequence of the individual recognition steps. Even though there is a strong succession of algorithms applied during the recognition process, chapters can be studied independently. ANPR solution has been tested on static snapshots of vehicles, which has been divided into several sets according to difficultness. Sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots which has been captured clearly. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties.

SCREENSHOTS

Original image:

Output:


imgOriginalScene
imgPlate
ZOOMN65
imgThresh
ZOOMN65
NEVADA
ZOOMN65
ZOOMN65
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:10 PM
09-Apr-19

LITERATURE REVIEW:

1. Author [1] presents Vehicle Number Plate Detection (VNPD) system algorithm based on template matching. They have devised an efficient method for recognition of Indian vehicle number plates. The algorithm used modified Otsu's method for threshold portioning. The algorithm aims at addressing the problems of scaling and recognition of position of characters.

2. Author [2] describes Automatic Number Plate Recognition (ANPR) systems which use Optical Character Recognition (OCR) to help determine the individual characters of the number plate. They work on generating key data sets through a simulation process that will generate car number plate images. As a first step, such plates will show variability in character spacing for assessing ANPR systems which will demonstrate the principles for benchmarking. This system avoids the need for carrying out any resource intensive field trials by the Police Force.

3. Author [3] presents inspection and recognition of Malaysian vehicles using optical character recognition (OCR). An intelligent OCR system has been used as a library and using Lab VIEW software. This system is tested for various implementations to ensure that that proposed method can be applied for real implementation.

4. Author [4] discusses implementation of a Number Plate Recognition (NPR). This system is designed to replace the current system of manual entry. Vehicle number plate region is extracted using the image segmentation in an image. The resulting data is then used to balance with the records on a database. The system is implemented and simulated in Mat lab, and performance is measured on the real image.

5. Author [5] develops an accurate and automatic number plate recognition system.

6. Author [6] discusses about the noisy contain image and applied Adaptive median filter to remove noise from the image and gives output as a filtered image. The main purpose of this is to remove noise from the image. For recognition new test has been applied using salt and pepper algorithm for removal of noise with an adaptive median filter.

7. Author [7] presents the automatic vehicle identification system using vehicle license plate. The OCR method used for recognition is sensitive to misalignment and to different sizes.

AUTHOR NAME AND YEAR	DESCRIPTION	TECHNIQUES USED	OUTCOMES
Hanit Karwal et al. (IEEE 2015) [1]	The proposed technique shows the necessity of use of automated systems to maintain vehicle information. In the proposed algorithm an efficient method for recognition for Indian vehicle number plate has been devised.	Paper presents VNPD system based on template matching with Normalized cross correlation. It also uses modified Otsu's method for threshold partitioning.	It obtained the accuracy of 98.07 %.
S. Ramalingam et al. (IEEE 2014) [2]	In this paper the author has determined the impact on ANPR performance caused by illegal spacing between characters of number plate. The causes of inaccurate ANPR read data are examined in detail and recommendations made as to how improvements could be introduced to minimize the risk of misreads.	In this key data sets are generated through a simulation process that will generate car number plate images. In this Optical Character Recognition (OCR) is used.	Variable spacing does appear to have an adverse impact on ANPR engines. They were not able to predict the impact of illegal spacing or syntax rules that cause a complete fail to capture or misreads. Therefore more work has to b done.
Abd Kadir Mahamad et al. (SPRINGER 2014) [3]	Automatic inspection and recognition system has been proposed for Malaysian vehicles using optical character recognition (OCR). This system is tested for various implementations to ensure that that proposed method can be applied for real implementation	System is based on Digital Image Processing and Optical Character Recognition (OCR). An intelligent OCR Training Interface has been used as a library and the system has been developed using LabVIEW Software.	The proposed system shows good performance for inspection and can recognize an alphabets and numbers of vehicle Number plate.
Mr.G.T.Sutar et al. (IJIRSET 2014) [4]	They implement Number Plate Recognition (NPR). This system is designed keeping in mind automation of number plate detection for the security reason that could replace the current	In this vehicle number plate region is extracted using the image segmentation in an Image. Optical Character Recognition (OCR) technique is used for	The result shows that the system works against different lightening conditions and can be implemented on the entrance of a highly restricted areas. System

	system of manual entry.	character recognition.	successfully detects and recognizes the vehicle number plate on real images with an accuracy of 93%
Kuldeepak et al. (IJECCT 2012) [5]	They introduced that high degree of accuracy has been required by the number plate recognition when roads are busy and number of vehicles are passing through. For this there is a need of automatic number plate recognition. It also gives us warning for the stolen vehicle which cannot be possible for man handling services.	Character segmentation is used to separate each image from the background. The proposed system has been implemented using vision assistant 8.2.1 & labview 11.0.	By optimizing various parameters, they have achieved an accuracy of 98%. But for the tracking stolen vehicles and monitoring of vehicles an accuracy of 100% cannot be compromised with. Therefore to achieve better accuracy optimization is required.
Quraishi et al. (RAIT 2012) [6]	The purpose of the author to present this work is to provide a new approach for image recognition using Artificial Neural Network (ANN).	In this alternative solution for Object Recognition using Artificial Neural Networks (ANN) is used.	If the avg. error is less than 45% ANN can be applied for training & testing for the purpose of recognition. Therefore the test image is recognized & matched successfully with original image.
Qadri et.al (ICETC 2009) [7]	In this the automatic vehicle identification system using vehicle license plate is presented. The system use series of image processing techniques of the system for identifying the vehicle from the database stored in the PC.	Optical character recognition (OCR) technique is used for character recognition. The system is implemented & simulated in MATLAB & performance is tested on real images.	There is a need of an improvement. High resolution camera can be used to increase the speed. The OCR method used is sensitive to misalignment and to different sizes.

REFERENCES

[1] Karwal, Hanit, and Akshay Girdhar, Vehicle NumberPlate Detection System for Indian Vehicles, *Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on*. IEEE, 2015.

[2] S. Ramalingam, M. Rhead, and R. Gurney. Impact of character spacing on the performance of Automatic Number Plate Recognition (ANPR) systems through simulation, *Security Technology (ICCST), 2014 International Carnahan Conference on*. IEEE, 2014.

[3] Mahamad, Abd Kadir, Sharifah Saon, and Sarah Nurul Oyun Abdul Aziz, A Simplified Malaysian Vehicle Plate Number Recognition, *Recent Advances on Soft Computing and Data Mining*. Springer *International Publishing*, 2014. 379-388.

[4] Sutar, G. T., and A. V. Shah, Number Plate Recognition Using an Improved Segmentation, *International Journal of Innovative Research in Science Engineering and Technology* 3 (2014): 12360-12368.

[5] Kuldeepak, Monika Kaushik, and Munish Vashishath, License Plate Recognition System based on Image Processing Using LabVIEW, *International Journal of Electronics Communication and Computer Technology (IJECCT) 2.4* (2012).

[6] Quraishi, Md Iqbal, J. Pal Choudhury, and Mallika De, Image recognition and processing using Artificial Neural Network, *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*. IEEE, 2012.

[7] Muhammad Tahir Qadir, and Muhammad Asif, Automatic number plate recognition system for vehicle identification using optical character recognition, *Education Technology and Computer, 2009. ICETC'09. International Conference on*. IEEE, 2009.

[8] Kaur, Er Kavneet, and Vijay Kumar Banga, Number Plate Recognition Using OCR Technique, *International Journal of Research in Engineering and Technology 2.09* (2013).

[9] Maeno, Kazuki, et al, Light field distortion feature for transparent object recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.

[10] Zhai, Xiaojun, Faycal Bensaali, and Reza Sotudeh, Real-time optical character recognition on field programmable gate array for automatic number plate recognition system, *Circuits, Devices & Systems, IET 7.6* (2013): 337-344.

[11] Friedrich, Markus, Prokop Jehlicka, and Johannes Schlaich, Automatic number plate recognition for the observance of travel behaviour, *Processding of 8th International Conference on Survey Methods in Transport*. 2008.

[12] Kranthi, S., K. Pranathi, and A. Srisaila, Automatic number plate recognition, *Int. J. Adv. Tech 2.3* (2011): 408-422.