DAYANANDA SAGAR UNIVERSITY

A Pattern Recognition Report

ON

"Licence Plate Recognition"

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

DEVANSH AWASTHI (ENG17CS0065)

DEEPAK PUROHIT (ENG17CS0062)

BVM ANIRUDH (ENG17CS0048)

VII Semester, 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF ENGINEERING

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE

BANGLORE-560068

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068

CERTIFICATE

This is to certify that Mr. Devansh Awasthi USN ENG17CS0064, Mr. Deepak R Purohit bearing USN ENG17CS0062 and Mr. BVM Anirudh bearing bearing USN ENG17CS0048 has satisfactorily completed their Pattern Recognition Project Report as prescribed by the University for the Seventh semester B.Tech. Program in Computer Science & Engineering during the year 2020 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: _____

_____

Signature of the faculty in charge

_____

Signature of the Chairman

Abstract:

Traffic control and vehicle owner identification has become major problem in every country. Sometimes it becomes difficult to identify vehicle owner who violates traffic rules and drives too fast. Therefore, it is not possible to catch and punish those kinds of people because the traffic person might not be able to retrieve vehicle number from the moving vehicle because of the speed of the vehicle. Therefore, there is a need to develop Automatic Number Plate Recognition system as a one of the solutions to this problem. There are numerous Automatic Number Plate Recognition systems available today. These systems are based on different methodologies but still it is really challenging task as some of the factors like high speed of vehicle, non-uniform vehicle number plate, language of vehicle number and different lighting conditions can affect a lot in the overall recognition rate. Most of the systems work under these limitations.

Aknowledgement

The satisfaction that accompanies the successful completion of task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman, Dr. Sanjay Chitnis**, for providing necessary depatmental facilities, moral support and encouragement.

We are very much thankful to our **Guide, Prof. Lavanya B Koppal** for providing help and suggestions in completion of this pattern recognition project completely.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directy or indirectly helped us in the successful completion of this project work.

Table Of Contents

1. Introduction

Number Plate Recognition System is an essential stage for the automation of traffic system. Use of vehicles is getting increased in today's era that is why traffic control is being tough. It is hard to store and maintain the record of vehicles manually.Number Plate Recognition System can be used for better control of vehicles and for store and maintain the record of vehicles automatically.

License plate recognition (LPR) system is able to detect vehicles on the monitored road and automatically extract vehicle license information and process it. LPR is a modern intelligent transportation system, an important part of one widely used. It is based on digital image processing, pattern recognition, computer vision and other technology.

By means of some post-processing, it can be used in highway toll, parking management, weighing systems, traffic guidance, traffic enforcement, road inspection, vehicle scheduling, vehicle inspection and many other occasions. To maintain traffic safety and urban security, to prevent traffic congestion, to achieve traffic automation management, the LPR system has practical significance.

1.1 Scope

License Plate recognition is one of the techniques used for vehicle identification purposes. The sole intention of this project is to find the most efficient way to recognize the registration information from the digital image (obtained from the camera). This process usually comprises of three steps. First step is the license plate localization, regardless of the license-plate size and orientation. The second step is the segmentation of the characters and last step is the recognition of the characters from the license plate. Thus, this project uncovers the fundamental idea of various algorithms required to accomplish character recognition from the license plate during Template Matching.

2. Problem Statement

The main focus in this research project is to experiment deeply with, and find alternative solutions to the image segmentation and character recognition problems within the License Plate Recognition framework. Three main stages are identified in such applications. First, it is necessary to locate and extract the license plate region from a larger scene image. Second, having a license plate region to work with, the alphanumeric characters in the plate need to be extracted from the background. Third, deliver them to an OCR system for recognition. In order to identify a vehicle by reading its license plate successfully, it is obviously necessary to locate the plate in the scene image provided by some acquisition system (e.g. video or still camera).

3. Overview Of Existing System

The high performance fibre optic sensors are used for detection of moving vehicles. A typical installation consists of an interface device with transmitter (LED), receiver (photo detector), and light guide connection cable (feeder) and fibre optic sensor. As the vehicle passes over the sensors there is a change in the signal levels obtained from the sensors. The output signals from the fibre optic sensors are fed into a signal processing and data evaluation unit which comprises of the algorithm, which computes axle count, axle spacing, vehicle lengths and vehicle classes based on time, distance formula, and amount of micro bending.

There are a number of possible difficulties that the software must be able to cope with. These include:

1. Poor image resolution, usually because the plate is too far away but sometimes resulting from the use of a low-quality camera.

2. Bad images particularly blur.

3. Poor lighting and low contrast due to overexposure, reflection or shadows.

4. An object obscuring (part of) the plate, quite often a tow bar, or dirt on the plate.

5. A different font, popular for vanity plates (some countries do not allow such plates, eliminating the problem).

6. Lack of coordination between countries or states. Two cars from different countries or states can have the same number but different design of the plate.

3.1 Proposed Solution

The process of automatic number plate recognition consists of four main stages:

(1) Preprocessing

(2) License plate localization

(3) Character segmentation

(4) Character recognition

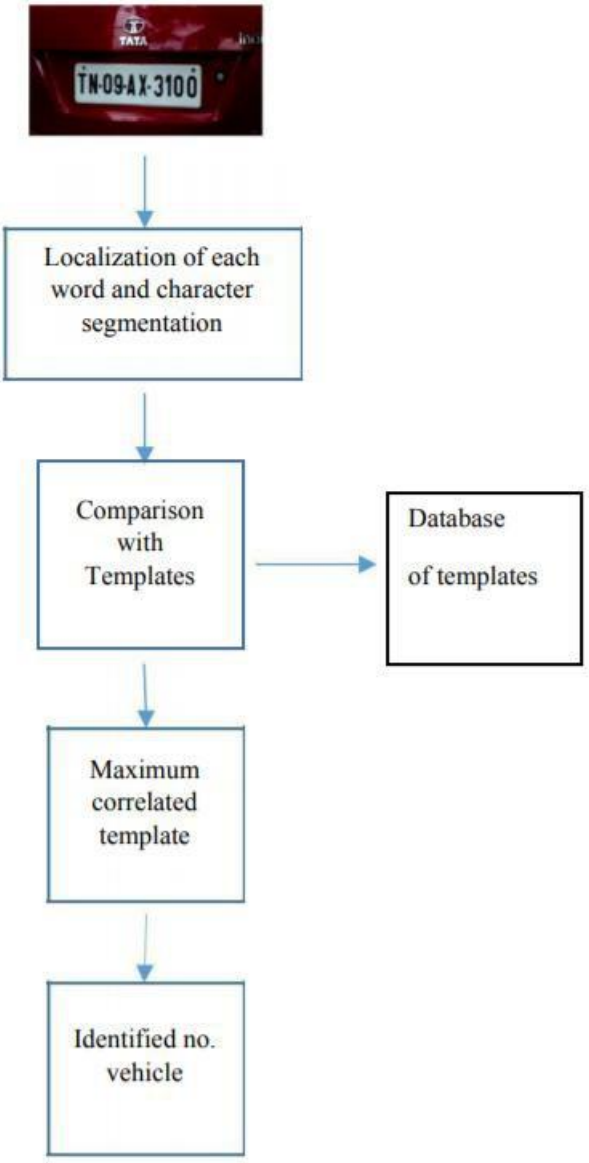
Preprocessing:
As mentioned before, the system of automatic number plate recognition faces many challenges. So, this step is essential to enhance the input image and making it more suitable for the next processing steps. The first step done in the preprocessing is to apply minimum filter to the image in order to enhance the dark values in the image by increasing their area. This is mainly done to make the characters and the plate edges bold, and to remove the effect of the light diagonal strips that appear in the characters and edges of the license plates.

License Plate Localization:
In this stage, the location of the license plate is identified and the output of this stage will be a sub-image that contains only the license plate.

Character Segmentation:
This stage is meant for segmentation of the characters from the plate. The output of this stage is a set of monochrome images for each candidate character in plate.

Character recognition:
The goal of this stage is to recognize and classify the binary images that contain characters received from the previous one. After this stage every character must have a label and an error factor, and this error factor if greater than a predefined value will be used to reject false characters accidently passed from the previous steps.

3.2 Block Diagram



The objective of this segment is to provide a detailed information about how to find a number plate in the captured image? Generally a monochrome camera with colour camera is used in ANPR system. Finding out the number plate area is a needed pioneer to certified plate identification. We can combine the approaches used to trace the number plate's position or section in images into three processing categories. To recognize separated characters, some processes use pattern image, grayscale, and colour. Character separation is a crucial method for recognition of characters, which we can similarly separate out /matching of template or learningbased classification.The flow chart explained in above figure,shows the various method involved in recognising the plate numbers.

4. Implementation

```
# 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("3.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()
```

5. Output Screenshots

Input Image-1:

Output Image-1:

Plate Image-1:

Threshold Plate Image-1:

Final Output-1:

Input Image-2:

Output Image-2:

Plate Image-2:

Threshold Plate Image-2:

Final Output-2:

6. Conclusion

The image segmentation problem in license plate recognition for Indian License Plates has been examined in two stages: license plate Localization and extraction from the scene followed by the separation of the characters from the previously extracted license plate region background. Various approaches for license plate detection in an image, and extensive experiments have been devised to test them are presented. The purpose of this project has been to investigate the scope of automatic license plate recognition under minimal restrictions. The main objective is aimed at contributing towards the research in the fields of machine vision, pattern analysis and image processing. The system developed investigates the possibility of automating the whole process of license plate recognition for a wide range of environments. Given an input image, the system extract extracts the license plate, isolates the characters, and finally identify the characters.

7. References

(1) https://www.anprcameras.com/about-us/understanding-anpr/

(2) https://en.wikipedia.org/wiki/Automatic_number-plate_recognition

(3) https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf

(4) https://ieeexplore.ieee.org/document/5169511

(5) https://medium.com/data-science-in-your-pocket/vehicle-number-plate-detection-and-ocr-tcs-humain-2019-a253019e52a1