License Plate Recognition using Python and OpenCV

Lad Harshkumar Sanjaykumar (1492280)

Troy University

Troy, AL.

laharsh726@gmail.com

***Abstract*-License plate Recognition system is a real time embedded system which automatically recognizes the license plate of vehicles. Nowadays, this application is helpful to recognize the license plate from complex security systems to common areas and from parking areas to urban traffic control. In the last decades vehicle license plate recognition systems are very useful for many traffic management and security systems such as automatic speed control, tracking stolen cars, automatic toll management, etc. There are several techniques and algorithms are available for license plate detection. Most of license plate recognition systems are built using proprietary tools like Matlab but it is too costly. So, instead of using Matlab, there are many other open source development tools are available in the market such as, Java, Python, or C/C++ and we can use OpenCV library for the computer vision, which is support computer vision development. This paper represents a alternative algorithm of recognizing license plate using Python and OpenCV.**

***Keywords*- License plate, Python, OCR, Grayscale, Threshold image, Python, OpenCV.**

1. Introduction

License Plate Recognition system is perform an important role in intelligent traffic control systems. This system can be use in many areas such as parking management system, border control and monitoring and tracking vehicles. There are many algorithms implemented for this system. Some of them are very accurate in the result with more complexity than others but some of are not working properly, not give such a good results and not work properly in some situation. Majority of License Plate Recognition systems are implemented in Matlab which is finically very expensive. In this paper we solve problems and drawbacks of other techniques and implement different algorithm than other using the Python and OpenCV technology.

In this paper, we discussed already implemented methods/algorithms in section 2, in section 3, described proposed algorithm and we described innovative part of proposed algorithm in section 4 than we compare it with other algorithm in section 5.We show the output of code by screen shots in section 6. Finally, we discussed conclusion and future work in the section 7.

2. Previous Methods

In the past, many algorithms have implemented for recognizing license plate such as, Gabor transform, Morphology, Edge-based & Color-aided license plate recognizing, OCR based license plate recognizing, License Plate Sizing and Orientation. Some of are very accurate but some of have many bugs and can’t give proper result.

In the past, recognizing license plate based on OCR (Optical Character Recognizing) was very accurate and it is still in many countries. In this algorithm, first it will take a image of vehicle as an input. Then the plate area is detected and extracted from the image. After successfully extracting the number, character segmentation is perform and recognize the number from license plate.

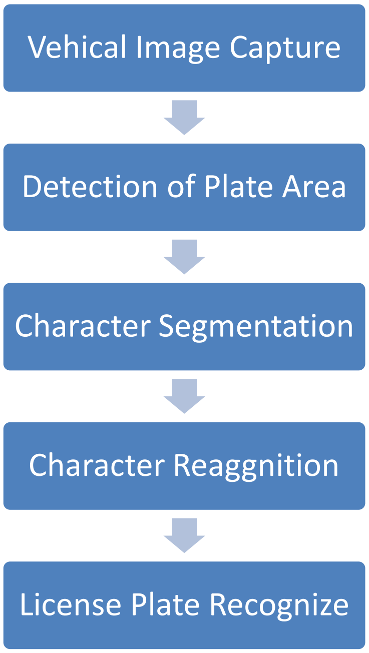


Figure 1. Algorithm of License Plate Recognition based on OCR (Optical Character Recognition)

Hough Transform algorithm was also implemented to recognized license plate. It is based on combination of Hough transform and contour algorithm. It looks for regions containing two parallel lines which consider as plate candidates. From the two horizontal lines of a candidate, it can calculate exactly how inclined the line was from horizontal coordinate. After that it applies a rotate transformation to adjust it to straight angle. After processed, these straight binary plate-candidate regions were passed to a number of heuristics and algorithms for evaluation. This algorithm is very accurate but it has some drawbacks like execution timing this technique may detect the headlights or windscreen falsely as license plate.

3. Proposed Algorithm

To overcome the problems of other algorithms we developed another algorithm that will work better than those algorithms. This algorithm is divided into two major parts: a) Find plates and b) Find characters within plate. In these both part, there will perform several operations to recognize the license plate. After took a image from digital camera, that image is go through these both part of algorithm and perform several steps to recognize license plate.

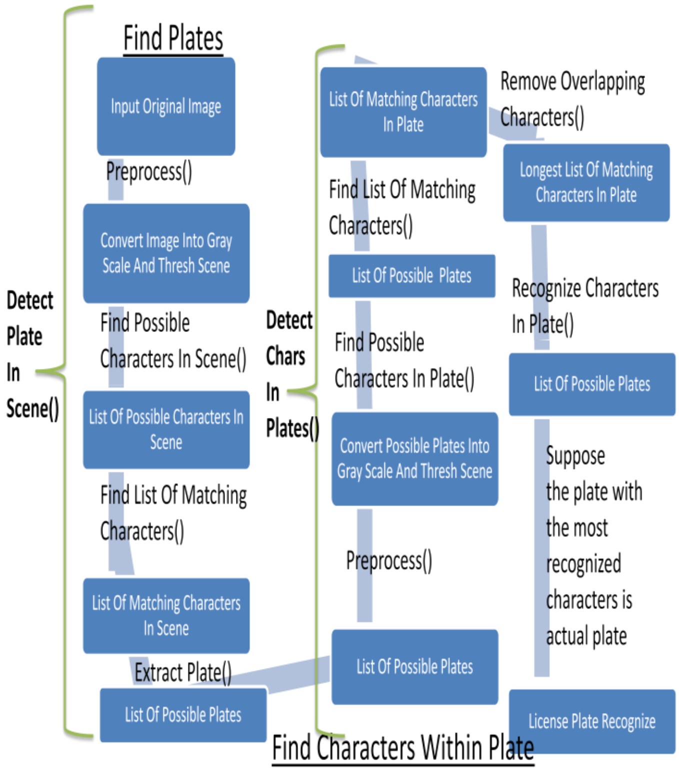

Find Plates
Input Original Image
Preprocess()
Convert Image Into Gray Scale And Thresh Scene
Find Possible Characters In Scene()
List Of Possible Characters In Scene
Find List Of Matching Characters()
List Of Matching Characters In Scene
Extract Plate()
List Of Possible Plates
Detect Plate In Scene()
Detect Chars In Plates()
List Of Matching Characters In Plate
Find List Of Matching Characters()
List Of Possible Plates
Find Possible Characters In Plate()
Convert Possible Plates Into Gray Scale And Thresh Scene
Preprocess()
List Of Possible Plates
Remove Overlapping Characters()
Longest List Of Matching Characters In Plate
Recognize Characters In Plate()
List Of Possible Plates
Suppose the plate with the most recognized characters is actual plate
License Plate Recognize
Find Characters Within Plate

Figure 2. Flow Chart of Algorithm

3.1 Find Plates

In this step, we detect possible plates from captured image. After took an image from digital camera, algorithm convert that image into gray scale image and threshold image. Than after it will find a possible characters from that and list all of the characters. Once it will get all the possible characters, than it will detect matching characters from the data by the fetching it. After that it will list all the matching characters from the image and extract all those possible plates from which we get possible characters and list all of those plates.

3.2 Find Characters within Plates

After get extracted plates our first step is finish. In the next part, it will convert those possible plates into gray style image and threshold image. Then it will search all possible characters just from those plates only and it will create a list of all matching characters from the plates. After that it will remove overlapping characters from the plate and again make long list of matching characters in the plates. Finally, it will fetch those characters in the training data and recognize the license plate.

4. Innovative Part

Innovative part of my algorithm is, after we convert original image into gray scale and threshold image, first it will detect all the possible number plates from the image than it will detect all the possible characters from those possible plates and remove overlapping characters from the plates. After that it will start comparing those number plates' characters with training data and recognize actual license plate.

4.1 Code of Innovative Part

```
if len(listOfPossiblePlates) == 0:
    print "\nno license plates were detected\n
  else:
listOfPossiblePlates.sort(key = lambda possiblePlate:
len(possiblePlate.strChars), reverse = True)
licPlate = listOfPossiblePlates[0]
    cv2.imshow("imgPlate", licPlate.imgPlate)
         cv2.imshow("imgThresh", licPlate.imgThresh)
    if len(licPlate.strChars) == 0:
         print "\nno characters were detected\n\n"
         return
end if
```

In this code, first it will check is there any possible plates or not. If it will detect any plate than it will go to next step, otherwise it will give a error message that there is not any possible plates. In the next step it will detect possible characters from those possible plates, and if it can't detect any characters than it terminates program and give a message that there is not any possible characters in the possible pates and if it found any character from the plates than it will recognize actual plate and display its number and also create separate image file in which indicate the plate number and with that car image.

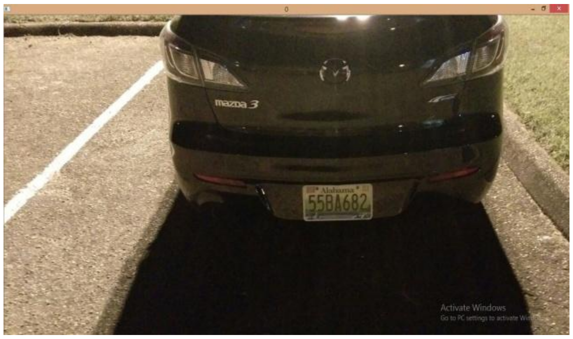
5. Proposed Algorithm vs. Previous Algorithms

Compare to some other algorithms, result of this algorithm is accurate and take less time to execute. Compare to OCR(Optical Character Recognition) algorithm the result of this algorithm is very accurate and this also take less time than OCR algorithm. In Hough Transform algorithm, first it's looking for two parallel lines which is consider as a license plate. The problem of Hough Transform algorithm is that it sometime detects car head lights or windscreen as a license plate. So, result will be false but in this algorithm possible plates is not detected based on its two parallel lines. So, result of this algorithm is more accurate than Hough Transform algorithm.

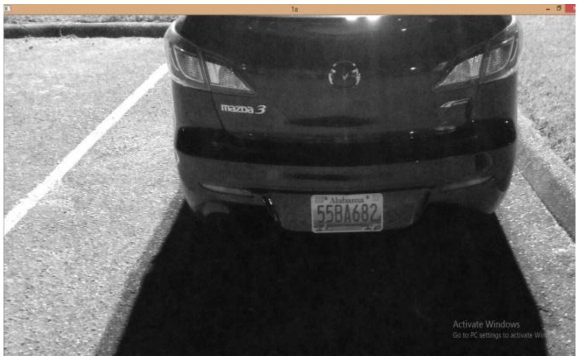
In Edge-based license plate detection, its recognize plate number based on the edge of characters of the plate. But in this algorithm because of it detect edge of the characters, sometimes it can't understand the difference between number zero (0) and alphabet 'O', but in this algorithm there is not occur this kind of problems.

Another algorithm

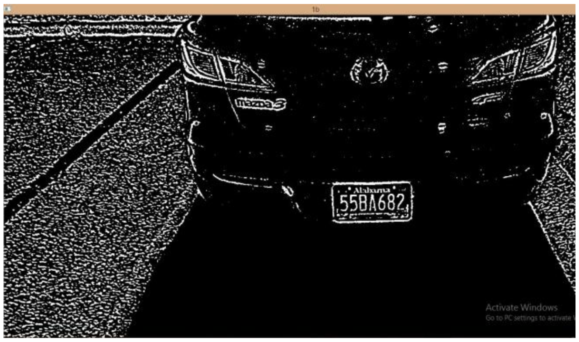
6. Simulation Results



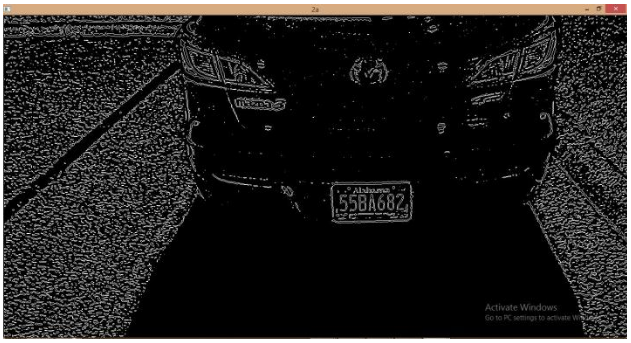
Original Image



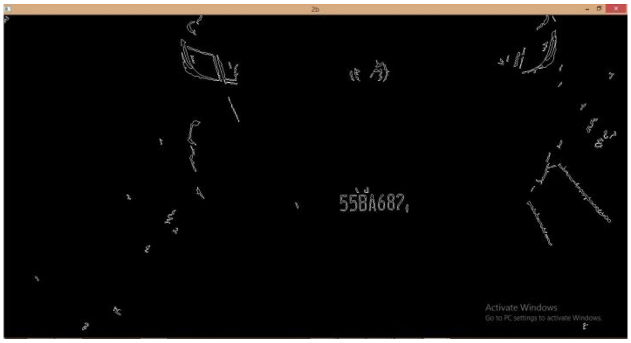
Grayscale Image



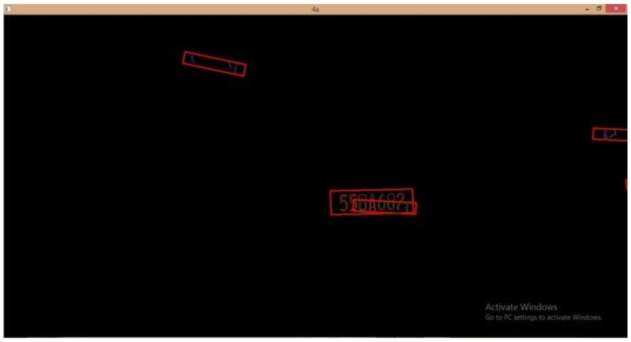
Threshold Image



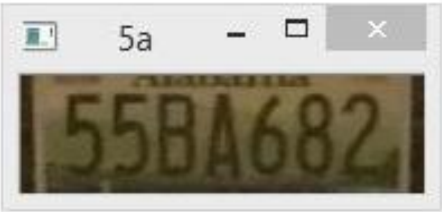
Find Possible Characters

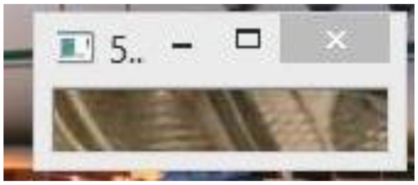


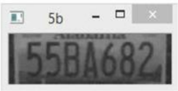
Vector of Matching Characters



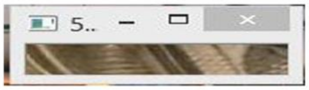
Vector of Possible Plates



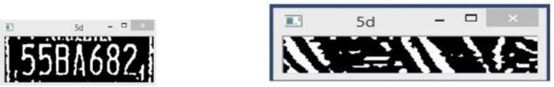




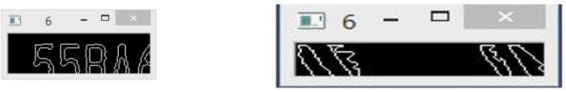
Grayscale Image



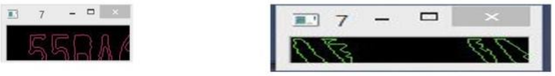
Threshold Image



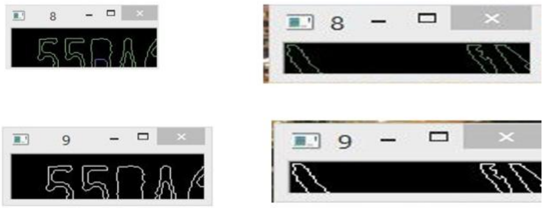
Possible Characters in Plate



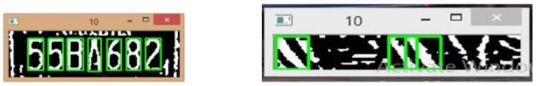
Matching Characters in Plate



Remove Overlapping Characters from Plate



Recognize Plate



Characters Found in Plate

7. Conclusion and Future Work

In conclusion, in this algorithm first we convert image into gray scale and thresh scene than detect possible plates from image and we recognize characters from possible plates. At the end, we get our actual plate with its number in the separate image.

In future we will recognize more than one license plate at a time and from the same image. And also from whatever angle we get the capture image, this algorithm will recognize correctly and give accurate result. Result will be accurate in any weather condition and weather we get bad image but it will recognize correctly.

Reference

[1] Duan, T. D, Hong, T. L., Hoang, T. V., “Building an Automatic Vehicle License Plate Recognition System”, International Conference on Computer Science, Research, Innovation, and Vision for the Future, 2005.

[2] Kahraman, F., Kurt, B., and Gokmen, M., “Licence Plate Character Segmentation Based on the Gabor Transform and Vector Quantization”, International Symposium on Computer and Information Sciences, 2003.

[3] Dlagnekov, L., “License Plate Detection Using AdaBoost”, La Jolla: Computer Science Engineering Department, University of California San Diego, March 2004 [4] Kang, D. J., “Dynamic Programming-based Method for Extraction of License Plate Numbers of Speeding Vehicles on the Highway”, International Journal of Automotive Technology, Vol. 10, No. 2, pp. 205–210, 2009.

[4] A. Conci, J. E. R. de Carvalho, T. W. Rauber, “ A Complete System for Vehicle Plate Localization, Segmentation and Recognition in Real Life Scene” , IEEE LATIN AMERICA TRANSACTIONS, VOL. 7, NO. 5,SEPTEMBER 2009.

[5] Ahmed Gull Liaqat, “Real Time Mobile License Plate Recognition System” IEEE White paper California, VOL.2 2011-12-05,Linnaeus University.

[6] Ondrej Martinsky (2007). "Algorithmic and mathematical principles of automatic number plate recognition systems" (PDF). Brno University of Technology. http://javaanpr.sourceforge.net/anpr.pdf.

[7] Bailey, D.G., Irecki, D., Lim, B.K., and Yang, L., “Test Bed for Number Plate Recognition Applications”, Proceeding of the IEEE International Workshop on Electronic Design, Test and Application, 2009

[8] Mahini, H., Kasaei, S., and Dorri, F., “An Efficient Features-based License Plate Localization Method”, Proceeding of IEEE International Conference on Pattern Recognition, 2006

[9] Kamat, V., and Ganesan, S., “An Efficient Implementation of the

Hough Transform for Detecting Vehicle License Plates Using DSP's", Real Time Technology and Applications Symposium, Page 58, 1995

[10] Parasuraman, K., and Kumar, P., "An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation", IEEE International Conference on Computational Intelligence and Computing Research, 2010

[11] Nguyen, C., Ardabilian, M., and Chen, L., "Unifying Approach for Fast License Plate Localization and SuperResolution", 2010**.**

[12] Broumandnia, A., and Fathy, M., "Application of Pattern Recognition for Farsi License Plate Recognition", International Journal of Graphics Vision and Image Processing, Vol. 5, pp. 25-31, 2005

[13] Abolghasemi, V., and Ahmadifar, A., "An edge-based color-aided method for license plate detection", Journal of Image and Vision Computing, Vol 27, pp. 1134-1142, 2009

[14] Lee, E.R., Kim, P.K., and Kim, H.J., "Automatic Recognition of a Car License Plate Using Color Image Processing", IEEE International Conference on Image Processing Austin, pp. 301-305, 1994

[15] Caner, H., Gecim, H.S., and Alkar, A.Z., "Efficient Embeded Neural-Network-Based License Plate Recognition System", IEEE Transactions on Vehicular Technology, Vol. 57, No. 5, pp. 2675-2683, 2008.