

License Plate Recognition from the given Static/Dynamic image of Vehicle/License Plate

Poojitha P¹, Prathima M N²

¹MTech Student, Dept of ISE, M S Ramaiah Institute of Technology, Karnataka, India

²Assistant Professor, Dept of ISE, M S Ramaiah Institute of Technology, Karnataka, India

Abstract - License plate Recognition system is the system which recognizes the license plate. Directly recognition of characters on the license plate cannot be done. The image should undergo several processes. Then only recognition of characters is done. Like as we know, in current world, population is raising. So that, the number of automobiles on the road also increasing. Because of this, lots of difficulties has been raised like difficult in handling traffic and also to know who breaks the rules, not knowing about the vehicles which has been stolen, difficult to know the vehicles parked in no parking place etc. because all the information should be collected manually, which is one of the difficult tasks as number of vehicles is more. In order to reduce the manual task, this VLPR system is designed. This system recognizes the license plate automatically by undergoing some steps like preprocess, detection of plate, preprocess of that detected plates, segmentation of characters and finally recognizing the characters present on the license plate approximately (might miss max of three characters, sometimes it can be recognized by adjusting the plate size of extraction and threshold image of license plate for proper character recognition). VLPR System try to recognize front view, rear view and also some slanted license plate. Recognizes irrespective of its back ground. But there are limitations like it recognizes single layered license plate, recognizes English number and alphabets only, recognizes all the countries license plates having license number which consists of only English alphabets and numbers.

Key Words: VLPR, License Plate Detection, Localization, Segmentation, License Plate Recognition, K-Nearest Neighbour (KNN)

1.OVERVIEW

License Plate Recognition system is the system which is the computer vision technology which identifies the number plates from the given static image/captured image without the involvement of human. Design of this type of systems is necessary because of increase in vehicles, also most of the image processing techniques has been developed. So, in order to make use of it and keeping the issue such type of systems is designing. Some of the applications are—to control crimes like vehicle stealing, vehicles breaking traffic rules, to know information of owner of particular car, to ease automatic toll collection.

But the design of such type of systems is not an easy task. Because during the design, it faces many challenges like number plate differences, illumination, background of license plate etc. Brightness also plays important role in detecting and recognizing number plates. Position of the number plate in the image, image quality, font style, color, inclination of the plates etc. all this has to be considered and the system has to designed appropriately and this is not an easy task.

In this project, images of vehicles are considered as input. Static as well as dynamic images can be used. Model is trained by making use of classification and flattened images text file. This text is the one which contains information of texts and vehicles images. But these classification and flattened images text file is in hexadecimal format. Converted to hexadecimal format using MATLAB. But the project is done without making use of MATLAB. So, direct images are not used for training purpose. This is the unique thing. Images required for input is taken from the google. Not considered the structured any vehicle dataset. After giving the input, the image will undergo preprocess step and plate will be localized. After that, plate detection is done. These possible plates undergo preprocess steps like converting image to greyscale, threshold image. Using threshold image, all the contours is found out and considered only characters contours and grouped them and then character recognition takes place.

After image undergoes preprocess streps, that is this preprocess activity includes collecting contours and selecting characters contours only and grouping them for plate detection. So, rectangle box will be drawn around the grouped characters and all the possible plates is obtained. These possible plates also undergo preprocess steps like it is converted to greyscale and threshold image. This threshold image is used for characters segmentation and character recognition. Characters on the plate is found out by same concept. That is collecting the contours, considering only character contours and grouping them and then character segmentation takes place.

2. RELATED WORK

In this paper, mainly explanation about the license plate detection and recognition is given. It is successful and most cost effective one. But the approaches used in this project differ according to the image quality, weather conditions etc. Mainly focused on identification of the license plate rather than recognition. But recognition of license plate is also included [1]. Aim is to identify multiple cars license plate from the single frame. But only license plate of cars is identified in this project.

In this paper, mainly explanation about the license plate identifying and recognizing using K-Means with CNN for ITS (Intelligent Transportation Systems) is given. Also Offered a different OKM-CNN [2] method for lively identifying and finding License plate of image only if image of vehicle is of front facing and the image should be clear enough with zoom format.

In this paper, mainly the project is aimed at identifying the number plate of the vehicle during the problematic condition like weather condition, distortion etc. The system built here captures the image of the vehicle from the video and detects the license plate using Faster R-CNN [3] and using optical character recognition, number recognition is done. But main focus is on identifying the number plate.

In this paper, main aim is to identify and recognize license plate of vehicles with having difficult background and if the license plate is tilted. According to them, available methods is not capable of doing these things. So, they are using end-to-end rectification network [4] based deep learning methods. Considered CCPD dataset.

In this paper, mainly project is aimed at identifying and recognizing license plate for parking system and generate tickets for that automatically based on the number plate. So that manual work is reduced. YOLO and ResNet model [5] used for identification and recognition purposes. Here YOLO used to detect the license plate and ResNet used to classify number plates.

In this paper, mainly explanation about the proposed license plate recognition system is given. In this project, Raspberry Pi is used. Camera is fixed into it and used for capturing license plate. Raspberry Pi processor is used to authenticate the user. Mainly used OpenCV and character recognition [6] instrument.

In this paper, mainly discussed about identifying and recognizing Bangla number plates which includes from different cities with multi-class as well. Most of the computer vision methods is used and YOLOv3 [7] model is used. Mainly Tesseract is used for character recognition, For training the model CNN is used to achieve maximum accuracy.

In this paper, mainly identification and recognition of number plate is done. But main aim is to identify and recognize Chinese number plate only. And there is restriction that they have fixed the location of the number plate and if it is present at that location only the system is able to identify properly. Otherwise, it fails. OpenCV [8] methods are used.

In this paper, mainly explanation about proposed recognition system is given. The images taken from the signal or from the parking place. This proposed system used in both outdoor and indoor place. Able to recognize whether it is sunny, dark, snow etc. Here for the detection and recognition, edge-based methods and texture-based methods [9] are used. Edge-based methods used for plate localization. Texture based methods used to recognize the characters.

In this paper, mainly discussed about the methods used for identifying the license plate without using the deep learning techniques. CT5L method [10] is the one which is used in this project for identification purpose. Otsu's method is used which gives good performance. Compared to other techniques, according to this paper, it gives best results and good performance.

3. SYSTEM PLAN

System plan is one the important step in order to implement the required system properly. Because system plan includes designing the architecture first so that implementation can be done according to that. Then based on the architecture, data flow diagrams can be done. From this we will get to know how data flow should takes place and what and all modules is required to build the required system. So, system plan mainly includes architecture and DFD.

3.1 SYSTEM CONSTRUCTION

The outline of Vehicle License Plate Recognition System is as shown in Figure 1: In this architecture, mainly first we need to give the input from the collected images or downloaded image or video as input. After giving the input, that is image

it undergoes pre-process step which involves converting given image to greyscale and then to threshold image. From this threshold image we will find all the possible contours and from these contours we consider only characters contours and grouped to get all the possible plates. So, this is the process of plate detection. After this, all the possible plates undergo same pre-process steps as image undergone. Then, from the threshold plate, contours are selected and from those only characters are considered and segmented. Finally, character recognition is done.

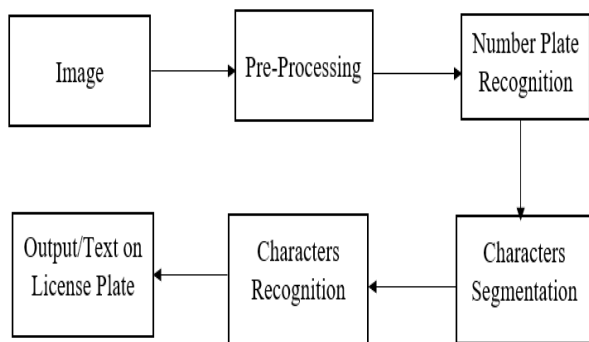


Fig 1: System Architecture

3.2 DATAFLOW DIAGRAM

The DFD of Vehicle License Plate Recognition System is explained as shown in Figure 2. Mainly includes vehicle image which is given as input. Then image undergoes preprocess steps. From that we get all the possible plates. These plates undergo preprocess steps and characters are segmented. Finally, character recognition is done using OpenCV functions. Also, we can get information about the vehicle owner based on the vehicle image given. This is an extra factor which is added to the system.

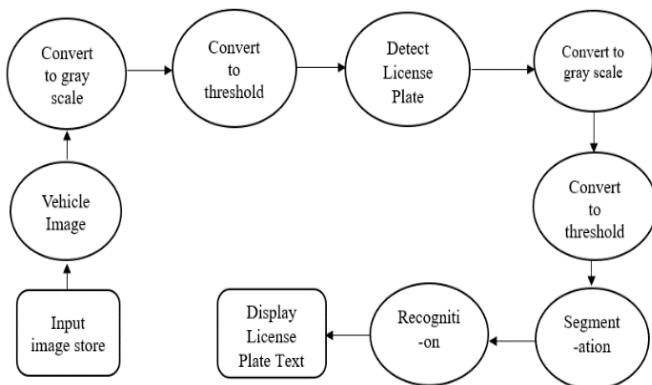


Fig 2: Data Flow Diagram

4. OPERATION

The implementation of a system is done after the development effort is completed. Before operating actual system that is coding, we need to collect and select the requirements required for implementation which includes data, software, coding language, libraries, some of the required functions, system setup etc. So, if all these requirements are properly collected and selected, then in order to implement the required system becomes easy and in a smooth way the process goes and get the required or satisfactory result.

4.1 STEPS

After planning the things to be done, before implementing it, we need to collect and select some of the basic requirements which is needed for the implementation. Without having proper resources, we cannot build the system which is required. So, these steps play an important role.

Collect the images of vehicles/video stream to use for input

Collecting the still images/video stream which will be given as input. This is the basic requirement which has to be done. As system takes still images as well as video stream as input it is challenging task. Taking video stream as input is the challenging task because system has to capture the image and then required process takes place.

Selecting required and comfortable programming language

Here mainly, making use of Python language for the coding purpose. Python is a strangely commanding lively, object-oriented programming language that is used in an extensive diversity of tender fields. Its proposals robust provision for mixing with other idioms and tools, and comes with widespread typical lending library.

For detection and recognition of license plate, tool should be selected (OpenCV)

Here mainly computer vision techniques are used for detecting and recognizing the license plate. Most of the cv2 functions is used in the implementation. OpenCV is a lending library of programming purposes for actual period computer vision initially industrialized by Intel and now maintained by Willoughby Garage. OpenCV is the best open-source computer vision lending library that creators and scholars can ponder of.

To implement the design part, select proper software

Proper software has to be selected which supports the work which will be done. Here PyCharm is the software which is collected and it is the software program which delivers keen cipher achievement, code reviews, on-the-fly error importance and quick-fixes, along with automatic code and rich triangulation competences. PyCharm is intended by computer programmer, for program writer, to deliver all the tackles we need for creative Python growth. All the lending library required for application of code is fixed in PyCharm software itself.

4.2 PROCEDURES

Mainly the planned system has quatern units. They are-

- Plate identification
- Plate Withdrawal(extraction)
- Segmentation
- Character Recognition

Plate Identification

In this planning, firstly we will take the still image/video stream as input and given to system designed. System captures the image and used for further process if video stream is given as input. After taking the input, system do preprocess steps on the given input. It involves converting the image into greyscale and then greyscale to threshold image. After that from the threshold image, all the contours will be found out and considers only characters from that. All the similar characters are grouped and red rectangle is drawn. From this process, all the possible is identified.

Sample code for identification:

```
image= cv2.imread("image label")

if image is None:

    print ("\n error: image not read from file \n\n")

    PossiblePlates=DetectPlates.detectPlates(image)

    GrayscaleImage, ThreshImage=Preprocess.
preprocess(image)

PossibleCharsInScene=findPossibleCharsInScene
(ThreshImage)
```

```
ListsOfMatchingCharsInScene=DetectChars.findListOfLists
OfMatchingChars(PossibleCharsInScene)
```

Plate Withdrawal (extraction)

After identifying all the possible plates, next step is to withdraw all the possible plates for further process which involves segmentation and character recognition. So, this the primary step and crucial one because this is the basic step which should be proper enough to recognise characters properly.

Sample code to extract plate:

```
for MatchingChars in ListsOfMatchingChars:
```

```
possiblePlate = extractPlate (image, MatchingChars)
```

Then,

Calculation of centre point of the plate, calculation of plate width and height, calculation of correction angle of plate region, then get the rotation matrix for our calculated correction angle, copy the cropped plate image into the applicable member variable of the possible plate.

Segmentation

After obtaining all the possible plates, next preprocess of possible plates takes place. All the possible plates converted to greyscale and then to threshold image. Then contours in from threshold image is collected and considered only characters contours. Using this character's contours, segmentation of characters takes place.

Sample code for segmentation:

```
possiblePlate.imgGrayscale,
possiblePlates.imgThresh=Preprocess.preprocess(possibl
ePlates.imgPlate)

for Char in MatchingChars:

    pt1 = (Char.intBoundingRectX, Char.intBoundingRectY)

    pt2 = ((Char.intBoundingRectX +
Char.intBoundingRectWidth), (Char.intBoundingRectY +
Char.intBoundingRectHeight))

    cv2.rectangle(imgThreshColor, pt1, pt2,
Main.SCALAR_GREEN, 2)

    # Draw green box around the char
```

Character Recognition

After the segmentation of characters, final step is to recognize the characters. Mainly here KNN algorithm is and this model is trained by classification txt and flatten images text. Using this algorithm and some of the OpenCV functions characters which are segmented is recognised approximately. Sometimes fails to recognize characters properly because of size of plates or characters or breakage in the characters. But recognition rate of characters is more than of failure rate.

Sample code for character recognition:

```
possiblePlate.stringChars=recognizeChars  
(possiblePlate.imgThresh, ListOfMatchingCharsInPlate)  
  
# Finally, we call findNearest  
  
returnvalue, npaResults, neigh_resp, dists =  
kNearest.findNearest(npaRegionOfInterestResized, k = 1)  
stringChar = str (chr (int (npaResults [0] [ 0])))  
  
# From the result char is obtained  
  
stringChars = stringChars + stringCurrentChar  
# add current char to full string
```

5. ESTIMATION MEASURES

In this project, mainly the given image should pass four phases like preprocess step, plate detection and extraction, segmentation and finally character recognition. So, first step is preprocess step. For evaluating this phase, considered some 150 vehicle images. Mainly checking how the given image is converted to greyscale and then to threshold image. Also focused on success rate of this process. Then the next step after undergoing preprocess step is plate identification. In order to identify all the possible plates, using threshold image, all the contours are found out. In order to evaluate the whole process of plate detection and extraction, considered 150 images. So, all the contours will be found out and in that only character's contours is considered. After this similar character contours are grouped and red rectangle is drawn to get possible plates and finally all these possible plates is considered for segmentation process. All this process has to be evaluated and considered the overall result.

Next is the segmentation process. For this also considered some of the license plates which already undergone preprocess step of given image and plate identification steps.

So, here all the plates undergo preprocess steps that is converting all the plates to greyscale and threshold and all the contours in threshold image is found out and grouped character contours only. Then the segmentation process takes place. Evaluated segmentation process in this way by considering the pre-processed license plate. At last, character recognition has to be done. Using some of the segmented characters license plate, character recognition is done in order to evaluate the system like how much characters has been recognized properly. From this process, number of recognising rate of all the characters is more than that of failure rate. Results of the evaluation measures is given in the table 1.

Phases	Success rate
Greyscale image	100%
Threshold image	100%
Plate detection and extraction	98%
Segmentation	97%
Character Recognition	93%

Table 1: Analysis table

6. OUTCOMES AND DISCUSSIONS

This explains about the results obtained after the execution of the intended system. This comprises all the screenshots got as outcome which contains the screenshot of license plate identification, plate withdrawal(extraction), segmentation of the characters and recognition of characters. Mainly here for performance analysis graphs/plots are not used for knowing the accuracy of the system. Instead of this, took some of the images and used for implementation and note down the passed results and failed results of pre-process, plate detection, segmentation and recognition steps.

After note down the passed and failed results, it is converted to percentage form and given in the above table.

In this section, all the stages results is given which includes pre-process step, Plate detection, segmentation and character recognition.

Preprocess Step:



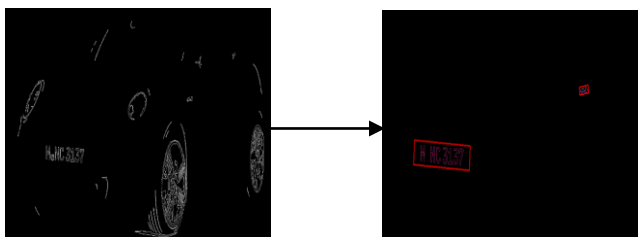
Input image given

Greyscale image



Threshold image

Plate Identification:



Character Contours

Possible plates

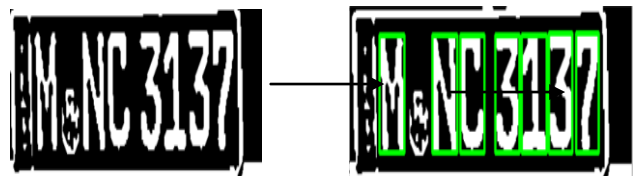
Plate Withdrawal(extraction):



False license plate

Correct license plate

Segmentation:



Threshold plate

Segmented characters

Recognition:



Result/Text on license plate

Information Retrieval:

```

Number Plate(Original)      LJ465H
Color                        Maroon
Owner                        David
Email                        david@gmail.com
Place                         Jamestown
Name: 17.jpg, dtype: object
    
```

3 possible plates found

license plate read from image = LJ465H

Sample one: Information obtained about owner

7. CONCLUSION AND FUTURE WORK

In this project, VLPR system from the given captured/still image of vehicle/License plate is planned and executed. This project mainly attentive on identifying all the countries license plate which is of having English numbers and alphabets only using KNN model. So, this is not restricted to only Indian license plate recognition. And also attentive on recognizing all the conceivable understandings of vehicle like front view, rear view, from side view, little bit blur, in dark, in sunny. Also, the finishing time is also measured. This scheme provides output fast compared to other built systems.

Designed structure can be enhanced further by using diverse algorithm for training the system, preprocess step so that all the characters on the license plate will be recognized properly. Also, can build the algorithm which recognizes two layered license plates. Also, can be improved to recognize license plate text other than English text and letter like Chinese license plate, kannada alphabets and numbers license plate etc.

REFERENCES

- 1] Farheen Ali, Himanshu Rathor, Wasim Akram, "License Plate Recognition System", International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021.
- 2] Irina Valeryevna Pustokhina, Denis Alexandrovich Pustokhin, Joel J. P. C. Rodrigue, Deepak Gupta, Ashish Khanna, K. Shankar, IEEE Member, Changho Seo, Gyanendra Prasad Joshi, "Automatic Vehicle License Plate Recognition using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems", IEEE Journal, 2020.
- 3] N.Palanivel Ap, T. Vigneshwaran, M.Sriv Arappadhan, R. Madhanraj, "Automatic Number Plate Detection in Vehicles using Faster R-CNN", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.
- 4] Zhuang Liu, Yuanping Zhu, "Vehicle License Plate Recognition in Complex Scenes", IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE), 2020.
- 5] Joshua, Janson Hendryli, Dyah E. Herwindiati, "Automatic License Plate Recognition for Parking System using Convolutional Neural Networks", International Conference on Information Management and Technology (ICIMTech), 2020.
- 6] Agbeyangi A. O, Alashiri O. A, and Otunuga, "Automatic Identification of Vehicle Plate Number using Raspberry Pi", International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 2020.
- 7] Md. Naimul Islam Suvon, Riasat Khan, Mehebuba Ferdous, "Real Time Bangla Number Plate Recognition using Computer Vision and Convolutional Neural Network", IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (ICAET), 2020.
- 8] Chao Xu, Haifei Zhang, Wei Wang, Jianlin Qiu, "License Plate Recognition System Based on Deep Learning", IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2020.
- 9] Chowdhury D, Mandal S, Das D, Banerjee S, Shome S, and Choudhary D, "An Adaptive Technique for Computer Vision Based Vehicles License Plate Detection System", International Conference on Opto-Electronics and Applied Optics (Optronix), 2019.
- 10] Ohzeki K, Geigis M and Schneider S, "License Plate Detection with Machine Learning Without Using Number Recognition", Proceedings of the Federated Conference on Computer Science and Information Systems, 2019.