TCS HumAIn – 2019

Powered by MFDM

Vehicle Number Plate detection

Identify the license place in the image and do an OCR to extract the characters from the detected license plate.

Category: Object detection & OCR

DATASET:

vehicle-number-pla
te-detection Dataset

Solution Submission

CT/DT Number: DT20184774185
Contestant Name: Krishnaa S
College Name: Nandha Engineering College, Erode.

Contents

1. Background

Vehicle Number Plate detection is to identify the license plate in the image is designed to recognize number from license plates from the front and rear of the automobile. Input to the system is an image sequence acquired by a digital camera that consists of a license plate and its output is the recognition of characters on the license plate. The system consists of the standard four main modules in the Vehicle Number Plate Recognitions and Extraction. Image acquisition, Number plate extraction, Number plate segmentation and Number plate recognition.

2. Your Understanding

Vehicle number plate detection is an image processing technology which uses number plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the owner, place of registration, address, etc.

3. Scope

Security is big issue in highly restricted area as well as in public places. Here the Vehicle number plate detection will be helpful in reducing crime and fear of crime, it reduces vehicle crime and traffic related nuisance.

The detection and reading of license plates is a kind of intelligent system and it is considerable in different aspects which include

- **Parking** - the plate number is used to automatically enter pre-paid members and calculate parking fee for non-members by comparing the exit and entry times.
- **Access Control** - a gate automatically opens for authorized members in a secured area, thus replacing or assisting the security guard. The events are logged on a database and could be used to search the history of events .
- **Law enforcement** - the plate number is used to produce a violation fine on speed or red-light systems. The manual process of preparing a violation fine is replaced by an automated process which reduces the overhead and turnaround time.
- **Marketing Tool** - the car plates may be used to compile a list of frequent visitors for marketing purposes, or to build traffic profiles, such as the frequency of entry verses the hour or day.
- **Tolling** - the car number is used to calculate the travel fee in a toll-road, or used to double-check the ticket.

4. Out of Scope

The Vehicle number plate detection follows the method of Optical Character Recognition so we can also able to implement and extend this feature in various fields.

- Automating data entry, extraction and processing.
- Deciphering documents into text that can be read aloud to visually-impaired or blind users.
- Archiving historic information, such as newspapers, magazines or phonebooks, into searchable formats.
- Electronically depositing checks without the need for a bank teller.
- Placing important, signed legal documents into an electronic database.
- Recognizing text, such as license plates, with a camera or software.
- Sorting letters for mail delivery.
- Translating words within an image into a specified language.

5. Assumptions

General Assumption

The purpose of this project was to design and implement an Vehicle Number Plate Detection system. The system has still images as the input, and extracts a string corresponding to the plate number, which is used to obtain the output user data from a suitable database. The system extracts data from a license plate and automatically reads it with no prior assumption of background made. License plate extraction is based on plate features, such as texture, and all characters segmented from the plate are passed individually to a character recognition stage for reading. The string output is then used to query a relational database to obtain the desired user data. This particular project utilized the intersection of a hat filtered image and a texture mask as the means of locating the number plate within the image.

The most common solutions to license plate localization in digital images are through the implementation of edge density estimation, color based segmentation, texture filtering and histogram analysis, morphological operators and Hough Transform. An edge approach is normally simple and fast. However, it is sensitive towards noise. Hough transform for line 7 detection gives positive effects on image assuming that the plate is made up of straight lines. However, it requires the outline of the plate to be obvious for satisfactory license plate localization, large memory space and a considerable amount of computing time. On the other hand, a basic histogram approach is not capable of dealing with images with considerable amount of noise and tilted license plates. Last but not least, the localization of license plates via morphological based approaches is not susceptible to noise but is slow in execution. A combination of two or more methods or modification of these methods is thus necessary. Localization of the license plate comes after the character segmentation process. Common character segmentation processes are based on histogram analysis and estimation of bounding boxes of characters.

Technical Assumption

If you scan a document into your PC and then open it in a word processor you cannot edit or alter it in any way. This is because it is simply one bitmap made up of thousands of individual pixels. However there is software available, frequently a freebie with scanners that can convert these groups of pixels into characters. This is Optical Character Recognition (OCR), which scans each group of pixels and estimates whether or not it could be a letter and replaces the pixels with the ASCII code for the letter. For instance the ASCII code for the lower case 'a' is 01100001. So, the software scans the whole document and produces a page of letters exactly the same as though you had typed them in, which can be edited or manipulated in any way.

OCR is the fundamental technology used in ANPR and provides the capability to store and sort data. ANPR cameras need to be a special type and set up within certain important parameters as will be described later. As a vehicle approaches the camera the software takes a series of 'snapshots' and stores them in a file. When the number plate is of sufficient size for the OCR software the frame is scanned and the registration number is converted to ASCII code and held in a list. This continues for a series of images according to the speed and position of the vehicle. The list is scanned for similarities and a 'favourite' selected to retain. The system would typically scan and compare 10-15 images, with 5 being considered the minimum for high accuracy.

This then, is the start of the ANPR capture and is totally dependent on the correct set up of camera, lens, illumination, angle of view and configuration. Get one wrong and you have a disappointed customer who won't pay the bill. At this stage we are concentrating on the number plate capture but there are many other aspects to be considered for a completely integrated system, which will be discussed later. Note that the ANPR capture considered here is monochrome.

6. Solution Approach

High Level Solution Approach

Vehicle Number Plate Detection is a process where vehicles are identified or recognized using their number plate or license plate. Vehicle Number Plate Detection uses image processing techniques so as to extract the vehicle number plate from digital images. Vehicle Number Plate Detection systems normally comprises of two components: A camera that used in capturing of vehicle number plate images, and software that extracts the number plates from the captured images by using a character recognition tool that allows for pixels to be translated into numerical readable characters. A license plate recognition system generally works in four main parts namely image acquisition, license plate detection, characters segmentation, and lastly character recognition.

1. **Image Acquisition**

The first step is the image acquisition stage. The image of the vehicle is captured using a photographic camera. The constraint is that the image of the vehicle should be captured in such a way that the selected input image contains rear or front view of the vehicle with the number plate.

The captured image is affected by many factors like: optical system, distortion, system noise, lack of exposure or excessive relative motion of camera or vehicle thus resulting in a degradation of a captured vehicle image hence adversely affecting the results of the overall image processing. As a correction mechanism, an image pre-processing stage is introduced to take care of any errors that may have occurred during the image acquisition stage. Image pre-processing mainly involves converting the RGB image into grey color, noise removal, and border enhancement for brightness. Image pre–processing is usually done through image filtering.

2. **Number Plate Detection**

The next stage that follows is the number plate recognition phase that does several functions such as resizing of the image to a feasible aspect ratio.Number plate detection searches an input image in order to identify specific features that contain the number plate. The number plate can be found anywhere within an image, it is impractical to check all the pixels of the image in order to locate the number plate. Therefore, we only focus on those pixels that have the number plate.

3. Character Segmentation

Character segmentation can be defined as a technique, which partitions images of lines or words into individual characters. It is an operation that seeks to decompose an image of a sequence of character into sub images of individual symbols. Character segmentation is an operation that seeks to decompose an image of a sequence of characters into sub-images of individual symbols.

Character segmentation is the process through which the text component within an image is isolated from the background. In order for proper text recognition to take place the line of text is first segmented, then from the segmented line the words are segmented and then from that the characters are segmented.

4. **Character Recognition**

Character recognition is process of detecting and recognizing characters from input image and converting it into meaningful text in ASCII (American Standard Code for Information Interchange) or other equivalent machine editable form. Character recognition is the process to classify the input character according to the predefined character class.

Step1: The camera takes a picture of the vehicle containing the number plate (Image acquisition).

Step2: The camera isolates the plate, adjusts the brightness and contrast and segments it into characters (Number plate detection and Character segmentation).

Step3: The pattern of each character is analyzed to convert the picture into text (Character recognition).

Use-Case Diagram:

Actor:

- User

Use Case:

- Capture image C1
- Verify Vehicle
- Identify Number

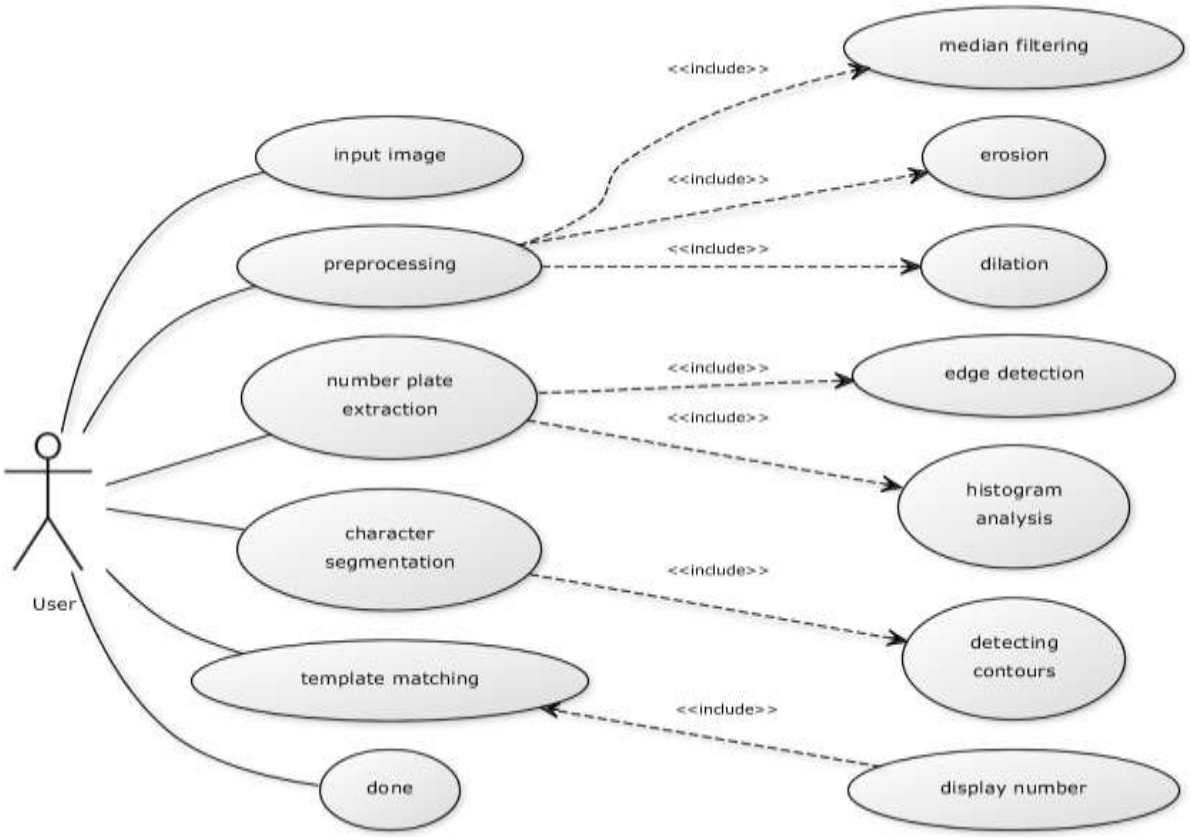

Figure 1(Use Case Diagram)

Class Diagram:

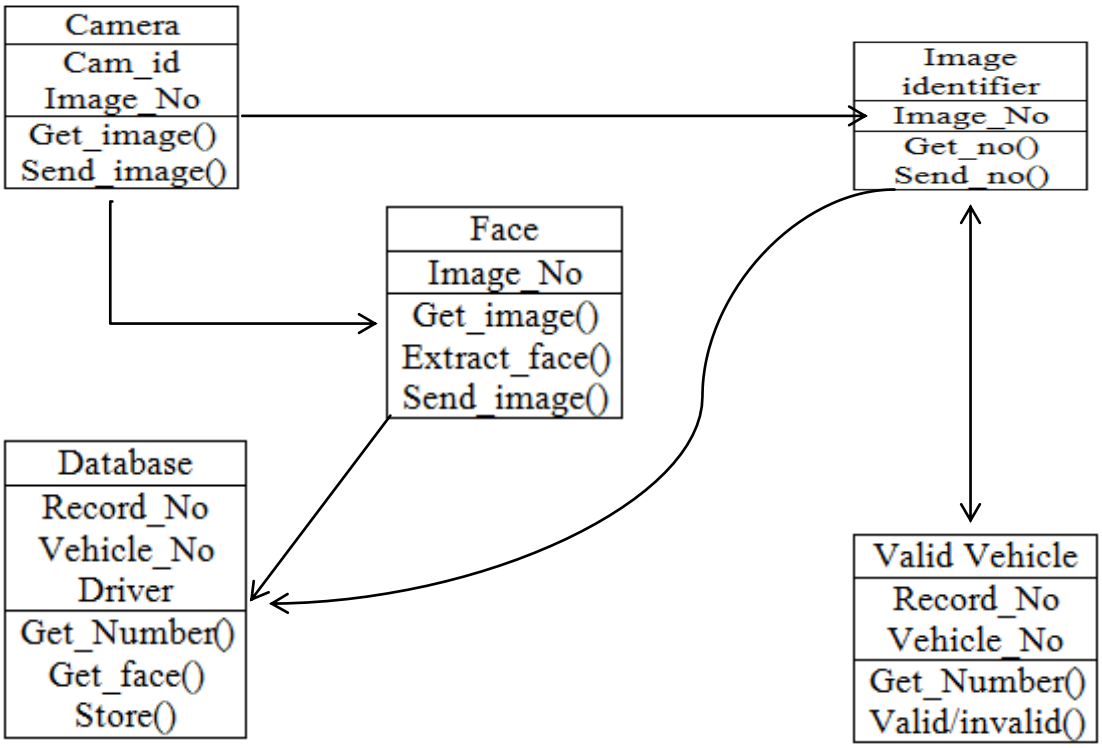

Figure 2

Sequence Diagram:

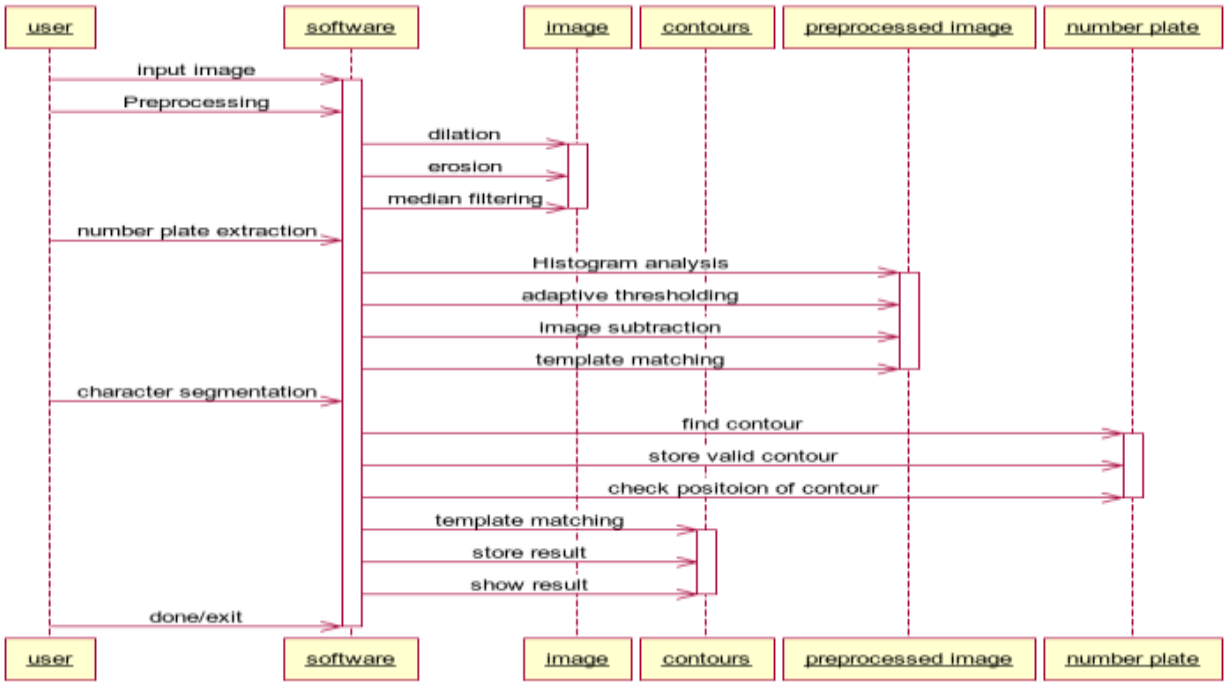

Figure 3

Activity Diagram:

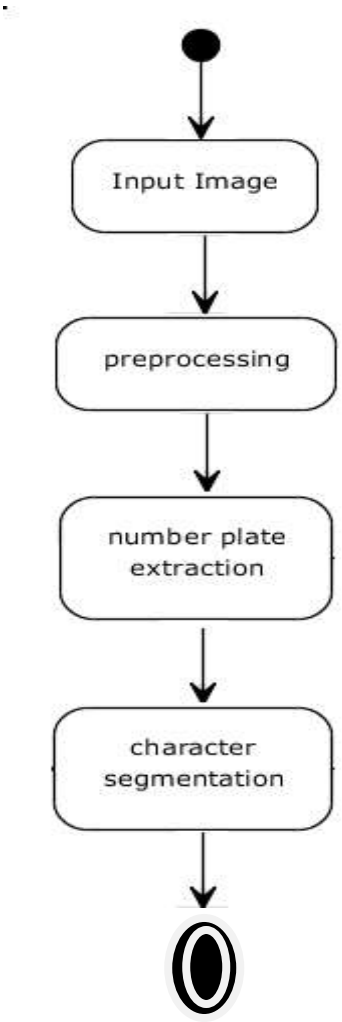

Figure 4(Activity Diagram)

7. Implementation Framework

To implement this project approach, proceed with the following steps:

1. Install Tesseract OCR in your machine from the given url https://github.com/tesseract-ocr/tesseract/wiki/Downloads and restart the system.

2. Import -numpy: Numpy library to generate the coordinates of the image. -pandas: Panda library to read JSON. -PIL: To make the changes in the images. -pytesseract: To extract the characters from the number plate. -urllib: To visit the url available in the JSON. -cv2: To read images and make changes. -json: To read JSON.

3. Put both the files python code and the given dataset in the same directory, after importing all the libraries.

4. Run the code, it will generate the result for the given dataset.

5. Exit.

Hardware Requirements:

- 1.6 GHz or faster processor.
- 1024 MB RAM (1.5 GB if running on a virtual machine).
- 3 GB of available hard-disk space.
- 5400 RPM hard-disk drive.
- DirectX 9-capable video card running at 1024 x 768 or higher display resolution.
- DVD-ROM drive.

Software Requirements:

- Python 3.7.4.
- Tesseract OCR.
- Linux.

Operating System:

- Linux.
- Windows Vista and newer for Python 3.7, Windows XP and newer for Python 2.7.
- FreeBSD 10 and newer.
- MacOS Leopard (MacOS 10.6, 2008) and newer.

8. Solution Submission

GitHub link:

https://github.com/Krishnaakrish/Vehicle_liscence_plate_detection/blob/master/Vehicle_number_plate_detection(OCR).py

9. Appendix

FUTURE SCOPE

The process of Automobile Number Plate Recognition requires a very high degree of accuracy when we are working on a very busy road or parking which may not be possible manually as a human being tends to get fatigued due to monotonous nature of the job and they cannot keep track of the vehicles when there are multiple vehicles are passing in a very short time. To overcome this problem, an effort can be made by the researchers across the globe by taking the systems image input using online hardware and specialised cameras installed at appropriate location where the automobile number plate recognition system is implemented. Though we have achieved an accuracy of 98% by optimizing various parameters, it is required that for the task as sensitive as tracking stolen vehicles and monitoring vehicles for homeland security an accuracy of 100% cannot be compromised with. Therefore to achieve this, further optimization is required. Also, the issues like stains, smudges, blurred regions & different font style and sizes are need to be taken care of. This work can be further extended to minimize the errors due to image capturing.

10. References

R. Gonzalez, R .Woods, Digital Image Processing, Prentice Hall, New Jersey, 2002, Operations on an image.

https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/.