CHAPTER 1

Introduction

* 1. Background

While the first industrial automatic system for Car License Plate Recognition (LPR) was introduced in the 80’s, an outburst of commercial systems occurred in the 90s. Although that a lot of LPR systems are available in the market, the research and development still continues and new sophisticated solutions to plate localization, character segmentation and recognition appear. Vehicle’s license plate recognition system has been a special area of interest in video surveillance area for more than a decade or so. With the advent of sophisticated video vehicle detection systems for traffic management applications, number plate recognition system finds wide varieties of places to fit itself beyond just controlling access to a toll collection point or parking lot.

The whole system into three following steps:

1. Plate location or finding location of plate in the vehicle image and cropping plate image from it.
2. Plate segmentation or cutting plate image to character’s images.
3. Character recognition or convert character’s images to final distinguished characters among them.
   1. Relevance

The project consists application of subjects namely Digital Image Processing and Artificial Intelligence. Digital Image Processing was useful in pre and post processing of the image, while Neural Networks is a part of Artificial Intelligence.

* 1. Literature Survey

The LPR technique proposed by Shyang-Lih Chang et. al [4] consists of two main modules: a license plate locating module and a license number identification module. The algorithm proposed the use of two state fuzzy aggregation. It included Fuzzy Disciplines for license plate location and Neural Networks for license plate number identification. The former characterized by fuzzy disciplines attempted to extract license plates from an input image, while the latter conceptualized in terms of neural subjects’ aims to identify the number present in a license plate. In the conducted experiment the license plate location rate of success was found to be 97.9% and the identification rate of success was 95.6%. Combining the above two rates, the overall rate of success for the LPR algorithm was 93.7%.

Christos-Nikolaos E. Anagnostopoulos et. al [3] carried out a survey experiment which comprised of following three steps: 1) Extraction of License Plate Region. 2) License Plate Segmentation. 3) Character Recognition. It presented a comprehensive and critical survey of up-to-date LPR methods. The survey provided a detailed review of techniques to detect license plates in a single image or video sequence. Character segmentation methods and criteria were discussed and it also demonstrated the character classiﬁcation techniques. The paper was concluded with a discussion of current trends and anticipated research in LPR.

Shan Du et. al [2] presented a comprehensive survey on existing ALPR techniques by categorizing them according to the features used in each stage. The features included the use of boundary information, global image information, texture features, colour features, character features and combining two or more features for license plate extraction. For segmentation, the suggested methods included using pixel connectivity, projection profile, knowledge of characters, using character contour and combined features. Character recognition was suggested using raw data and using extracted features. Comparisons of these in terms of pros, cons, recognition results, and processing speed were addressed. A future forecast for ALPR was mentioned. It was proposed that the future research of ALPR should concentrate on multi-style plate recognition, video-based ALPR using temporal information, multi-plates processing, high deﬁnition plate image processing and ambiguous-character recognition.

Christos Nikolaos E. Anagnostopoulos et. al [5] analyses a new algorithm for vehicle license plate identiﬁcation on the basis of a novel adaptive image segmentation technique (sliding concentric windows) and connected component analysis in conjunction with a character recognition neural network. The algorithm was tested with several natural-scene gray-level vehicle images of different backgrounds and ambient illumination. The camera was focused in the plate, while the angle of view and the distance from the vehicle was varied according to the experimental setup. The efficiency of license plates which were properly segmented was 96.5%. The optical character recognition system is a two-layer probabilistic neural network whose performance for entire plate recognition reached 89.1%. The PNN was trained to identify alphanumeric characters from car license plates based on data obtained from algorithmic image processing. Combining the above two rates, the overall rate of success for the license plate-recognition algorithm was 86.0%.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No | Title | Year | Author | Algorithm | Efficiency |
| 1 | Automatic License Plate Recognition. | March 2004 | Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen | Two state fuzzy aggregation:  License Plate Location: Fuzzy Disciplines.  License Plate Number Identification: Neural Networks. | Plate location (97.9%)  Identification (95.6%)  Combined (93.7%) |
| 2 | A License Plate Recognition Algorithm for Intelligent Transportation System Applications. | Sept 2006 | Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Vassili Loumos, and Eleftherios Kayafas | Probabilistic Neural Network | Segmentation (96.5%)  Recognition (89.1%)  Combined (86.0%)  Better Conditions (90-95%) |

Table 1.1 IEEE papers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | Title | Year | Author | Contents |
| 1 | LPR from still images and video sequences: A survey | Sept 2008 | Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas | It mentioned 3 steps for LPR:  Extraction of License Plate Region  License Plate Segmentation  Optical Character Recognition |
| 2 | ALPR: A State of Art Review | Feb 2013 | Shan Du*,* Mahmoud Ibrahim, Mohamed Shehata*,* and Wael Badawy | Classification of the basic steps of ALPR as mentioned below was included.  License Plate Extraction:  Using boundary/edge information  Using global image information  Using texture features  Using colour features  Using character features  Combining two or more features  License Plate Segmentation:  Using pixel connectivity  Using projection profile  Using prior knowledge of characters  Using character contour  Using combined features  Character Recognition  Using raw data  Using extracted features |

Table 1.2 IEEE survey

* 1. Motivation

The complexity of smart license number plate recognition work varies throughout the world. For the standard number plate, ALPR system is easier to read and recognize. In India this task becomes much difficult due to variation in plate model and their size. Character recognition part is also very difficult in Indian number plate. So flexible algorithm is required for solving this task.

Nowadays vehicles play vital role in transportation. Also the use of vehicles has been increasing because of population growth and human needs in recent years. Therefore, control of vehicles is becoming a big problem and much more difficult to solve. License Plate Recognition systems are used for the purpose of effective control.

* 1. Problem Definition

The aim of the project is to perform character recognition on a number plate and take the help of Neural Networks algorithm for updating the database. Presently, we have pre-processed the image and started with the segmentation process.

* 1. Scope and Objectives

The code has higher efficiency for images with good resolution and proper viewing angle. It is currently implemented for static data base i.e. images. The objective of the project is to accurately recognize, fully automate and self train the process of character recognition for license plate detection

* 1. Technical Approach

The input image is pre-processed and the license plate has been extracted. With the help of different kind of structural elements various kinds of noise were identified and eliminated. For the purpose of boundary detection, operations like dilation and erosion has been used. The efficiency of character recognition is to be improved by using neural networks. The precision of data base or function for character recognition will be improved after every iteration of code using neural networks.

* 1. Organization of Report

Contents of each chapter and report organization is to be included briefly.

CHAPTER 2

Block Diagram and Theory

2.1 Block Diagram

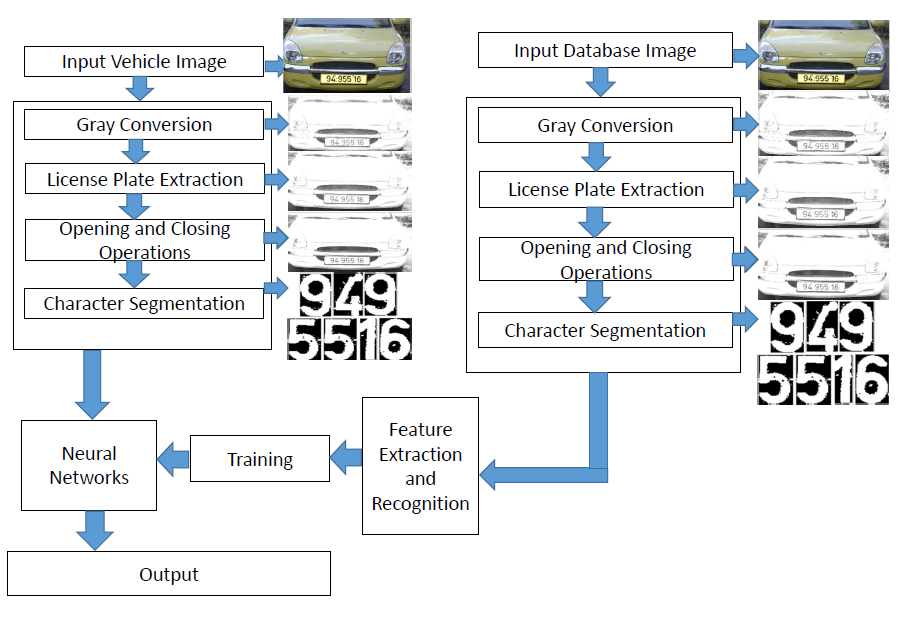


Fig 2.1 Block diagram

A camera is used to capture the number plate of a car as an input to the system. The input plate then undergoes series of extraction and segmentation operations. The output result of extraction and segmentation is compared with the already present data base. If the characters are already present in the database, then they are recognized and directly displayed on the display. But if the characters aren’t present, then the data base is first updated using neural networks and then the characters are displayed on the display.

* 1. Input Vehicle image

The input license plate is taken with the help of camera. The input image may vary with respect to the types of plate which are to be recognized and also in the environment in which they are taken.

They are summarized as follows:

1) Plate variations:

* Location: Plates exist in different locations of on a car.
* Size: Plates may have different sizes due to camera distance and zoom factor.
* Color: Plates may have various characters and background colors due to different plate types.
* Font: Plates have variety of numbers and alphabets.
* Occlusion: Plates may be obscured by dirt.
* Inclination: Plates may be tilted.
* Others: In addition to characters, a plate may contain frames and screws.

2) Environmental factors:

* Illumination: Input images may have different types of illumination, mainly due to environmental lighting and vehicle headlights.
* Background: The image background may contain patterns similar to plates, such as numbers stamped on a vehicle, bumper with vertical patterns and textured floors.
  1. Gray Conversion

Grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Gray scale is used for following reasons:

* Signal to noise ratio: Color information doesn't help us identify important features like edges, crests of the license plate.
* Complexity of the code: To want to find the edges based on luminance and chrominance is added to an additional work (debugging, additional pain in supporting the software, etc.). Using gray scale images helps to simplify these tasks.
* Speed: With modern computers, and with parallel programming, it's possible to perform simple pixel-by-pixel processing of a megapixel image in milliseconds. But recognition segmentation and other tasks can take much longer than that. Hence using gray scale will help to reduce the computational power and hence increase the processing speed.
  1. License Plate Extraction

Number plate extraction is the key step in License Plate extraction, which influences the accuracy of the system significantly. Extraction of number plate is a difficult task because of the inaccuracies involved during capturing of an image. The goal of this phase, given an input image, is to produce number of regions, with high probability of containing number plate and validate for true number plate.

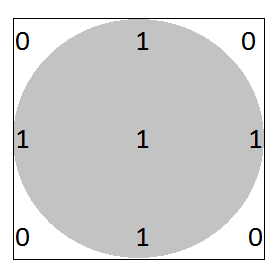
Opening and Closing Operation

It helps to further segment and detect the characters.

1. Structural Element

An essential part of the dilation and erosion operations is the structuring element used to probe the input image. Two-dimensional, or flat, structuring elements consist of a matrix of 0's and 1's, typically much smaller than the image being processed. The center pixel of the structuring element, called the origin, identifies the pixel of interest the pixel being processed. The pixels in the structuring element containing 1's define the neighborhood of the structuring element. These pixels are also considered in the dilation or erosion processing.

For example, the following illustrates a disk-shaped structuring element



Origin

Fig.2.2 Structural element

1. Erosion

The erosion of a binary image *f* by a structuring element *s* (denoted by *f https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/sign-erosion.gifs*) produces a new binary image *g* = *f https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/sign-erosion.gifs* with ones in all locations (*x,y*) of a structuring element's origin at which that structuring element *s* fits the input image *f*, i.e. *g(x,y)* = 1 is *s* fits *f* and 0 otherwise, repeating for all pixel coordinates (*x,y*).

Erosion is mainly used for eliminating the irrelevant details of an image. Figure 2.3(a) shows binary image to eliminate all squares except the largest once. Hence by performing erosion by using a structural element which is smaller than the objects required and figure 2.3(b) shows the eroded image.



Figure 2.3 (a) Binary Image (b) Eroded Image

1. Dilation

The dilation of an image *f* by a structuring element *s* (denoted *f https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/sign-dilation.gifs*) produces a new binary image *g* = *f https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/sign-dilation.gifs* with ones in all locations (*x,y*) of a structuring element's origin at which that structuring element *s* hits the input image *f*, i.e. *g(x,y)* = 1 if *s* hits *f* and 0 otherwise, repeating for all pixel coordinates (*x,y*). Dilation has the opposite effect to erosion it adds a layer of pixels to both the inner and outer boundaries of regions. Dilation is mainly used to enclose holes in a single region, reduce the gaps between different regions and to fill the intrusions into boundaries of a region.

Figure 2.4(a) shows image to be dilated and figure 2.4(b) shows the dilated image.



Figure2.4 (a) Original image (b) Dilated image

.

CHAPTER 3

Segmentation

* 1. Boundary Extraction

In this step, the eroded image is subtracted from the dilated image to obtain the boundaries of the characters.

Figure 3.1 (a) shows the original image. Figure 3.1 (b) shows the boundaries of the image obtained by subtracting the dilated and the eroded images.



Figure 3.1(a) Original image (b) Boundary

After boundary extraction, unwanted horizontal lines are again eroded using a structural line element for extracting the dimensions of the number plate.

* 1. Region Fill

Next step is to develop a simple algorithm for region filling based on set dilations, complementation, and intersections. Beginning with a point p inside the boundary, the objective is to fill the entire region with ‘black’. If we adopt the convention that all non-boundary (background) points are labeled ‘white’, then we assign a value of ‘black’ to p to begin.

The following procedure then fills the region with ‘black’:

Xk = ( Xk-1⊕B) ∩ AC k=1,2,3,

Where, X0=p, B is the symmetric structuring element

∩ - is the intersection operator (Fig. 3.2)

AC – is the complement of set A (Fig. 3.2)

The algorithm terminates at iteration step k if Xk = Xk-1. The set union of Xk and A contains the filled set and its boundary.



Figure 3.2 Region filling

* 1. Thinning Operation

Thinning is a morphological operation that is used to remove selected foreground pixels from [binary images](http://homepages.inf.ed.ac.uk/rbf/HIPR2/binimage.htm), somewhat like [erosion](http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm) or [opening](http://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm). It can be used for several applications. In this, it is used to tidy up the output of [edge detectors](http://homepages.inf.ed.ac.uk/rbf/HIPR2/edgdetct.htm) by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. The thinned useless lines are later removed using erosion.

* 1. Segmentation

Segmentation can be done using template matching

1. Template loading

This operation loads a template of character. Take 24 X 42 pixel A to Z alphabet and 0 to 9 number images. Read this all image and store in database. It becomes 36 character templates. This template made global.

1. Character Normalization

Segmented characters have very much variation in size. In this phase, all the characters are normalized to predefined height (Vertical Length) in pixel. As the characters always have variable width (Horizontal Length),each character image is normalized to a size of 24 X 42, by image mapping technique.

1. Template matching

Normalized character image compare with each template character image and find correlation between segmented character and template character. Selecting the most relevant image and write into text file.

* 1. Feature based character Recognition

Feature Extraction serves two purposes; one is to extract properties that can identify a character uniquely. Second is to extract properties that can differentiate between similar characters. A character can be written in a variety of ways, and yet can be easily recognized correctly by a Human. Thus, there exist a set of principles or logics that surpass all variation differences. Thus, the features used by the system work upon such properties which are close to the psychology of the characters.

CHAPTER 4

Result

4.1 Data Base:

Images with different clarity and different viewing angle has been considered as inputs for license plate extraction. Depending on the quality of the image, the efficiency of the code will vary.



Fig 4.1 Ideal Image



Fig 4.2 Blurred Image

  
Fig 4.3 Tilted Image

4.2 Outputs

Image taken as an input



Figure 4.4 Original License Plate Image

Median filter was used to reduce salt and pepper noise and the eroded image was subtracted from the dilated image to find the boundaries as shown in the figure 4.2



Figure 4.5 Boundary Extracted Image

F

The image was eroded using a line structural element to find out the horizontal lines in the image. These were eliminated by subtracting them fom the image. The detected horizontal lines are shown in the figure 4.3.



Figure 4.6 Extracting Dimensions

The horizontal lines obtained in figure 4.3 were subtracted from the image in 4.2 and the obtained image is shown in figure 4.4. The intensity of the image was adjusted to remove lower and higher intensity noise. The holes in the image were filled.



Figure 4.7 Region Filling

Erosion was performed using a structuring line element of size (3,90) to retain only those objects similar to the size of necessary characters. Opening operation was performed to remove the objects with fewer pixels than (a/15\*b/15) where (a, b) is the size of the image. Higher intensity regions were enclosed in a bounding box. The final step implemented till now is shown in the figure 4.5



Figure 4.8 Character segmentation

A GUI of the code implemented till now was prepared as shown in figure 4.6.

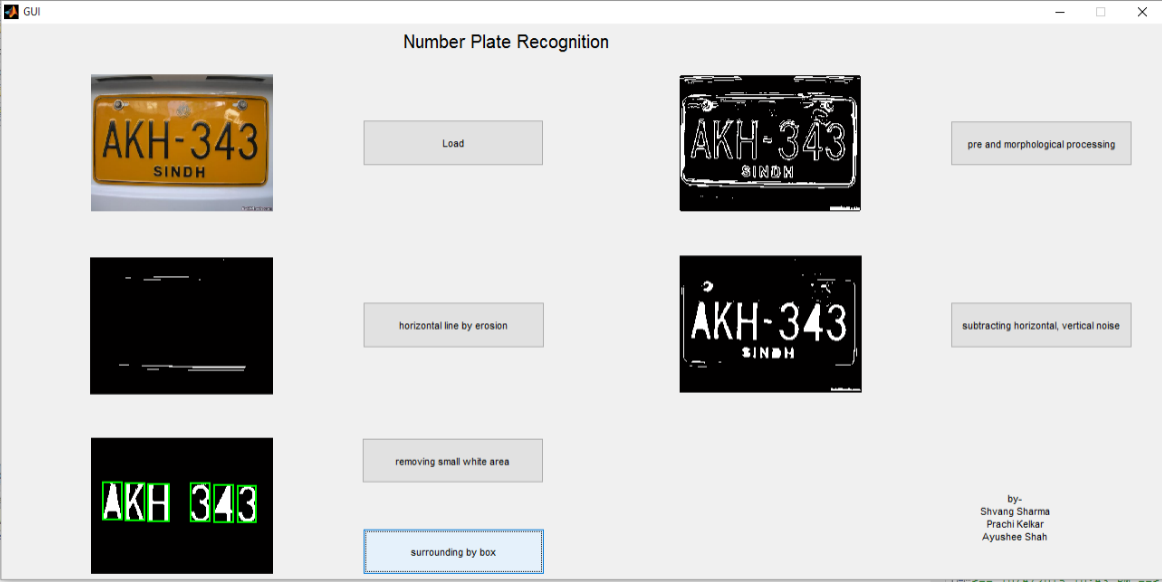


Figure 4.9 GUI in Matlab

CHAPTER 5

Conclusion

The noisy input image has been treated with various filters and image enhancement operations which leads to better efficiency of further steps. A median filter has been used to remove salt and pepper noise. For extracting the boundaries, the eroded and dilated image was subtracted from each other. Intensity adjustment has been performed on the image to remove low and high intensity noises. Horizontal lines have been eliminated using a line as a structural element. Holes have been filled and the image has been thinned.

Opening operation has been performed where the elements in the image smaller than a set value has been eliminated. Segmentation process has been started and the high intensity regions of the image were enclosed using bounding blocks. The further, segmentation will include separation of the blocks and selecting its validity.

Important applications include ticketless parking management, access control which permits access to a facility to only the non-restricted vehicles, road tolling, law enforcement and traffic analysis.

The Matlab code produces expected output for ideal images for number plate in focus. Hence, the reliability of the code decreases as the quality of input image deteriorated.

CHAPTER 6

Future Plan

The entire code works well only if the license plate is focus and is of good resolution. A proper function for license plate extraction is to be added. So that even if there is background, the image can still be processed.

Functions for proper segmentation is to be added, the efficiency of segmentation will be improved by using neural networks. The precision of data base or function for character recognition will be improved after every iteration of code using neural networks.

If time permits real time processing with multiple number plate extractions will also be implemented to improve the usability and widen the scope of the project.

References

1. Rafael C. Gonzlez and Richard E. Woods, “Digital Image Processing,” *Pearson Education International*, Third Edition, pp. 652-660.
2. Shan Du*,* Mahmoud Ibrahim, Mohamed Shehata*,* and Wael Badawy, “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review,” in IEEE transactions on circuits and systems for video technology, vol. 23, no. 2, February 2013.
3. Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas, “License Plate Recognition From Still Images and Video Sequences: A Survey” in IEEE transactions on intelligent transportation systems, vol. 9, no. 3, September 2008.
4. Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen, “Automatic License Plate Recognition” in IEEE transactions on intelligent transportation systems, vol. 5, no. 1, March 2004.
5. Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos,

Vassili Loumos, and Eleftherios Kayafas, “A License Plate-Recognition Algorithm for Intelligent Transportation System Applications” in IEEE transactions on intelligent transportation systems, vol. 7, no. 3, September 2006.

1. Michael A.Nielsen, “Neural Networks and Deep Learning”: http://neuralnetworksanddeeplearning.com, Determination Press, 2015.