# License plate localization and recognition in camera pictures

**A Project Report**

*Submitted in partial fulfillment for the award of the course*

**SWE1010-Digital Image Processing**


**Submitted By:**

MANO BALA.R(16MIS0433)

NITHEESHKUMAR.A(16MIS0018)

MOHAN.V(16MIS0522)

*Under the guidance of*

**Dr. Agilandeeswari L**

Associate Professor




**School of Information Technology & Engineering**


OCT 2017

## DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled "**License plate localization and recognition in camera pictures**" submitted by me to SITE, VIT University, Vellore in partial fulfillment of the requirement for the award of the DIGITAL IMAGE PROCESSING is a record of bonafide project work carried out by me under the guidance of **Agilandeeswari L.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course in this institute or university.

Place : Vellore Signature of the Candidates

Date :

**ABSTRACT**

In last couple of decades, the number of vehicles has increased drastically. With this increase, it is becoming difficult to keep track of each vehicle for purpose of law enforcement and traffic management. License Plate Recognition is used increasingly nowadays for automatic toll collection, maintaining traffic activities and law enforcement. Many techniques have been proposed for plate detection, each having its own advantages and disadvantages. The basic step in License Plate Detection is localization of number plate. The approach mentioned in this project is a histogram based approach. This approach has an advantage of being simple and thus faster. Initially, license plate localization is implemented using MATLAB and verified for its functionality.

In this approach, a digital camera is used to capture video and feed it to the kit. The kit processes each frame individually and provides the co-ordinates of location with maximum probability of having a number plate. Later, this information is used for recognizing actual number of the license plate.

**Keywords**: license plate recognition, object character recognition, neural nets, image processing

## INTRODUCTION

With increasing number of vehicles on roads, it is getting difficult to manually enforce laws and traffic rules for smooth traffic flow. Toll-booths are constructed on freeways and parking structures, where the car has to stop to pay the toll or parking fees. Also, Traffic Management systems are installed on freeways to check for vehicles moving at speeds not permitted by law. All these processes have a scope of improvement. In the center of all these systems lies a vehicle. In order to automate these processes and make them more effective, a system is required to easily identify a vehicle. The important question here is how to identify a particular vehicle? The obvious answer to this question is by using the vehicle's number plate.

Vehicles in each country have a unique license number, which is written on its license plate. This number distinguishes one vehicle from the other, which is useful especially when both are of same make and model. An automated system can be implemented to identify the license plate of a vehicle and extract the characters from the region containing a license plate. The license plate number can be used to retrieve more information about the vehicle and its owner, which can be used for further processing. Such an automated system should be small in size, portable and be able to process data at sufficient rate.

Various license plate detection algorithms have been developed in past few years. Each of these algorithms has their own advantages and disadvantages. As multiple detections are available for single license plate, post –processing methods are applied to merge all detected regions. In addition, trackers are used to limit the search region to certain areas in an image. In this approach, initial image processing and binarization of an image is carried out based on the contrast between characters and background in license plate. After binarizing the image, it is divided into different black and white regions. These regions are passed through elimination stage to get the final region having most probability of containing a number plate.

## Literature study

The first report we are discussing is: License Platelocation in Camera Images (Paper 1) This is from Halina Halina Kwaśnicka and Bartosz Wawrzyniak (Wrocław University of Technology)

First it deals with the conditions of the license plate: weather, light, place of number plate, movement, damage and other forms of letters and numbers in the image. Most of these points could be solved by improving light invasion, better cameras and high-quality image processing techniques.

Using algorithms with shape and color would be inefficient because there are many different number plates. The correct way to do Number Recognition is to use the difference between background color and letter color, which is very clear for each number plate.

The second paper is from Naikur Bharat kumar Gohil called Car License PlateDetection. Here the full implementation is discussed in Matlab code.

## Purpose of this Project

The main purpose of this project is to detect and extract license plate number from an image provided by a camera/image. An efficient algorithm is developed to detect a license plate in various luminance conditions. This algorithm extracts the license plate data from an image and provides it as an input to the stage of Car License Plate Recognition. The image of a vehicle is given as an input from the camera. Extracted image of the number plate can be seen on television for verification purpose.

The scope of this project is to detect the license plate from the given image and observe the output. This project can work as a base for future improvements in the field of image processing, especially in license plate extraction and plate number recognition.

## Significance of this Project

Through this project, an algorithm for license plate number extraction from an image is implemented in MATLAB.

## FUNDAMENTALS OF IMAGE PROCESSING

An image is used to convey useful information in a visible format. An image is nothing but an arrangement of tiny elements in a two-dimensional plane. These tiny elements are called Pixels. A large number of pixels combine together to form an image, whether small or large.

Each pixel represents certain information about the image, like color, light intensity and luminance. A large number of such pixels combine together to form an image. Pixel is the basic element used to describe an image. Mostly, each pixel in an image is represented in either RGB (Red Green Blue) format or YCbCr format. In case of an RGB image, all the three components, namely R, G and B combine together to convey information about the color and brightness of a single pixel. Each component consumes certain memory space during image processing.

In case of a YCbCr image, each pixel in an image is represented as a combination of Y and Cb/Cr values. Here, Y stands for luminance, which describes light intensity, and Cb/Cr stands for chroma component, which describes color information for an image. Over the time, it has been found that YCbCr components of an image convey sufficient amount of information compared to its counter parts RGB, with less amount of memory space. This is a major advantage nowadays, as most of the applications require sufficient information at very high speed and less storage.

### RGB Format

In case of an RGB image, each pixel is represented by three different components R, G and B. Each of these components requires at least 8 bits for their storage. In general, a single pixel may require upto 8 * 3 bits for its storage. An example of a representation of single pixel in RGB format is shown below.


Fig. 1.0 Representation of pixels in RGB format.

The value of R, G and B each ranges from 0-255. A value of (0, 0, 0) represents a black pixel, (255, 0, 0) represents a red pixel and (0, 255, 0) represents a green pixel. So, 8 bits are required to store value for a single component.

## YCbCr Format

In contrast to RGB format, the YCbCr format is available with various kind of interleaving. For example, a 4:2:2 YCbCr format suggests that a single pixel is represented by two components, Y and C. Cb and Cr components are interleaved among the pixels. So if one pixel is represented by a combination of Y and Cb, the adjacent pixel will be represented by a combination of Y and Cr. Even if the Cb and Cr components are interleaved, its effect is not visible to human eye.

| Y | Cb | Y | Cr | Y | Cb |
|---|---|---|---|---|---|

Fig. 1.1 Representation of pixels in YCbCr format.

Values for Y, Cb and Cr vary from 0-255. Thus, to store a single pixel, the amount of storage required is 8 * 2 bits, which is less compared to that required by RGB format. For this project, Texas Instrument's EVM320DM6437 kit is to be used for license plate detection. The kit contains internal buffers to store the incoming frames of video. The format for the type of storage is shown below.

| Cb | Y | Cr | Y |
|---|---|---|---|
| Cb | Y | Cr | Y |
| Cb | Y | Cr | Y |

Fig. 1.2 A part of frame buffer storage for input video frames.

From the above image, it is seen that the storage of frame starts with a C component and then a Y component. Therefore, at the $0^{th}$ location, one can find the C component while at the $1^{st}$ and alternate locations of Frame Buffer, one can find the Y component.

## NTSC and PAL Standards

NTSC and PAL are the two most commonly used standards used for broadcasting. NTSC stands for National Television System Committee. This standard is being used in most parts of Northern America and countries like South Korea, Japan, and etcetera. Videos broadcasted using NTSC standard contains a sequence of images with resolution of 720 * 480 pixels. The video is displayed at the frame rate of 30 frames per second.

PAL stands for Phase Alternate Line. PAL Standard is used mainly in countries like India, China, United Kingdom, and etcetera. This standard supports the video resolution of 720 * 576 pixels at the frame rate of 25 frames per second.

## MATLAB IMPLEMENTATION

This part describes the implementation of License Plate Detection algorithm using MATLAB. MATLAB is a very powerful software tool used to implement the tasks that require extensive computation. It provides easy and quicker implementation of algorithms compared to C and C++. The key feature in MATLAB is that it contains a rich library functions for image processing and data analysis. This makes MATLAB an ideal tool for faster implementation and verification of any algorithm before actually implementing it on a real hardware. Sometimes, debugging of errors on actual hardware turns out to be a very painful task. MATLAB provides an easy approach for debugging and correction of errors in any algorithm. Other than this, MATLAB contains many features including workspace, plot, imread, imhist, imshow, etc. for data analysis and image processing, which makes it a better choice over other software languages like C and C++.

Considering the above advantages, the writer of this project initially implemented an algorithm for License Plate Detection using MATLAB. The algorithm initially used various inbuilt functions and implemented few user defined routines related to image processing. Once the algorithm was developed, it was verified with multiple input images containing car number plates. The input images contained number plates that were aligned horizontally as well as at some angle from horizontal axis. Once the algorithm

was completely verified, the in-built functions of MATLAB were replaced by user- defined functions. A flow-chart showing the basic implementation of algorithm is shown below:

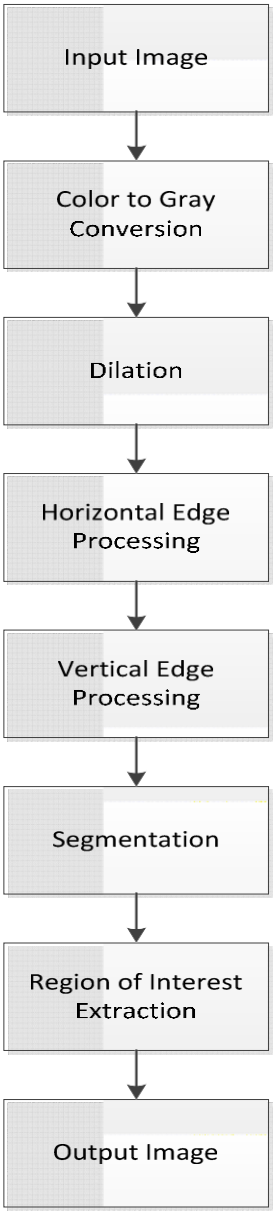



Fig. 1.3 Flowchart showing license plate detection algorithm in MATLAB.

The steps of implementing License Plate Detection algorithm in MATLAB are described below.

## Convert a Colored Image into Gray Image

The algorithm described here is independent of the type of colors in image and relies mainly on the gray level of an image for processing and extracting the required information. Color components like Red, Green and Blue value are not used throughout this algorithm. So, if the input image is a colored image represented by 3-dimensional array in MATLAB, it is converted to a 2-dimensional gray image before further processing. The sample of original input image and a gray image is shown below:


Fig 1.4 Original color image.


Fig. 1.5 Gray image.

## Dilate an Image

Dilation is a process of improvising given image by filling holes in an image, sharpen the edges of objects in an image, and join the broken lines and increase the brightness of an image. Using dilation, the noise with-in an image can also be removed. By making the edges sharper, the difference of gray value between neighboring pixels at the edge of an object can be increased. This enhances the edge detection.

In Number Plate Detection, the image of a car plate may not always contain the same brightness and shades. Therefore, the given image has to be converted from RGB to gray form. However, during this conversion, certain important parameters like difference in


Fig. 1.6 Dilated image.

color, lighter edges of object, etc. may get lost. The process of dilation will help to nullify such losses.

## Horizontal and Vertical Edge Processing of an Image

Histogram is a graph representing the values of a variable quantity over a given range. In this Number Plate Detection algorithm, the writer has used horizontal and vertical histogram, which represents the column-wise and row-wise histogram respectively. These histograms represent the sum of differences of gray values between neighboring pixels of an image, column-wise and row-wise.

In the above step, first the horizontal histogram is calculated. To find a horizontal histogram, the algorithm traverses through each column of an image. In each column, the algorithm starts with the second pixel from the top. The difference between second and

first pixel is calculated. If the difference exceeds certain threshold, it is added to total sum of differences. Then, algorithm will move downwards to calculate the difference between the third and second pixels. So on, it moves until the end of a column and calculate the total sum of differences between neighboring pixels. At the end, an array containing the column-wise sum is created. The same process is carried out to find the vertical histogram. In this case, rows are processed instead of columns.

## Passing Histograms through a Low Pass Digital Filter

Referring to the figures shown below**,** one can see that the histogram values changes drastically between consecutive columns and rows. Therefore, to prevent loss of important information in upcoming steps, it is advisable to smooth out such drastic changes in values of histogram. For the same, the histogram is passed through a low-pass digital filter. While performing this step, each histogram value is averaged out considering the values on it right-hand side and left-hand side. This step is performed on both the horizontal histogram as well as the vertical histogram. Below are the figures showing the histogram before passing through a low-pass digital filter and after passing through a low-pass digital filter.

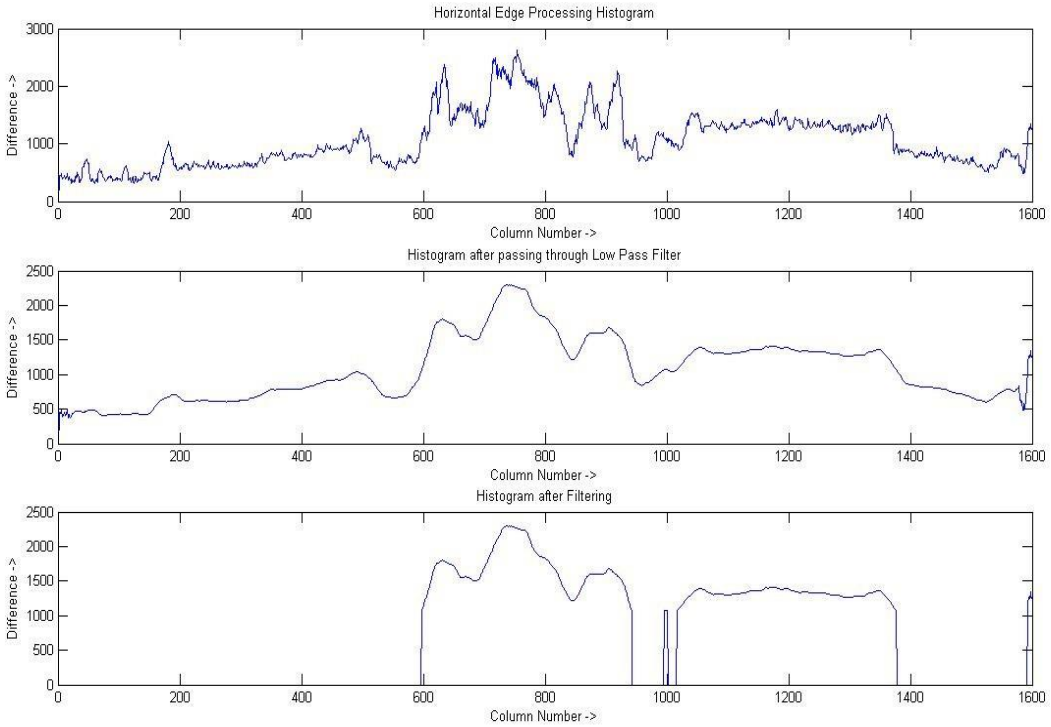



Fig. 1.7 Horizontal edge processing histogram

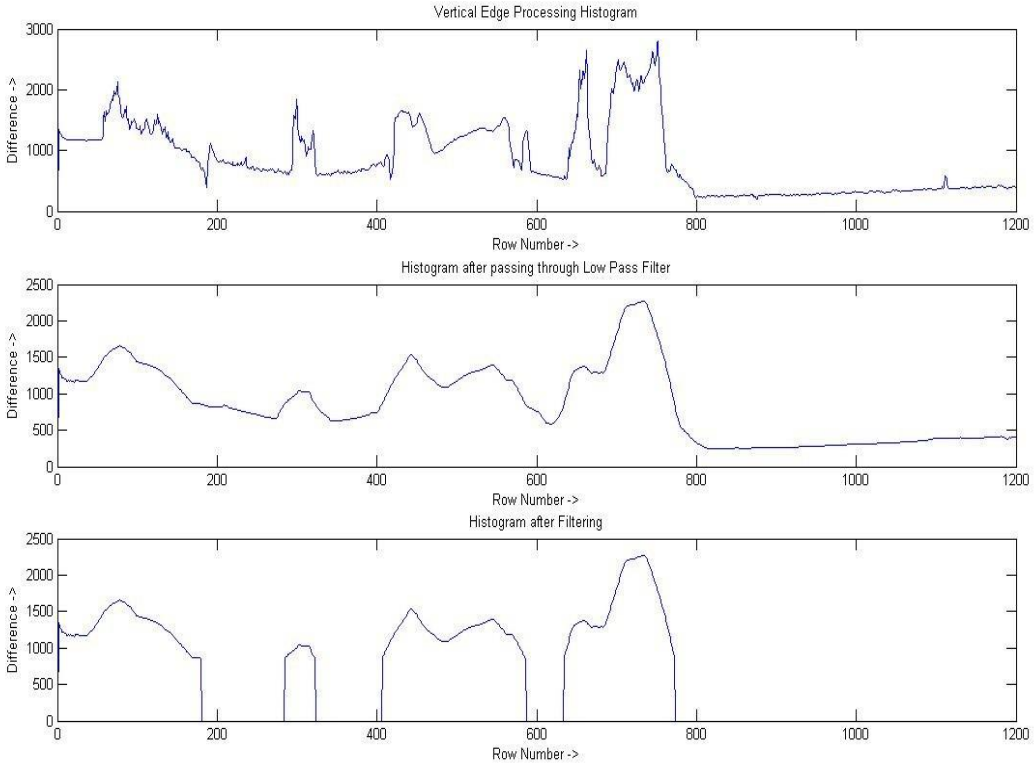



Fig. 1.8 Vertical edge processing histogram.

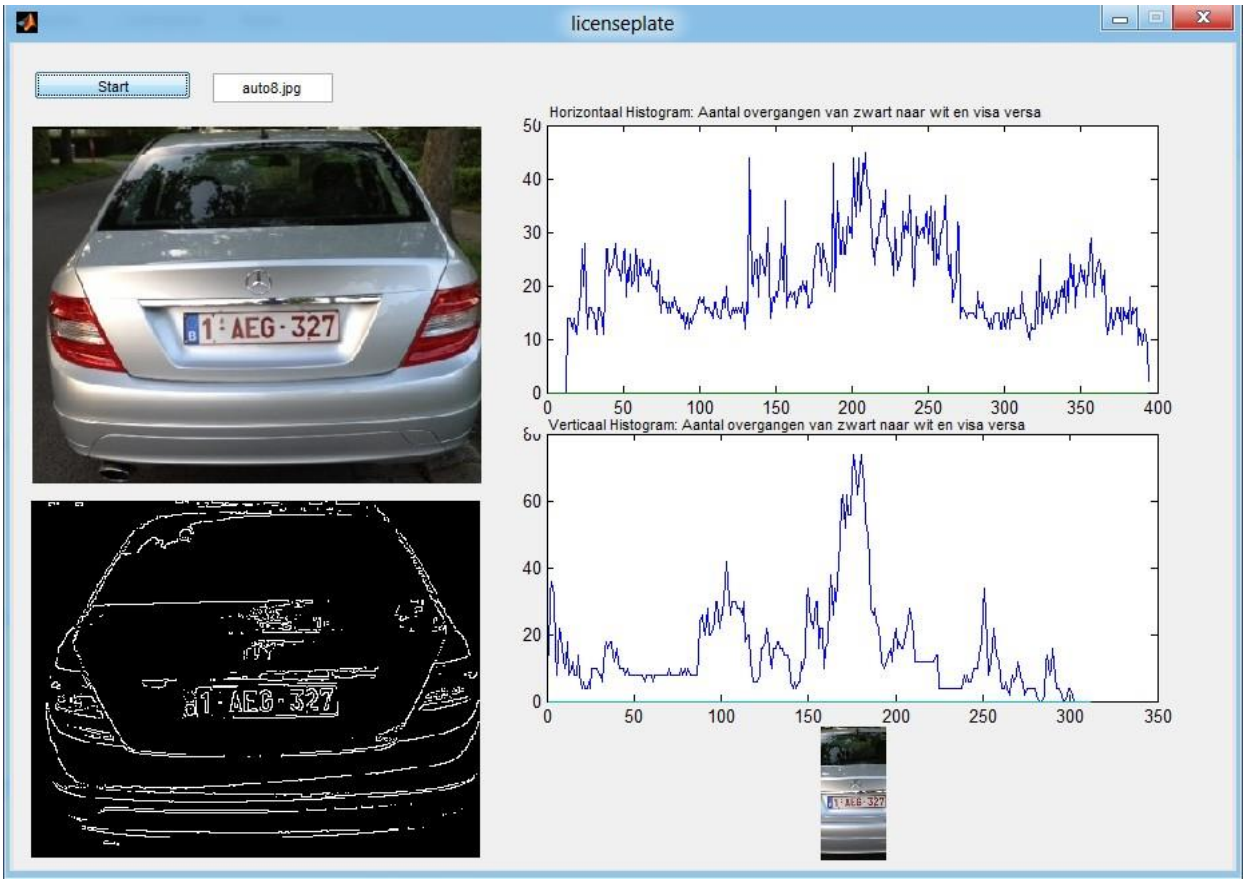



Fig. 1.9 Image Processing

## Optical Character Recognition template matching

Although not specified in our specification, we went one step further by implementing an optical character recognition algorithm. The implication of the ocr algorithm used by us is explained below in steps:
1.We get an already segmented license plate inside the ocr function (Figure 2.1).
2.This image is converted to grayscale for calculating a treshold (Figure 2.2).
3. For the treshold image is converted to a binary image (Figure 2.3).
4.All linked regions are segmented.
5.The average number of black pixels per region is calculated as treshold.
6.If the number of black pixels per region is smaller than the treshold (average), it will be deleted.
7.All regions where the height is smaller than the width (x10%) are removed. The final result is shown in Figure 2.4
8. Finally, we compare each remaining region with each letter from the database (Figure 2.5),
This comparison occurs by a logical AND, the region with the most corresponding black
Pixels are considered to be that letter.
9. Any letter found will be attached to a 'total string'.

Note that the implemented OCR works only for German license plates because we have only found a German number plate letter database.

Figure 2.1

Figure 2.2

Figure 2.3

Figure 2.4

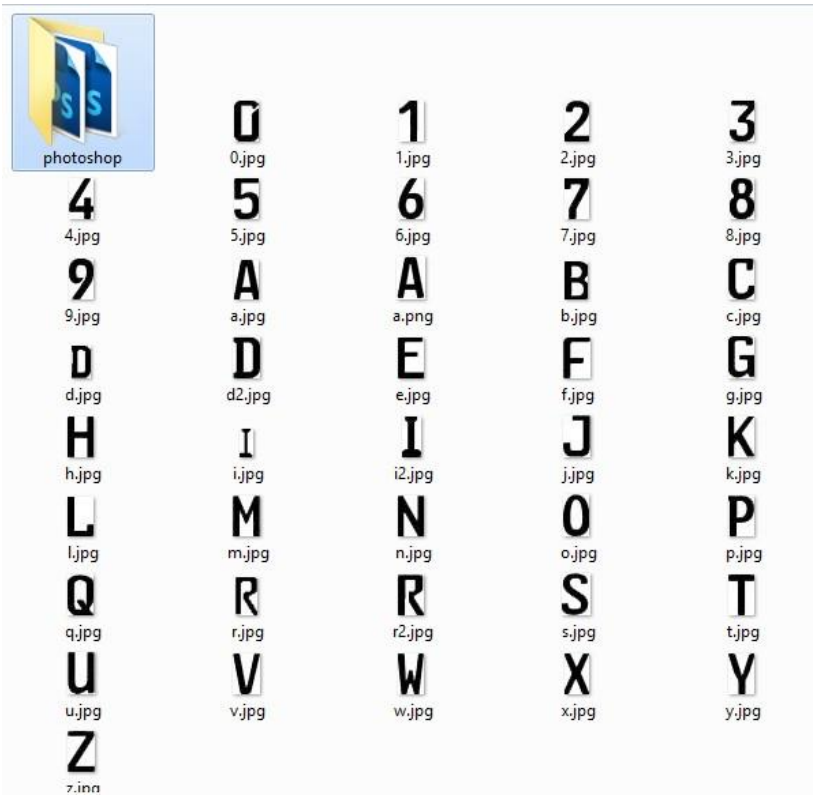



Figure 2.5

## Filtering out Unwanted Regions in an Image

Once the histograms are passed through a low-pass digital filter, a filter is applied to remove unwanted areas from an image. In this case, the unwanted areas are the rows and columns with low histogram values. A low histogram value indicates that the part of image contains very little variations among neighboring pixels. Since a region with a license plate contains a plain background with alphanumeric characters in it, the difference in the neighboring pixels, especially at the edges of characters and number plate, will be very high. This results in a high histogram value for such part of an image.

Therefore, a region with probable license plate has a high horizontal and vertical histogram values. Areas with less value are thus not required anymore. Such areas are removed from an image by applying a dynamic threshold.

In this algorithm, the dynamic threshold is equal to the average value of a histogram. Both horizontal and vertical histograms are passed through a filter with this dynamic threshold. The output of this process is histogram showing regions having high probability of containing a number plate.

## Segmentation

The next step is to find all the regions in an image that has high probability of containing a license plate. Co-ordinates of all such probable regions are stored in an array. The output image displaying the probable license plate regions is shown below.


Fig. 2.6 Output of segmentation.

## Region of Interest Extraction

The output of segmentation process is all the regions that have maximum probability of containing a license plate. Out of these regions, the one with the maximum histogram value is considered as the most probable candidate for number plate. All the regions are processed row-wise and column-wise to find a common region having maximum horizontal and vertical histogram value. This is the region having highest probability of containing a license plate. The image detected license plate is shown below:

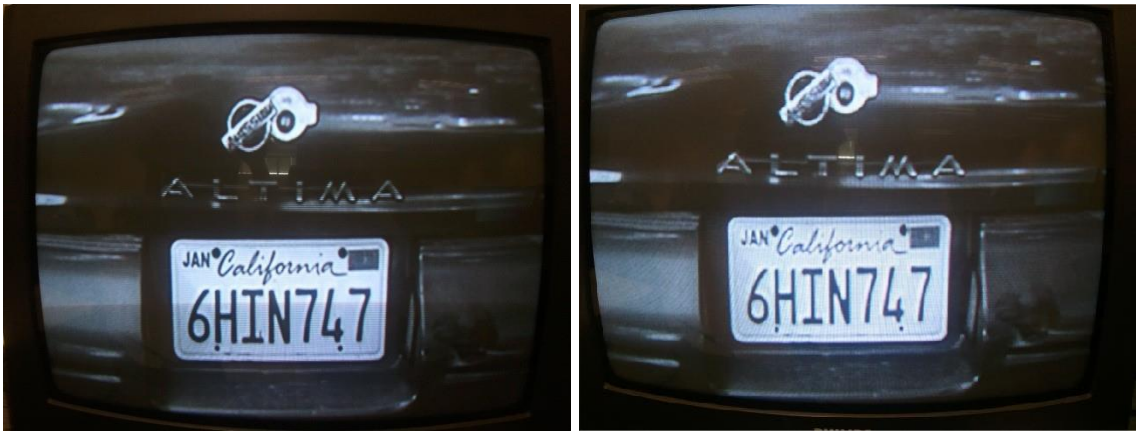

Fig. 2.7 Detected license plate.

This algorithm was verified using several input images having resolution varying from 680 * 480 to 1600 * 1200. The images contained vehicles of different colors and varying intensity of light. With all such images, the algorithm correctly recognized the

number plate. This algorithm was also tried on images having number plate aligned at certain angle (approximately 8-10 degree) to horizontal axis. Even with such images, the number plates were detected successfully.

## Noise Removal

In this step, the input image is passed through a linear filter for noise removal. An image is passed through a 9 * 9 mask. Therefore, value of each pixel is set equal to the average value of its 8 neighboring pixels. This type of filter for noise removal is called linear smoothening filter. Such a filter is very effective in terms of performance and speed. Thus, this filter was selected for implementation on actual hardware.




(a) (b)

Fig. 2.8 (a) Image after noise removal (b) Image after dilation.

## Dilation

In order to join broken edges and remove noisy pixels, dilation is implemented on the image after noise removal. In this step, each pixel is compared with its neighboring pixels and its value is set equal to the maximum value out of both the neighboring pixels. This process makes edges of an image sharper. In turn, it helps in better detection of an image. If an input image is blurred, this step will help to improve such blurred image and make it easy for detection.

## Vertical Edge Processing

After removing noise and passing an image through dilation, the next step is to process an image for license plate detection. This is done in two steps, processing image column-wise and row-wise. In case of vertical edge processing, the image is processed column-wise. Each of the 720 columns is traversed one after another. A histogram is

prepared based on this processing. This histogram is stored in an array of 720 elements. The histogram generated, will have some very drastic changes due to presence of noise and disturbance in an image. To remove such unnecessary disturbance, the histogram is passed through a low-pass filter to smoothen out the changes in histogram values. To implement this step, each histogram value is set to a value equal to the average of previous fifteen histogram values and next fifteen histogram values. After performing this step, the histogram is passed through a band-pass filter. In this case, a dynamic threshold is applied and all the values less than this threshold are set to 0. This will remove the unnecessary columns from an image.

## Horizontal Edge Processing

This step is similar to the previous step except some of the changes. This step is performed on an image one row after another. A total of 480 histogram values will be obtained and stored in an array. The array will be passed through a low-pass smoothening filter and then through the band-pass filter. The result will be an array of histogram with necessary rows only.

## Segmentation

Once the processing of an image is completed, the next step is segmentation. On this hardware, the segmentation is performed by observing the values of filtered histograms. A set of row and column numbers having highest probability of containing a

number plate are prepared. These set of values are passed to the next step known as region of interest extraction.

## Edge detection – Roberts Technique

To delete unnecessary information from The image, we only need to work with the edges of the image because of this, we follow the procedure described in paper 3. To detect the edges we use a built-in matlab function. But first of all, we convert the original image to a grayscale image, which allows us to calculate matlab a treshold for the different intensities in the grayscale image. Everything above that particular treshold will be white and all under the threshold black. So, we have a binary image, and finally the edgedetection algorithm is applied.


Fig. 2.9 Edge Detection

## Region of Interest Extraction

For this project, the region of interest in any given image is the license plate. This region is found by applying a concept that for a given image, the region containing a number plate will have maximum number of edges compared to any other part in an image. Applying this concept to all the extracted segments, the co-ordinates of the required region are extracted.

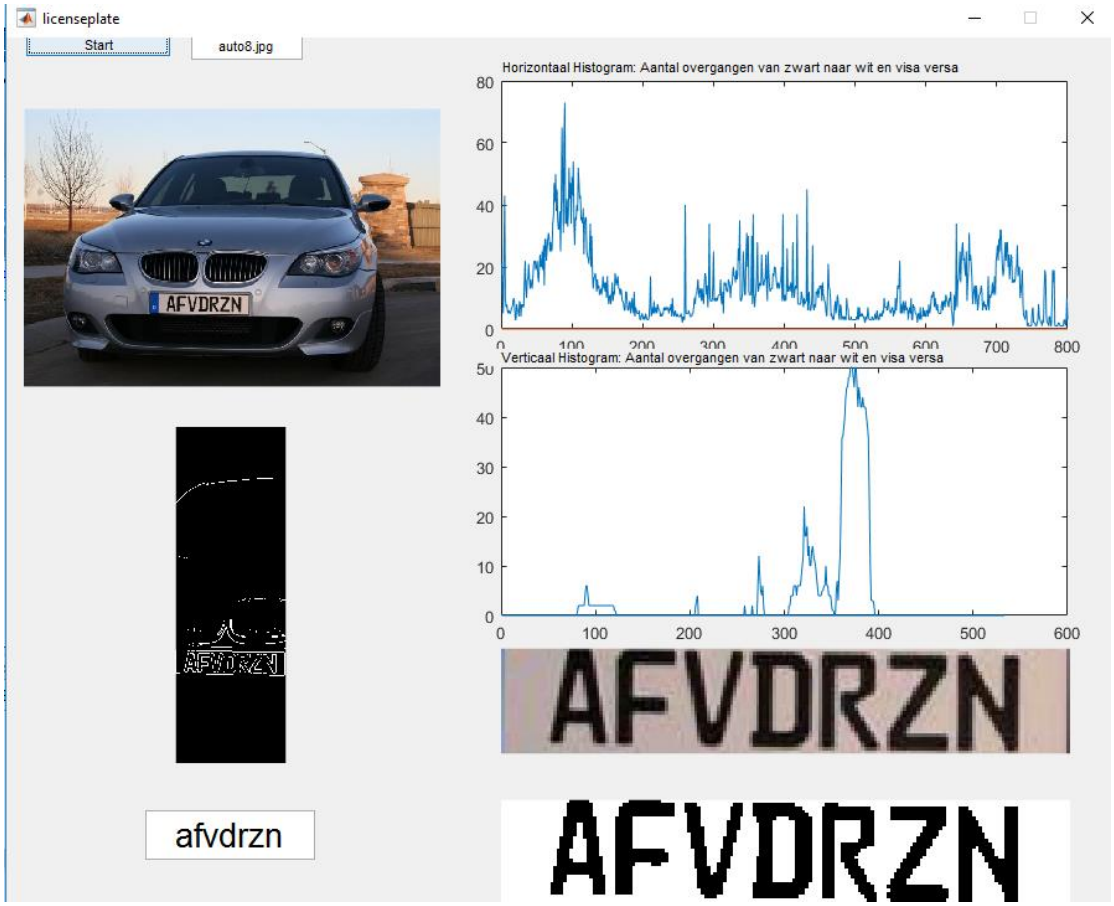
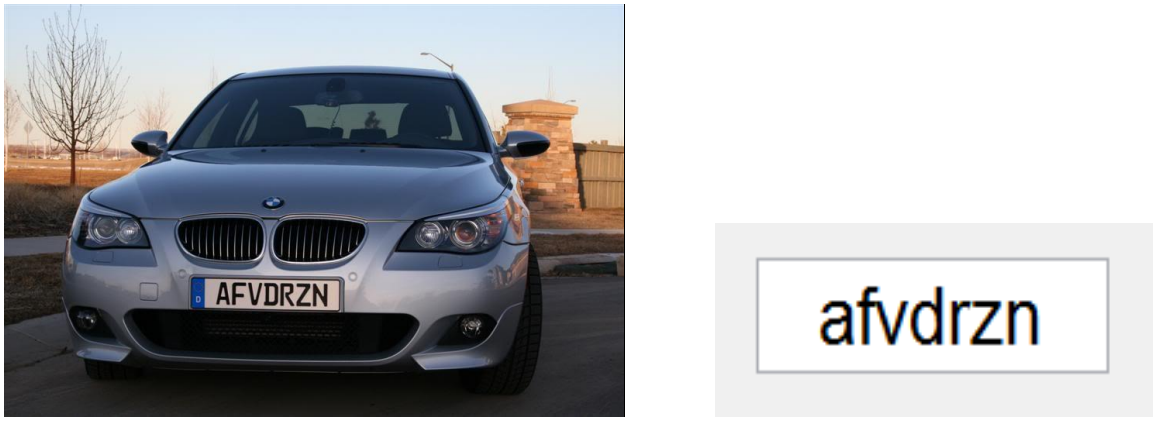



Fig. 3.1 License plate extracted from original image.

**RESULTS**

The algorithm was tested using different license plates having various background conditions, light condition and image quality. Some of the output results are shown below:




(a) (b)

Fig. 3.2 (a) Image of actual car (b) Image of Extracted Numbers of license plate.

The table below shows the type and number of plates and success ratio for detection of plates.

| License Plate Conditions | Success using MATLAB Implementation (%) | Success using Hardware Implementation (%) |
|---|---|---|
| Sunlight | 100 | 94 |
| Cloudy weather | 100 | 92 |
| Shade | 100 | 94 |
| Different Backgrounds | 100 | 95 |

Table 1.0 Performance of License Plate Detection under Various Conditions

**BIBLIOGRAPHY**

[1] Clemens Arth, Florian Limberger and Horst Bischof, "Real-Time License Plate Recognition on an Embedded DSP-Platform", Proceedings of IEEE conference on Computer Vision and Pattern Recognition, pp 1-8, June 2007.

[2] Halina Kwasnicka and Bartosz Wawrzyniak, "License plate localization and recognition in camera pictures", AI-METH 2002, November 13-15, 2002.

[3] Pramod Kapadia, "Car License Plate Recognition Using Template Matching Algorithm", Master Project Report, California State University, Sacramento, Fall 2010.