Addis Ababa University
School of Graduate Studies
Department of Computer Science

Recognition of Ethiopian Car Plate

By: Seble Nigussie Ayalew

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

April, 2013

Addis Ababa University
School of Graduate Studies
Department of Computer Science

Recognition of Ethiopian Car Plate

By: Seble Nigussie

Approval by Board of Examiners: Signature

Dr. Yaregal Assabie ________________________

Advisor

______________________________ ________________________

Examiner

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my advisor Dr. Yaregal Assabie for his valuable advice and support. I am also grateful to all my families, colleagues and friends, especially to Yoseph Birhanu for helping me in the data collection process.

My special thanks go to my love Tamirat Feyissa, who is been beside me at all the worst and good times. Thank you very very very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ITS	–	Intelligent Transportation Systems
CPR	–	Car Plate Recognition
LPR	–	License Plate Recognition
PD	–	Plate Detection
CC	–	Connected Component
CS	–	Character Segmentation
CR	–	Character Recognition
CCL	–	Connected Component Labeling
CCA	–	Connected Component Analysis
OCR	–	Optical Character Recognition
ANN	–	Artificial Neural Network
SAD	–	Sum of Absolute Differences
SSD	–	Sum of Squared Differences
SE	–	Structuring Element
GUI	–	Graphical User Interface

ABSTRACT

As time goes, application areas of information and communication technologies are growing dramatically as well. In these days, every potential problem is automated or it is being automated in order to solve or make things better. In the area of transportation systems, a lot of applications have been developed with development of communication and information processing technologies, officially called Intelligent Transportation Systems (ITS).

In this work, one of the fundamental elements of ITS called Car Plate Recognition (CPR) is developed for Ethiopian car plates. The proposed system has three major modules: Plate Detection, Character Segmentation and Character Recognition. For plate detection, a gabor filter based method is proposed. In this module, even though, the gabor filter is the core unit that roughly locates possible plate regions, the module also applies a series of other techniques, namely, binarization, morphological closing operation and connected component analysis consecutively on the filter response, to detect the legitimate plate region. For character segmentation process a connected component analysis method is used. But before, the actual segmentation process, the plate image passes through a number of preprocessing tasks that dramatically increase the accuracy of the segmentation outcome. Of these preprocessing tasks plate's orientation correction and plate's frame removal are the major ones. For plate's orientation correction we used a combination of hough transform and shear transform. For plate's frame elimination we used a series of binary operations. Besides these preprocessing tasks, the module performs further post segmentation operations that are done in the segmentation outcome of the CCA. The main objective of the post segmentation stage is to separate connected character objects (if any exists) with the help of plate structure information. Finally, a correlation based template matching method is used for character recognition. In addition to the correlation value, the recognition process is supported by color analysis techniques and location information of characters. The prototype of the proposed system is developed using MATLAB™ and its performance is tested on 350 RGB car images that are taken under different angle, distance, motion and illumination conditions. The developed system results in an accuracy of 63.14% and also it is able to recognize a plate between 2-5 seconds depending on whether post processing operations are needed or not.

Keywords: Ethiopian Car Plates, Ethiopian Car Plate Recognition, License Plate Recognition, Plate Detection, Plate Characters Segmentation, Character Recognition, Plate's Frame Elimination, Ethiopian Car Plate Type Identification

CHAPTER ONE

INTRODUCTION

1.1 Background

Lately, with the development of economy, auto population has been increased **[**26**].** As a result, the transportation system has been struggling to serve users efficiently. In order to solve this problem, many countries in the world still depend on building more and more road infrastructures to reduce traffic stress. But, for different reasons this solution is still not capable enough to solve the existing problem. So, this is where the concept of using Intelligent Transportation Systems (ITS) emerges.

Intelligent transportation systems are systems that apply technologies of information processing and communication, on transport infrastructures to improve the transportation outcome. In these days, numerous intelligent transportation systems are being used by users in various applications. One of these systems is Car Plate Recognition (CPR).

CPR, well known as License Plate Recognition (LPR), is one of the most important and popular elements of ITS. CPR is a mass surveillance system that captures the image of vehicles and recognizes their license number [37]. The main objective of CPR systems is to detect and recognize license plate of vehicles from their image, and present the identified information about the plate to machines, in a way that further applications can understand it. Some of these applications, where CPR systems are gaining popularity include: access control in restricted areas, law enforcement, stolen car detection, automatic toll collection, car parking automation and border crossing control.

A typical CPR system constitutes four major phases: Image Acquisition, Plate Detection, Character Segmentation and Character Recognition.

Process of CPR begins with image acquisition. Image acquisition is the process of acquiring vehicle images that are going to be fed to the CPR system. The input images could be static pictures of cars or they could be extracted from video sources. Both of the techniques have their own pros and cons on the whole recognition process. Especially, quality of images from video sequences is usually not clear as static images [3]. Accordingly, CPR systems that detect plate regions from a video source [1, 2] passes through additional preprocessing tasks than other CPR systems that work on still images.

The second phase of CPR is Plate Detection (PD). This phase is a key step of CPR systems, due to the fact that the execution of the upcoming modules depends on the successful completion of this module. The main goal of the PD module is to detect and extract out license plate regions from the acquired input image. The detection process is usually very sensitive to quality of input images, such as, the images noise level and illumination condition. Thus, normally the input image passes through a number of preprocessing tasks that increase the quality of the image as well as the performance of the plate detection module. Some of the common preprocessing tasks include: noise filtering, contrast enhancement and edge enhancement. Finally, once these preprocessing tasks are done, a series of image processing techniques and algorithms will be applied on the enhanced input image to detect plate regions.

The next phase of CPR is Character Segmentation (CS). At this stage of CPR, various plate characters segmentation techniques are used to identify and segment out each alpha-numeric characters found on the segmented plate region image obtained from the PD module. In this phase, diverse aspects make the character segmentation task complicated; including, image noise, plate‟s frame, space mark, plate‟s rotation, and light variance. This phase is usually a prerequisite for the next module (i.e., character recognition). Hence, a CPR developer has to pay more attention on selecting the right character segmentation algorithm for the successful completion of the whole recognition process.

The final and most important phase of CPR is Character Recognition (CR). In this stage, the alpha-numeric character images that are extracted from the plate by the CS module will be recognized. Unlike applications like document character recognition, character recognition in CPR systems face a lot of challenges, including noisy and low quality images, various illumination conditions and a high degree of plate‟s orientation.

The first automatic CPR system is introduced in the late 1970‟s, although; it gets more attention since 1990‟s with the development of high quality digital cameras and computation devices [6]. Since then, a large number of CPR systems that utilize a variety of image processing algorithms have been developed. The reason that CPR systems kept developing is, due to the fact that CPR systems are dependent on country specific car plate features.

1.2 Statement of the problem

In these days, most countries have their own CPR systems that can transform the human readable plate characters into machine manipulatable representations. These CPR systems are usually dependant on country‟s plate style, format, character and signs included in the plate region. Although, there are commercially available CPR systems that are designed in an ambition to work on any country‟s plate, they are still affected by country specific plate features.

In Ethiopia, the development of intelligent transport system is in its early stage. In contrast, the number of vehicles on the road is increasing dramatically from time to time, which leads to difficulty of traffic management [26]. This indicates that it is time to use ITS technologies. One of these intelligent transport systems is CPR. CPR system is the first requirement for a number of applications that compose ITS. For instance in Ethiopia, a simple CPR system can be used in border crossing controlling systems, automated parking systems, and also entrance admission systems. However, studies done on Ethiopian car plate recognition systems are rare. Till now, there is only one published study that works on recognition of Ethiopian car plates [4]. The developed system in [4] works in too many constraints. For instance, the input car images for the system have to be taken under too close range camera distance, and front view camera angle. Besides, the images have to have high quality. Moreover, the system only recognizes small class of Ethiopian car plates, and also it is not automatic (i.e., it needs to be provided with car plate images instead of car images), which makes the system inadequate to be used in the above CPR applications.

Ethiopian car plate style differs from other countries plate in a lot of features. For instance, besides English characters, Amharic characters are also included in the plate. In addition, character‟s font, color, size and the way they put together is different from other country‟s car plates. Moreover, Ethiopian plates include additional shapes like circle inside the plate. Accordingly, CPR systems developed for other countries, and also commercially available ones cannot be readily used for recognition of Ethiopian car plates. So, this is where the motivation comes for this work: to have a CPR system that can work on Ethiopian car plates.

1.3 Objective of the study

1.3.1 General objective

The general objective of this study is adapting and/or developing an automatic CPR system that recognizes and identifies Ethiopian car plates.

1.3.2 Specific objectives

The specific objectives of this work include:-

- ✓ Reviewing related literatures on CPR techniques and methodologies
- ✓ Studying various features of Ethiopian car plates
- ✓ Adapting and/or developing methods for plate detection, character segmentation and character recognition tasks
- ✓ Creating a standard database of Ethiopian car plate pictures that can be used for further studies
- ✓ Developing full featured automatic CPR system for Ethiopian car plates using MATLAB^{TM}
- ✓ Evaluating the performance of the developed system using test images

1.4 Methodology

To achieve the objectives of this work, the following methodologies have been used:

Literature Review

Literature review is the major methodology that has been used in order to explore and study related works. Using this methodology a variety of CPR technologies, techniques, approaches, and also algorithms are investigated. This investigation is done for all modules that constitute CPR systems (i.e., Image acquisition, PD, CS and CR), and also various preprocessing tasks that add a significant performance change to the system are explored and studied in each of the phases. As well, various features of Ethiopian car plates: their types, characters included, dimension, font and color are studied. Finally, the selected techniques and algorithms are studied deeply, to find out, if they can be customized for Ethiopian car plate style and/or to get an idea to develop a new one.

Data Collection

In this work, data collection has been the toughest and also critical task of all that highly influences the successful completion of the work. Nevertheless, for both training and testing of the system, we managed to collect 350 pictures of cars that show part of their plate regions from various camera distance, angle, car motion, and illumination conditions.

In the data collection process, a lot of effort has been made to address all kinds of Ethiopian car plate types. Unfortunately, the reason that people are not willing and cooperative enough to be their car photographed has made its own impact on the success of the project, especially, in the testing phase. In addition, getting car pictures with regional plates has been challenging.

Tools

For the prototype development of the system, MATLAB™ Version 7.1.0.246 (R14) Service Pack 3 is used.

Testing and Evaluation

The developed system is tested on the collected pictures that are taken under real environment scenario. The performance of each sub-modules of the system is tested and evaluated alone and in groups.

1.5 Scope and limitation of the study

The scope of this work is limited to the following:

- ✓ The system recognizes only car plates registered in Ethiopia.
- ✓ The system recognizes only single car plate in a given input image.
- ✓ Recognition of partially occluded car plates, and also car plates that are affected by dirt is not addressed.
- ✓ The system is not tested on all existed Ethiopian plates, due to the problems that exist in the data collection process.

1.6 Significance of the study

As discussed earlier, CPR systems have a lot of applications. For instance, having a CPR system in Ethiopia can be used by the following applications:-

- Entrance admission controlling system in restricted areas,
- Border crossing controlling system (i.e. using CPR systems, vehicle"s plate number can automatically be detected and can be checked in criminal databases or in lost vehicles),
- Automated parking systems: CPR systems can be utilized in order to control how much time a vehicle parks by associating the detected plate number to time stamps,
- Traffic monitoring: as forensic for law enforcement to detect committed violations like red light disobedience

1.7 Organization of the thesis

The rest of this document is organized as follows. In the Second Chapter, various common image processing techniques and methodologies that have been used in different CPR systems for plate detection, character segmentation, and character recognition module are discussed. In the Third Chapter, the various features of Ethiopian car plates, and also techniques and algorithms used in the proposed system are discussed deeply. In the Fourth Chapter, the testing procedure and the developed system result is presented. In addition, evaluation of the system according to the testing result is discussed from different angles. On the last chapter, an overall conclusion about the system and its result is given. Furthermore, some recommendations for further continuation of this work are provided.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Since the first industrial automatic CPR system was introduced in the late 70‟s, a lot of approaches for CPR problems are developed. However, the research and development still continues. This is due to the growing demand for automatic car identification systems required in various CPR applications mentioned earlier, and the requirement for reliable car identification at different illumination conditions, presence of noise, and nationality specific features. Consequently, all these reasons have been a major driving force for the development of diverse CPR approaches. Besides, the competition to develop a CPR system that can recognize any country‟s car plate plays a great role, for introduction of novel algorithms and CPR approaches.

In this chapter, different image processing techniques and approaches that have been developed for plate detection, character segmentation, character recognition as well as preprocessing operations are discussed.

2.2 Common preprocessing techniques

Preprocessing tasks in CPR systems have an important role in the successful completion of plate detection, character segmentation and recognition processes. Hence, in CPR, the input image usually passes through a variety of preprocessing techniques to improve the image quality and prepare it to the next stages. Some of these preprocessing tasks include: edge detection, contrast enhancement, binarization, size normalization, orientation adjustment, and image enhancement (i.e., including noise reduction, de-blurring etc...). In this section common preprocessing tasks used in numerous CPR systems are discussed.

2.2.1 Edge detection

Edge detection is the process of detecting group of pixels that define boundaries between regions in an image, which helps with segmentation and object recognition [29]. This process is one of the common preprocessing tasks in CPR systems, where their plate detection module relies on edges of a plate like, Hough transform [7]. There are a number of edge detection tools that have been used in CPR systems for years, but the popular ones are canny [21] and sobel edge filters [5, 6, 12].

2.2.2 Binarization

Binarization [30] is the simplest image segmentation technique that segments image pixels into groups based on their intensity value. In the binarization process a pixel can be assigned either 0 (black) or 1 (white) depending on whether the intensity value of the pixel is above or below a threshold value, resulting with a binary image.

In CPR, binarization is frequently used in plate detection and character segmentation phases. In the plate detection phase, the process of binarization is usually used as a preprocessing task for binary operations, like morphological operations and connected component labeling. On the other hand, binarization is used in character segmentation to segment plate characters from the background pixels, which have higher intensity differences.

In binarization, the key problem is choosing the right threshold value to binarize an image. There are various proposed methods for selection of appropriate threshold value for a given image. In [8, 18], the popular threshold value selection method Otsu is used.

2.2.3 Orientation adjustment

In CPR, orientation adjustment [31] refers to the process of adjusting skew of plate regions to an angle that improves the outcome of the character segmentation as well as the character recognition process. Therefore, in most CPR systems, the output of the plate detection module (i.e., plate region) passes through orientation adjustment algorithms before proceeding to the character segmentation process.

Plate‟s orientation adjustment has two main consecutive steps: plate‟s skew angle detection and adjustment. To achieve the first process, a number of techniques have been used in CPR systems. For instance, in [9], Wen C. et al., used Hough transform based orientation adjustment method that detects plate‟s border lines using Hough transform, and use them to deduct the inclined angle of the plate. Similarly, in [8], plate‟s orientation angle is deducted from orientation of detected characters from the plate.

The second and last step of the orientation adjustment process is usually accomplished by performing a spatial transformation on the plate image, according to the provided plate orientation information from the first process. In the next section, one of the well known spatial transformation techniques used in our work is discussed.

Affine Transformation

In image processing world, changing geometrical structure objects in an image using a spatial transformation technique is nothing new. Spatial transformation in image processing refers to changing location of image pixels to a desired location. This process ends up in changing geometrical structure of objects in the image. The spatial transformation can be one directional transformation, where either the x or y-position of pixels is changed or it could be two directional transformation where both positions of a pixel changes.

There are various types of spatial transformations including affine, projective and polynomial. Each of these spatial transformation methods are chosen based on a desired geometrical transformation. One of these spatial transformations is affine transform.

Affine transformation is usually used to achieve translation, rotation, scaling or shearing. This transform is able in keeping straight lines to remain straight, and parallel lines to remain parallel, but rectangles become parallelograms (see Figure 2.1).

One of the popular affine transforms is shear transform, which is used when shapes in the input image exhibit shearing. This transform is defined with the following transformation matrix to perform 2D spatial transformation.

$$\begin{bmatrix} 1 & Sy & 0 \\ Sx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where, Sx and Sy specify the shear factors along the x and y-axis respectively. The last column of this kind of spatial affine transformation matrix is always $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

For a shear transform,

$$[\boldsymbol{Xin} \quad \boldsymbol{Yin}] = [\boldsymbol{Xout} \quad \boldsymbol{Yout} \quad \mathbf{1}] * \begin{bmatrix} \mathbf{1} & \boldsymbol{Sy} \\ \boldsymbol{Sx} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \textbf{(2.1)}$$

$$\boldsymbol{Xout} = \boldsymbol{Xin} - (\boldsymbol{Sx} * \boldsymbol{Yout}) \quad \textbf{(2.2)}$$

$$\boldsymbol{Yout} = \boldsymbol{Yin} - (\boldsymbol{Sy} * \boldsymbol{Xout}) \quad \textbf{(2.3)}$$

Where, *(Xin, Yin)* spatial location of a pixel in the input image and *(Xout, Yout)* are spatial location of the pixel *(Xin, Yin)* in the output image.

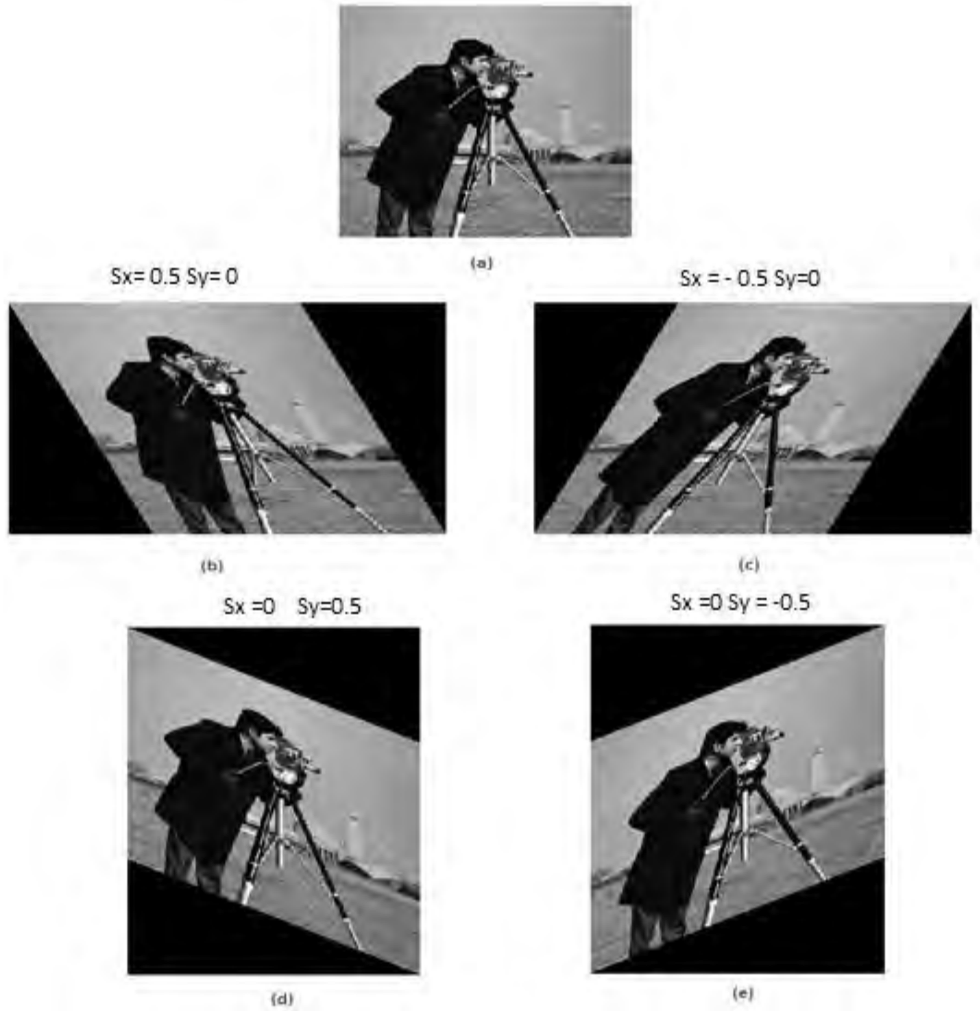


Figure 2.1: Outcome of a shear transform applied on intensity image *(a)* with different shear factors along the x-axis *(Sx)* and the y-axis *(Sy).*

2.2.4 Size normalization

Consistency in size of input images is an important factor for accuracy of every module in CPR. In a typical CPR system, size normalization can be done at most three times, at different stages. The first one is usually done before the plate detection process [4]. In this step, the raw input image is normalized into a manageable size that minimizes the computation cost of the plate detection process. The second normalization process is done, on the extracted plate image [18]. At this stage the normalization process helps in minimizing the influence of input plate images variation on the character segmentation process. The last normalization process could be done, as a preprocessing task for the CR module on the segmented character images [38].

2.3 Plate detection approaches

Unlike the other CPR modules (i.e., CS and CR) that use a particular method to carry out their task, the PD phase usually uses a series of techniques and algorithms to detect plate regions. Although, one of these algorithms is usually a base for all the other techniques/methods involved. This core algorithm is generally used at the early stage of the plate detection process to roughly locate potential plate regions. And, in order to achieve this task, this kind of algorithms take advantage of plate‟s and/or plate character‟s features to detect likely plate regions. The features related to a plate, include height-to-width ratio, color, texture, shape, and any unique plate signatures. The character features include: layout of characters, intervals that exist between characters and aspect ratio of characters are the main ones. In this section the commonly used PD techniques are discussed.

2.3.1 Morphology based approach

Mathematical morphology [33] is an image analysis technique that uses the concept of mathematical set theory for extracting meaning from images based on geometrical structures. There are a number of morphological operations that are used to understand the structure or form of an image. But, the two principal morphological operations are dilation and erosion [33].

Dilation allows objects to expand; potentially filling in small holes and connecting disjoint objects, while, erosion shrinks objects by eroding their boundaries. Other morphological operations such as, opening, and closing are special cases of these primary operations that are frequently used in morphology based CPR systems. In the next sections the morphological operation (i.e., closing) that is used in our work is briefly discussed.

Morphological closing operation is used to fill in holes and small gaps found in objects that are found in images. Technically, morphological closing of an image is a dilation followed by erosion using the same structuring element for both operations. Structuring element (SE) is an array that defines certain characteristic of a structure and features to measure the shape of desired objects in an image. The shape and size of the structuring element plays crucial role in image processing and is therefore chosen according to the condition of the image and demand of processing.

The closing of an image I by S (SE) is obtained by the dilation of A by S, followed by erosion of the dilation output by S [33]:

$$\boldsymbol{I \bullet S = (I \oplus S) \ominus S} \quad \textbf{(2.4)}$$

where, I is a binary or grayscale image, $\bullet$ is the closing operation, $\oplus$ is the dilation operation, $\ominus$ is the erosion operation and S is the structuring element.

Morphology based PD methods [11, 12], use a variety of morphological operations to locate possible plate candidate regions with a prior knowledge of plate"s shape, which can be represented by a structuring element. In [11], a series of binary and morphological operations are applied on the input car image to detect candidate plate regions. Phalgun Pandya and Mandeep Singh [11], first, filter the input image with a median filter to create a blurred image (to prevent over segmentation). Then they closed and opened the difference image that is computed by subtracting the image that resulted by opening the original input image from the image that resulted by closing the original input image with the same SE. Finally, the resulting image is binarized using global threshold value, and CCA is used to detect possible plate candidate regions.

This approach of PD can still end up detecting many objects that have a relatively similar shape as the plate. As a result, further post processing operations are usually used to distinguish the genuine plate region out of the candidates. In order to do this, plate features like width to height ratio, density of regions, and color analysis have been used in various CPR systems. As the study of Hadi Sharifi et al., [10] shows, this method has the lowest accuracy and execution speed, but it is the easier one to implement.

2.3.2 Hough transform approach

Hough transform [34] is a feature extraction technique, which is used to find instances of objects within a certain class of shapes from an image. It is most commonly used for the detection of simple curves such as lines, circles, and ellipses within a given image.

The simplest case of Hough transform is the linear transform for detecting straight lines. In image space, the straight line can be described as y = mx + b and can be graphically plotted for each pair of image points (x, y). In Hough transform, the main idea is to consider characteristics of a straight line not as image points (x1, y1), (x2, y2), etc., but instead, in terms of its parameters (i.e., the slope parameter m and the intercept parameter b). Based on that fact, the straight line y = mx + b can be represented as a point (b, m) in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameters m and b. Hence, to address this problem, Hough transform uses the following (Equation 2.6) polar coordinate parametric line equation that uses different pair of parameters, denoted ρ and Θ, Θ ∈ [0, π] in radians, and ρ ∈ [−D, D] where D is the diagonal of the image [35].

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{\rho}{\sin\theta}\right) \quad \textbf{(2.5)}$$

That can be rearranged to: -

$$\rho = x\cos\theta + y\sin\theta \quad \textbf{(2.6)}$$

Where Θ is the orientation of the line that passes through (x, y) with respect to the x axis and ρ is the length of a normal from the origin to the line (see Figure 2.2).

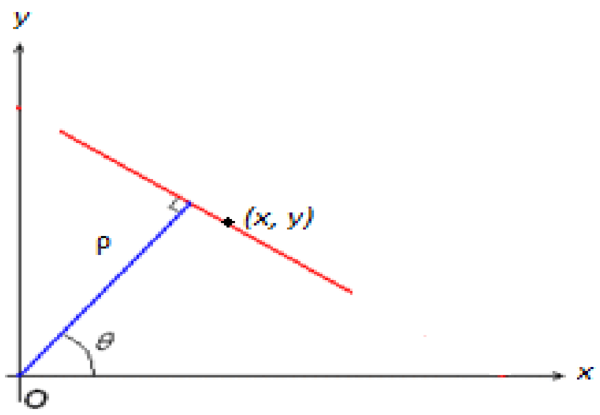


Figure 2.2: A line in xy-space described in terms ρ and Θ

For an arbitrary point on the image plane with coordinates, e.g., (x_i, y_i), the lines that go through it are the pairs (ρ,θ) with $\rho = x_i\cos\theta + y_i\sin\theta$, where ρ is determined by θ. This corresponds to a sinusoidal curve in the ρθ plane, which is unique to that point.

This means two points (x_i, y_i) and (x_j, y_j) in the xy-plane will generate two different curves in ρΘ-plane. And if these two points are collinear on the xy-plane (Figure 2.3(a)), then the two curves generated by these two points in the ρΘ-plane will intersect at a point (ρ', Θ') in the ρθ plane (Figure 2.3 (b)).

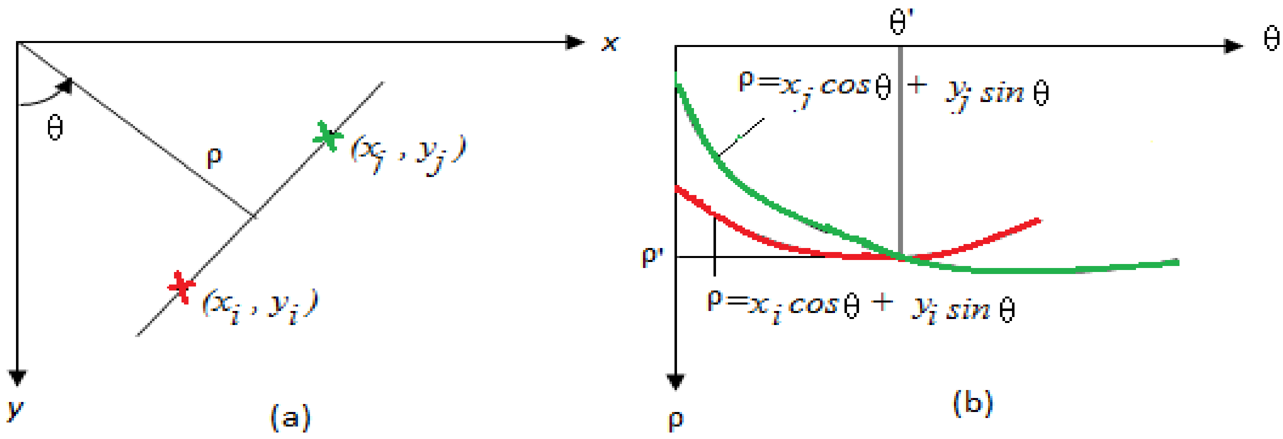


Figure 2.3: Hough transform: xy-plane point to ρΘ-curve representation

The Hough transform algorithm uses an array, called an accumulator, to detect the existence of lines. The dimension of the accumulator for linear Hough transform corresponds to quantized values for (ρ, θ). For each pixel in the image space and its neighborhood, the Hough transform algorithm determines if there is enough evidence of an edge at that pixel. If so, it will calculate the parameters of that line, and then look for the accumulator's bin that the parameters fall into, and it increases the value of that bin. By finding the bins with the highest values, typically by looking for local maxima in the accumulator space, the most likely lines will be extracted.

A number of CPR systems [7, 9, 13] has been using this method to detect edges of plate regions. As we know, a plate region has two pairs of parallel lines/edges: horizontal and vertical. Accordingly, what this method actually does is, it searches the entire image to detect these pair of parallel lines. Then a contour algorithm is used to find two interacted parallel lines from the detected lines. This step detects closed boundary objects that are possible plate candidates. However, this method has difficulty in extracting plate regions when the boundary of the plate is not clear or the images contain lots of vertical and horizontal edges like radiator grilles. In addition, according to the study of [10], the method has got higher computation time than any other PD approaches, which makes it inconvenient to be used in real time CPR systems.

2.3.3 Gabor transform approach

Plate regions have rich texture information, because of the pattern created, as a result of high intensity variations that exist between plate characters and the plate background. Thus, this

PD approach takes advantage of plate"s texture information to roughly locate potential plate regions.

In CPR systems, various techniques have been used to detect textural properties of plate regions. However, Gabor filters are the massively and frequently used texture analysis tool in CPR systems, because of their ability of multi-scale and multi-direction execution [10].

Gabor filter is a bandpass filter, which was originally developed for signal processing by Dennis Gabor. Nevertheless, the necessity of having texture analyzer that perceives textures as a human visual system leads to the ground breaking invention of J.G.Daugman [36]. Daugman discovered that simple cells in the visual cortex of human brains can be modeled by Gabor functions [21].

Physiological studies show that simple cells of human visual cortex are selectively tuned to a particular orientation as well as frequency, which help them to discriminate textures. Having this knowledge, Gabor filters are found to be the best local bandpass filters that are capable of imitating/representing the work of human visual cortex simple cells using 2D spatial and frequency domains.

In spatial domain, a 2D Gabor filter is a gaussian kernel function modulated by a sinusoidal plane wave. The family of Gabor function used in this study can be mathematically described as follows [32]:

$$\boldsymbol{r(x,y) = i(x,y) \otimes g_{f,\Theta}(x,y)} \quad \textbf{(2.7)}$$

where $\otimes$ denotes convolution, i is a gray scale image, r is the filter response and g is the gabor function tuned in specific frequency (f) and orientation (Θ):-

$$\boldsymbol{g_{f,\Theta}(x\,,y) = G(x',y')\ \sin(2\pi f x')} \quad \textbf{(2.8)}$$

where f is the frequency of the sinusoidal function and G is a 2D gaussian function described as:-

$$\boldsymbol{G(x',y') = e^{\left(-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)+\left(\frac{y'}{\sigma_y}\right)\right]\right)}} \quad \textbf{(2.9)}$$

where $(x',y') = (xcos\Theta + ysin\Theta, -xsin\Theta + ycos\Theta)$ are rotated spatial domain rectilinear coordinates, Θ is the orientation of the gabor filter and σ_x and σ_y are variances along x and y axes respectively.

By substituting equations in one another, we found a Gabor filter that acts as a bandpass filter with center frequency of f and a bandwidth controlled by σ_x and σ_y.

$$g_{f,\Theta}(x, y) = e^{\left(-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)+\left(\frac{y'}{\sigma_y}\right)\right]\right)} \sin(2\pi f x') \quad (2.10)$$

After locating possible plate regions using the Gabor filter, a thresholding operation is usually performed on the filter response to separate the high intensity regions (i.e., high texture information) from the low level ones.

2.4 Character segmentation approaches

Once the plate region is extracted, CS algorithms are used to segment out each alpha-numeric characters found in the plate. To achieve this task, a number of character segmentation techniques has been developed by different scholars. However, the frequently used approaches to character segmentation of CPR systems are projection method and connected component analysis.

2.4.1 Projection method

Projection method is one of the pioneer and extensively used method for car plate characters segmentation [15, 25]. The method uses vertical or horizontal projection of plate regions to analyze segmentation points. There are two kinds of projection methods: horizontal projection and vertical projection.

Horizontal projection method is used for line segmentation. This method is usually used on plates that have more than one row. The approach follows two basic steps. First, the horizontal projection of the plate is taken. Then the projection will be analyzed in order to separate each rows. This approach is highly dependent on the gap that exists between rows and the orientation of the plate.

On the other hand, we have a vertical projection method which is used to segment columns. The steps used to isolate each characters/columns are the same as the horizontal one, except that this one, depends on the gap that exists between characters instead of rows. This method is usually used for column/character segmentation on single row plates. However, it can also be used in double row plates after row separation is done.

The main challenge in projection method is locating the split point which is called valley. Segmentation lines are created at these valleys. This process can be worse by the effect of plate orientation, noises, and rivets that might connect separated characters. The process of locating valleys in the projection is called projection analysis.

In [15], Kumar et al., used vertical and horizontal projection analysis method to segment Indian plate characters prior to performing noise filtering and thinning preprocessing tasks on the input image.

2.4.2 Connected component analysis

CCA [39] is an image processing technique that provides Connected Component Labeling (CCL) operation and detected connected components by CCL analysis. The first stage of CCA (i.e., CCL) is an operation where groups of connected pixels (connected component) are classified as disjoint objects with unique identifiers. In order to accomplish this task, the CCL algorithm scans the image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values for gray scale image and 1-valued pixels for binary images.

Pixels connectivity is usually expressed using 4-neighborhood or 8-neighborhood connectivity ways. In 4-neighborhood connectivity, pixels are connected if their edges touch. This means pair of pixels are part of an object, only if they are both on and are connected along the horizontal or vertical direction. In 8-neighborhood connectivity, pixels are connected if their edges or corners touch. This means that if two pixels are on, they are part of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.

The later stage of CCA, which is detected connected regions analysis, is usually performed on CPR systems to identify the legitimate plate region from potential plate region candidates by CCL (i.e., if the CCL is used to detect possible plate regions) [11] or to validate plate characters detected by CCL from a provided plate image [16]. In either way, the analysis is done based on a pre-specified criterion that describes plate or character features respectively in each of the cases.

In these days, CCA becomes one of the widely used plate character segmentation approaches due to its relatively simple implementation, and its ability of detecting multi-direction and multi-scale objects. However, CC-based methods can group characters into a single component, if there is a connection between the characters, which usually happens because of low resolution noisy input images.

2.5 Character recognition approaches

The last phase in CPR is recognition of segmented alpha-numeric plate characters obtained from the CS module. Plate‟s character recognition is different from document character recognition in a number of ways. For instance, unlike document character recognition applications [23, 24], character‟s font, appearance and type is known in plate‟s character recognition. However, CPR systems generally operate on noisy and low quality images, in which illumination conditions frequently cause difficulties. Nevertheless, the recognition methods used for both of the problems are often the same.

In this section the two popular character recognition approaches [17]: neural network and template matching are discussed.

2.5.1 Neural network

An Artificial Neural Network (ANN) [28] is an information processing paradigm that is inspired by the way biological nervous systems, process information. It is composed of a large number of highly interconnected processing elements (neurons) working together to solve specific problem.

Recognition in neural network is a learning outcome. First, a classifier is trained using challenging and vast amount of data. Then the trained classifier will be applied on a new character to classify it, based on the acquired knowledge while learning. In [19], a multilayer neural network for recognition of characters is used to classify plate characters with a presence of 50% noise density. And the network classifies 95% of the characters correctly. In [20], Back-Propagated Neural Network is used for plate characters recognition, while [25] uses multilayer feed-forward network, that contains 158 neurons in the hidden layer and 40 neurons in the output layer is used to recognize Bangla characters found in Bangla car plates.

Neural network based CR methods has some advantages over template matching techniques, due to the fact that neural network trained classifiers are resistant to noises and some shape modifications of characters [19] unlike template matching techniques. The main drawback of this method is, the large amount of data needed for training classifiers [10].

2.5.2 Template matching

Template matching is one of the pioneer and intensively used object and character recognition techniques [4, 11]. This method has been used to recognize characters from a document as well as plate regions.

In template matching, individual image pixels are used as features. Classification is performed by comparing an input character image with a set of templates from each character class. Each comparison results in a similarity measure between the input character and the template image. One measure increases the amount of similarity when a pixel in the observed character is identical to the same pixel in the template image. If the pixels differ, the measure of similarity may decrease. After all templates have been compared with the observed character image, the character's identity is assigned as the identity of the most similar template.

These days a variety of template matching based recognition techniques like structural/syntactic and elastic template matching techniques are gaining popularity in character recognition world. These techniques focus on the primitive structures that make up a character. Structural features can be defined in terms of character strokes, character holes, or other character attributes. However, in plate characters case, it is difficult to identify the structural features for each character due to the various reasons we mentioned earlier that affect the input image quality, which directly affect characters appearance. As a result, these different types of template matching methods are rarely used for plate‟s character recognition.

For recognition to occur, the current input character is compared to each template to find either an exact match, or the template with the closest representation of the input character. To accomplish this, template matching techniques, use various computed values between the source and template images, for classification decision. Of these computed values, SAD (Sum of Absolute Differences), SSD (Sum of Squared Differences) and correlation are the popular matching scores [40].

For a template image (T) and source image (I), SAD, SSD and correlation values are calculated as [41]:

$$SAD(i,j) = \sum_{i=1}^{Irow} \sum_{j=1}^{Icol} |I(i,j) - T(i,j)| \tag{2.11}$$

$$SAD = \sum_{i} \sum_{j} SAD(i,j) \tag{2.12}$$

$$SSD = \sum_{i,j}\left(I(i,j) - T(i,j)\right)^2 \qquad (2.13)$$

$$corr = \frac{\sum_i \sum_j (I(i,j)-\bar{I})(T(i,j)-\bar{T})}{\sigma_I \sigma_T} \qquad (2.14)$$

where $\bar{I}$ *and* $\bar{T}$ are the mean values of I and T respectively, while σ_I *and* σ_T are the standard deviation of I and T that normalize the correlation value to lie between 0 and 1.

The first two equations (SAD and SSD) values absolutely depend on corresponding pixel values of I and T. On the other hand, the correlation value measures the degree to which the two images agree, not necessarily in actual value but in general behavior. For this reason, a template matching approach that uses correlation value for classification decision is usually preferable than the other values.

2.6 Related Work

Till now, there is only one published research paper that works on Ethiopian car plates [4], "Design and Implementation of Car Plate Recognition System for Ethiopian Car Plates", 2011.

The developed CPR system has two main components: plate detection and character recognition. In plate detection, first, the input image passes through two consecutive preprocessing operations, namely, gray scale conversion and median filter processes. Then the output image goes to a binarization process that uses a pre-set threshold value (140) to binarize every input image. Finally, CCA is used to detect potential plate regions from the binarized form of the input image.

After the plate region is detected and extracted, the binary version of the plate region image is passed to the character recognition system, where CCA and template matching techniques are used for character segmentation and recognition processes respectively.

The developed system is tested on 23 static snapshots of cars that are taken under too close range and front view (see Figure 2.4). According to the paper, the reason for this small size of testing data set is "because it was somehow difficult to convince the people the need to take the photo". Nevertheless, unit and integration testing are performed using the testing data set. The PD module results in 82.6% of detection accuracy, by correctly detecting 19 plate regions out of 23 input images. Similarly, the CR module results in 82.11%. Overall, the CPR system achieves an accuracy of 52.63%, by correctly identifying information of 10 plates out of 19.

Figure 2.4: Sample input images used in [4]

So overall, from the information presented in the document we conclude that the system is developed for:

- Ethiopian plates that contain 10 characters
- Ethiopian plates that have plate structures like (Figure 2.4)
- High quality, close range, front view and constant camera distance input images

In addition, as presented in the document, the accuracy of the system is (52.63%), which is computed with respect to the number of successfully detected plate regions by the PD module (i.e., 19), instead of the total input size for the system (i.e., 23). So, this shows that the developed system considered plate images to be provided, instead of car images.

CHAPTER THREE

THE PROPOSED SYSTEM

3.1 Introduction

Ethiopian plates can be classified into 13 classes based on the category of the vehicle owner and the vehicle's purpose (see Table 3.1). Each class can be identified by its code, which is either numeric or character. Besides, most of the classes vary in foreground color, even though they have common white background color except police vehicles (i.e. yellow background).

Table 3.1: Ethiopian car plate classes

Number Code	English Code	Amharic code	Classes	Foreground color
1	-	-	Taxi	Red
2	-	-	Private	Blue
3	-	-	Trade	Green
4	-	-	Governmental	Black
5	-	-	Non-governmental	Orange
-	CD	ኮዲ	Diplomatic	Black
-	AU	አህ	Africa Union	Light Green
-	AO	ዕድ	NGO	Orange
-	UN	የተመ	United Nation	Light Blue
-	-	ፖሊስ	Police	Black
-	-	የዕለት	የዕለት	Red
-	-	ተላላፊ	ተላላፊ	Light Blue
-	-	ልዩ ተ	ልዩ	Red

On top of the above classification, Ethiopian plates are further classified into 11 regional classes. Each of these regions can print their own plate of code 1-5, 'የዕለት', 'ተላላፊ' and 'ልዩ'. The rest of the plates are controlled by the federal authority. Consequently, a given plate number can occur repeatedly in various regions and also in a number of classes (Table 3.1). Thus, identifying plate's number without the code and/or the region does not provide enough information in identifying a vehicle. To identify a vehicle's region, Ethiopian plates include character codes both in Amharic and English language (see Table 3.2).

Table 3.2: Ethiopian car plate's region codes

No.	Region	English Code	Amharic Code
1	Addis Ababa	AA	አ አ
2	Amhara	AM	አ ማ
3	Afar	AF	አ ፋ
4	Benshangul Gumuz	BG	ቤ ጉ
5	Dire Dawa	DR	ድ ሬ
6	Gambella	GM	ጋ ም
7	Harar	HR	ሐ ረ
8	Oromiya	OR	ኦ ሮ
9	Somali	SM	ሶ ማ
10	Southern Nations and Nationality People	SP	ደ ሕ
11	Tigray	TG	ት ግ

Moreover, Ethiopian plates differ in basic features, such as dimension and plate format. The plates exist in five primary dimensions: 460x110, 280x200, 305x155, 340x200 and 520x110mm. And also, the plates have two basic formats: single and double row plates.

Figure 3.1: Various Ethiopian plate style

3.2 Architecture of the system

The proposed system has three major modules: Plate Detection, Character Segmentation and Recognition. The system takes car image as an input. Then, it applies a series of various methods and techniques of plate detection, character segmentation and recognition algorithms on the input image. Finally, it stores the car identification number, region/category and code (if exists) in a flat file in a way that any further application can read and understand it. Figure 3.2 shows the architecture of the proposed system:

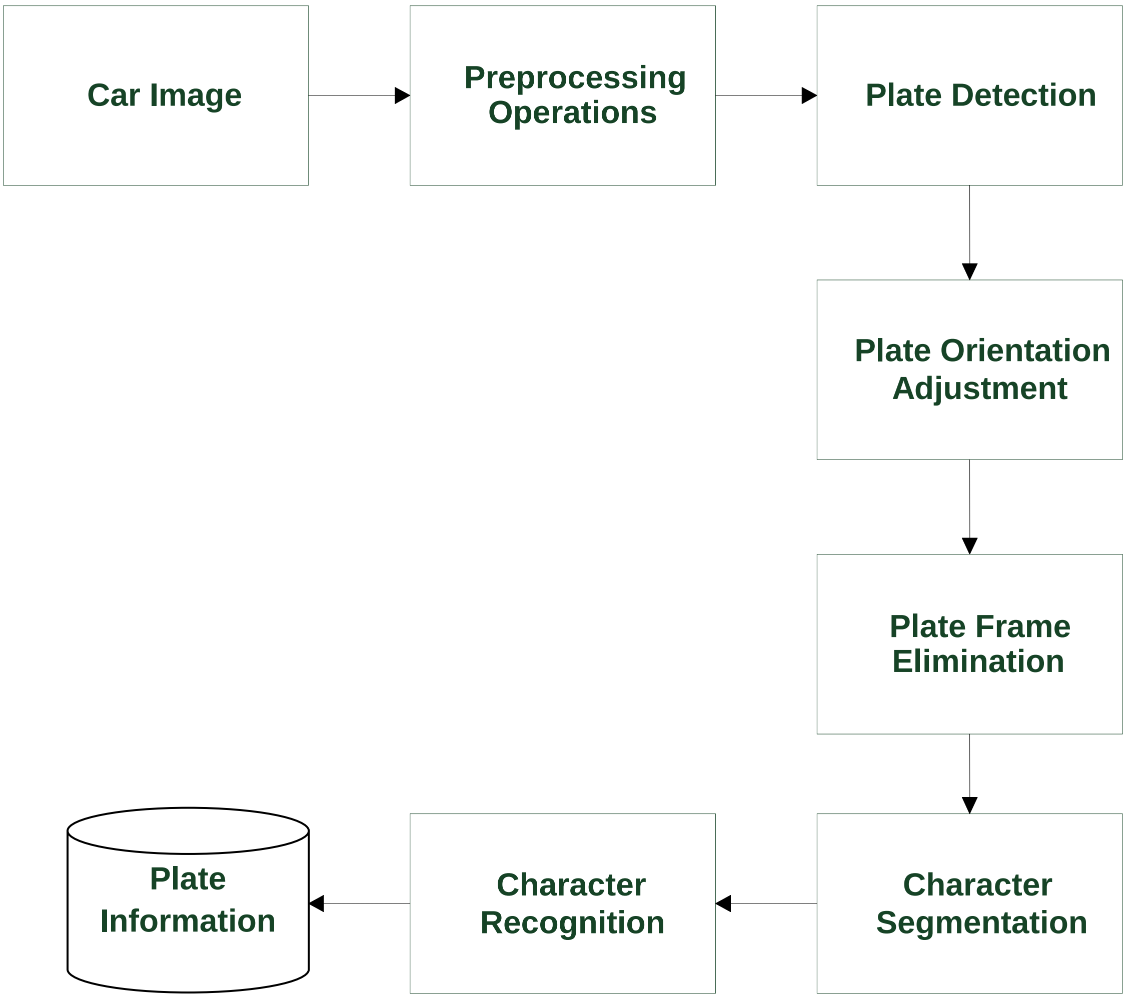


Figure 3.2: Architecture of the proposed system

3.2.1 Plate Detection

This module of CPR involves detection and extraction of plate region from a given car image. This process is the most critical phase of the whole system, as the execution of the other modules of the CPR (i.e., CS and CR) depends on its outcome. Thus, a lot of effort is taken into account in selecting PD techniques and algorithms that compromise the tradeoff between accuracy and computation time.

In the proposed system, a series of vital techniques are used to detect plate regions: *Gabor filter, Morphological closing operation and Connected Component Analysis (CCL)*. Moreover, prior to performing these algorithms, the input image passes through some pre-processing tasks, namely, size normalization and image type conversion in support of the upcoming PD techniques. Figure 3.3 shows the architecture of the plate detection module:

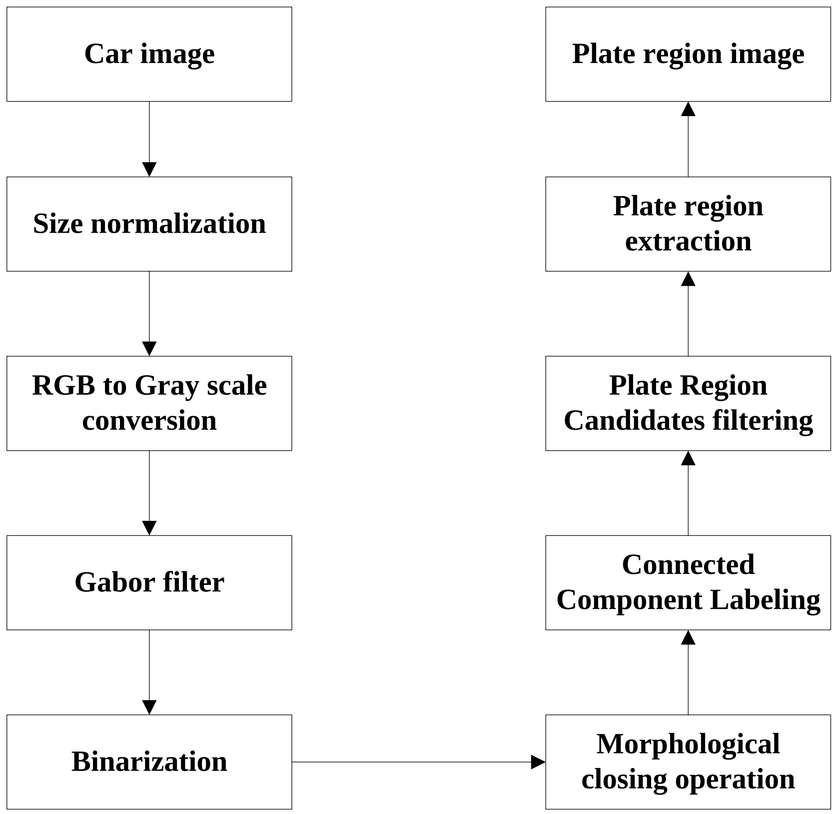


Figure 3.3: Architecture of the plate detection module

3.2.1.1 Pre-Processing

Size Normalization and Gray Scale Conversion

About 350 colored (RGB) car pictures are captured using a digital camera with a resolution of 12.1 mega pixel. The pictures are taken from different angle, distance, motion and illumination conditions. The dimension of most of the input images varies from one another. As a result, before going further, the pictures are normalized to the standard dimension (i.e., 640 x 480) [2] to create consistency in input images size.

Once the image resizing task is done, the image type (i.e., originally RGB) of the inputs is converted to intensity (gray scale) image to reduce the size of the input image array by 2/3. This action radically reduces the processing time of the entire system.

In an intensity image a spatial location in an image shows the intensity of the pixel at that location. Intensity/value of a pixel ranges from 0 (black) – 255 (white), depending on the pixel‟s brightness.

To convert the RGB input images to their equivalent intensity images (see Figure 3.4**)**, the following standard equation [20] is used:

$$\boldsymbol{Ig(i,j) = 0.299 * R + 0.587 * G + 0.114 * B} \quad \textbf{(3.1)}$$

where Ig is the gray scale image, i and j are particular array indices, and R, G and B are the Red, Green and Blue components of a pixel at position (i,j) in the RGB image.

(a)

(b)

(c)

Figure 3.4: Sample normalized RGB input images and their equivalent intensity images

3.2.1.2 Plate detection and extraction

Gabor filter

The proposed PD module focuses on taking advantage of one of the popular features of plates (i.e., texture), to locate potential plate regions. As we discussed earlier, plate regions have rich texture information, because of the pattern created, as a result of high intensity variations that exist between plate characters and the plate background. Accordingly, in this work, we used Gabor filter to highlight regions with high texture information, which are found within the provided grayscale image. Through this action potential plate region locations will be located.

Accordingly, the provided grayscale image (*i*) is convolved with the 2D Gabor filter (g) of **(**see Equation 3.2**)**.

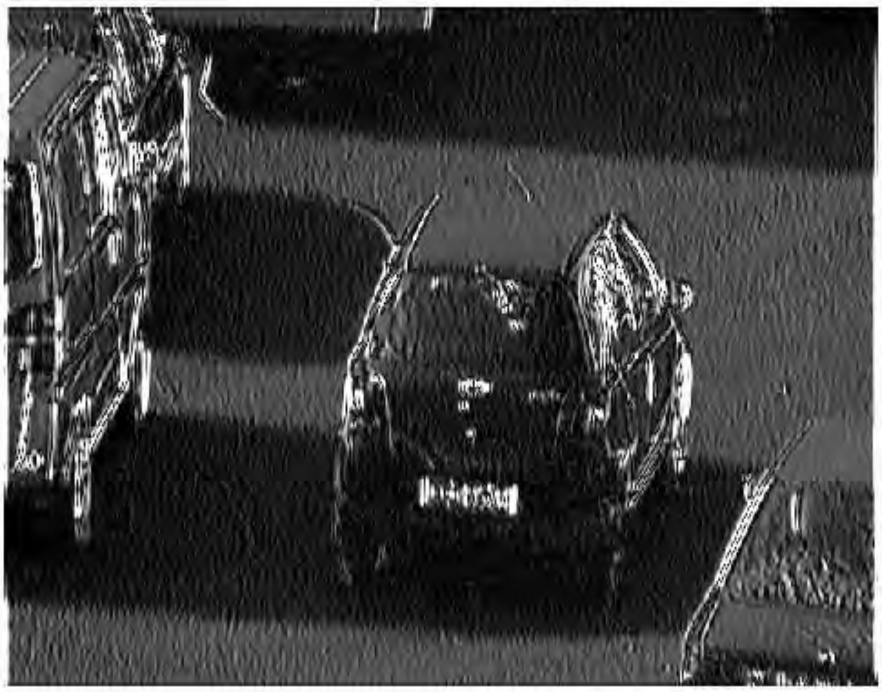
$$\boldsymbol{r(x,y) = i(x,y) * \; e^{\left(-\frac{1}{2}\left[\left(\frac{x'}{\sigma_x}\right)+\left(\frac{y'}{\sigma_y}\right)\right]\right)} \; \sin(2\pi f x')} \qquad \textbf{(3.2)}$$

where $x' = x\cos\Theta + y\sin\Theta$ and $y' = y\cos\Theta - x\sin\Theta$

The parameters (f, Θ, σ_x and σ_y) for the Gabor filter are selected by performing a simple experiment on the actual data, and analyzing the filter responses for different parameters. Figure 3.5 shows the Gabor filter response performed on *Figure 3.4 (a), (b)* and *(c)* intensity images respectively.

(a)

(b)



(c)

Figure 3.5: The Gabor filter response

Binarization and morphological operations

Binarization

In the Gabor filter response (Figure 3.5) high values indicate regions that have high texture differences, one of which is the plate region. Thus, binarizing the filter response separates out higher intensity regions like the plate, form other regions that have low intensity value.

The threshold value for this work is selected by the *'graythresh()'* function of MATLAB™ that uses *Otsu* method to come up with appropriate threshold value for a given image.

Morphological Operations

Before proceeding to the next CC-detection procedure, first, we performed morphological closing operation on the outcome of the binarization process, in order to merge individual character objects of the plate and form a solid rectangular plate region. This operation prevents the plate region, from being detected as a set character objects by the next CC-detection procedure.

Once the closing operation is done, small objects removal is performed on the output of the closing operation, to ease the work of the next CCL procedure by eliminating a number of small connected objects that do not need to be detected as well as processed by the CCL. Figure 3.6 shows the result of the binarization, closing and unwanted small object removal operations applied on the Gabor filter response (*Figure 3.5(a))* respectively.

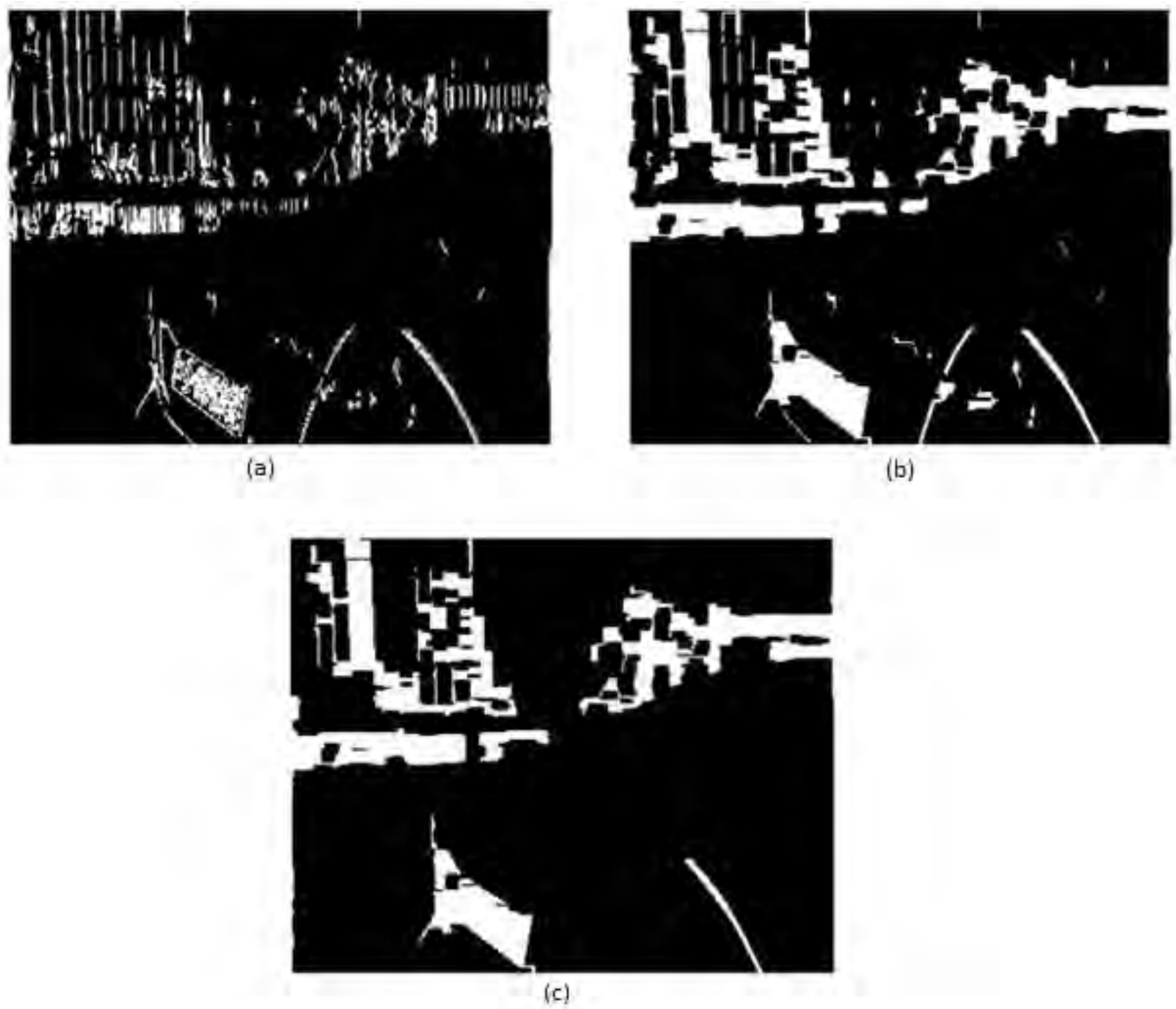
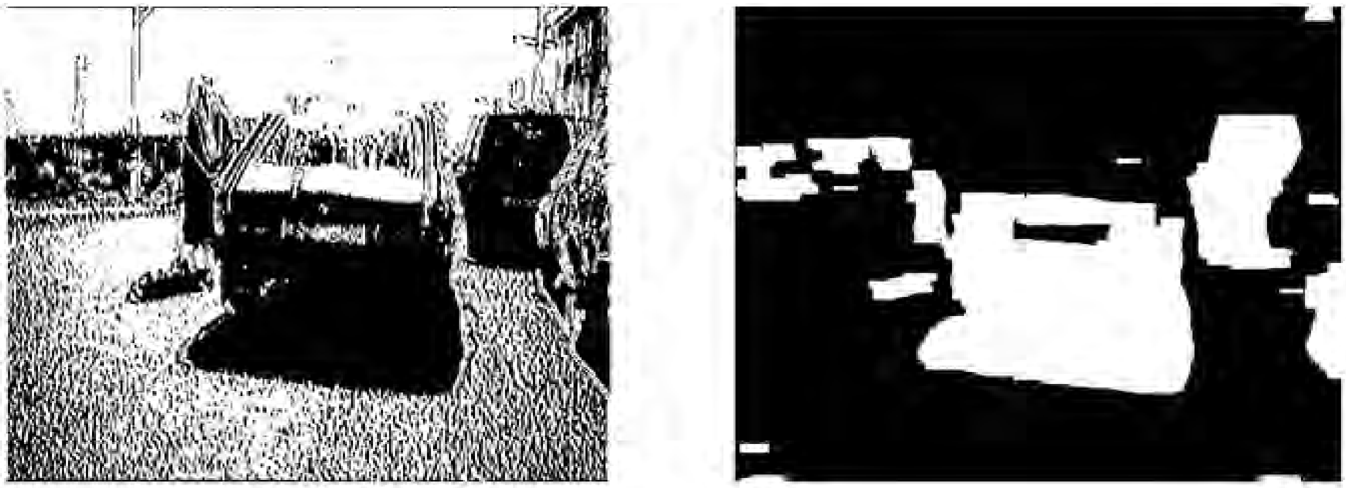


Figure 3.6: The response of binarization process (a), the morphological closing operation (b) and unwanted objects removal operation (c) performed on *(Figure 3.5(a)).*

As well Figure 3.7 shows the response of the binarization and morphological closing operation performed on the Gabor filter response images (*Figure 3.5 (b) and (c)*) respectively.



(b)

(c)

Figure 3.7: Response of the binarization and morphological closing operation performed on *Figure 3.5 (b)* and *(c)* respectively.

Connected component labeling (CCL)

In this section CCL is used to detect connected components/objects (one of which is a plate region) found in the binary image provided by the last operation. For this particular case, 8-neighborhood pixels connectivity is selected as a criterion to label objects. Once objects are labeled, acceptable aspect-ratio and density values of plate regions are used to minimize set of candidate plate regions. Finally, the region with the acceptable number of character with expected characters size, is selected as, the genuine plate object (see Figure 3.8).

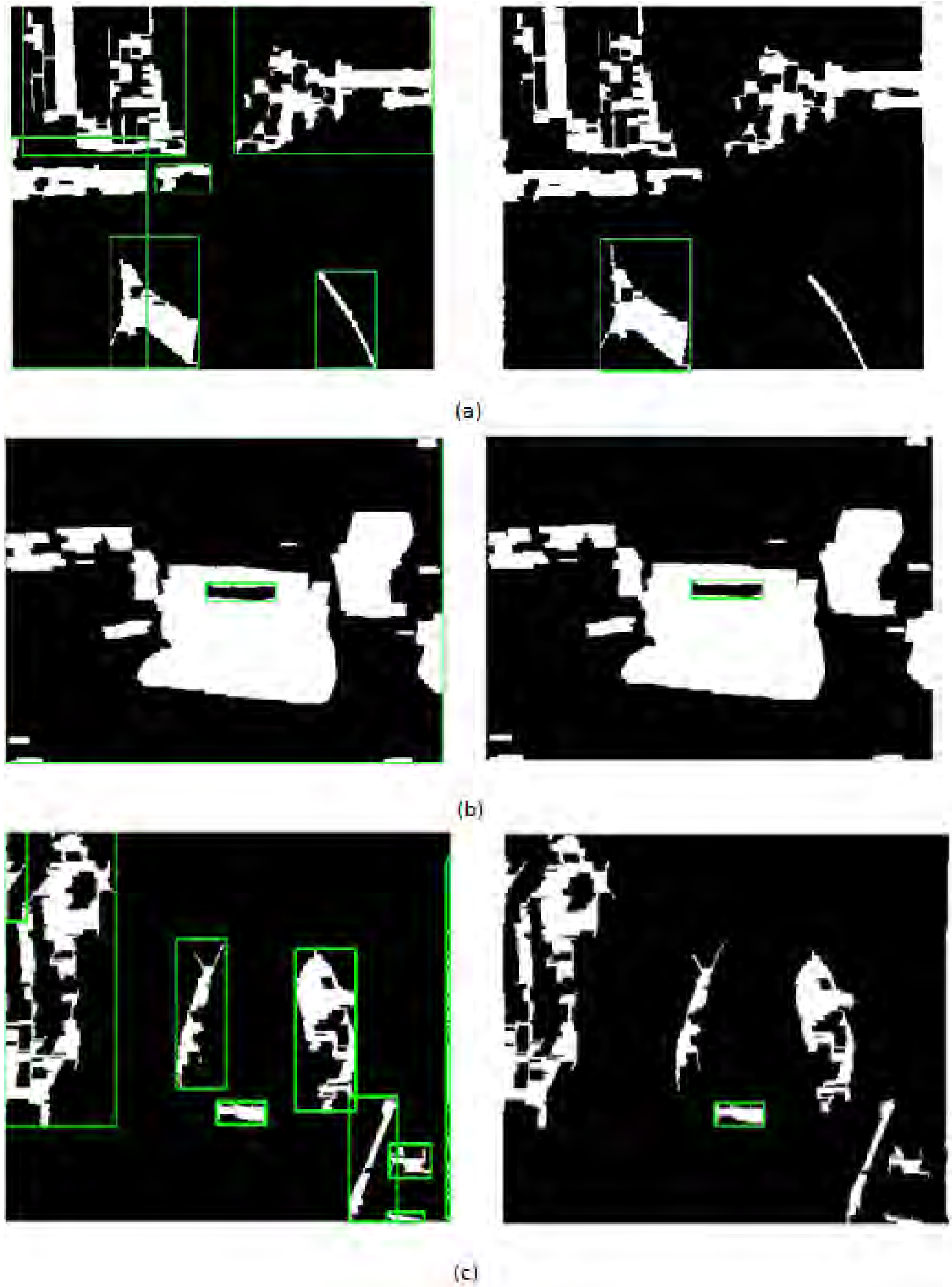
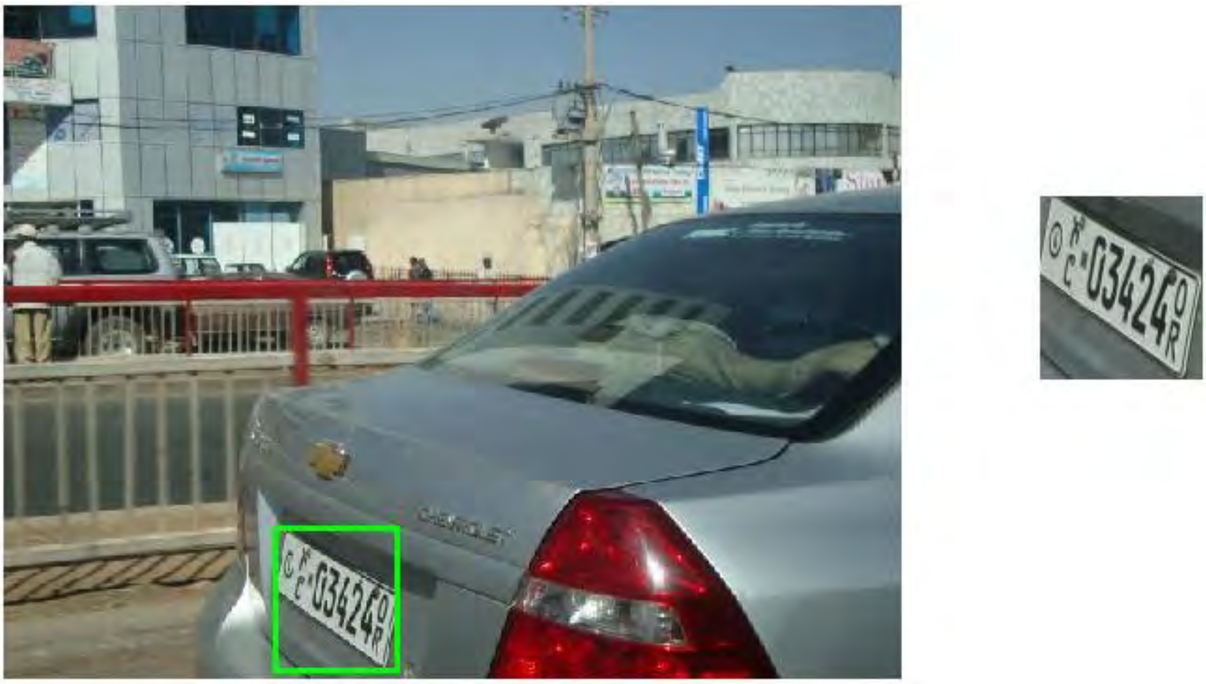


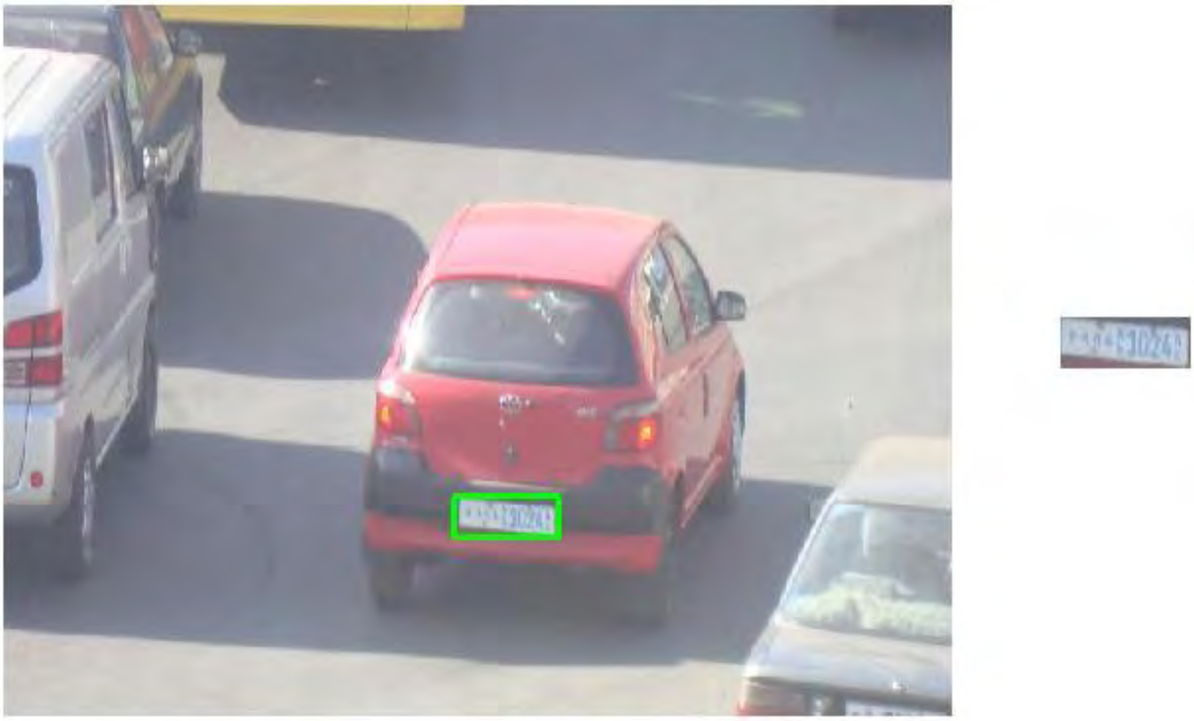
Figure 3.8: The outcome of the CCL and plate candidates filter process performed on *Figure 3.6(c)* is shown at (a) and *Figure 3.7 (b and c)* is shown at (b) and (c) respectively.

The last task of the plate detection phase is plate region extraction. This task is accomplished, by simply cropping the normalized RGB input image (Figure 3.4) at a location where the plate region is detected (see Figure 3.9).



(a)

(b)



(c)

(d)

Figure 3.9: Sample RGB input images and the result of the plate detection module

3.2.2 Character segmentation

This module is dedicated to detecting and segmenting each alpha-numeric characters found on the extracted plate image. The module comprises three consecutive core processes: Pre-Processing, Segmentation and Post-Segmentation. Figure 3.10 shows the architecture of the module.

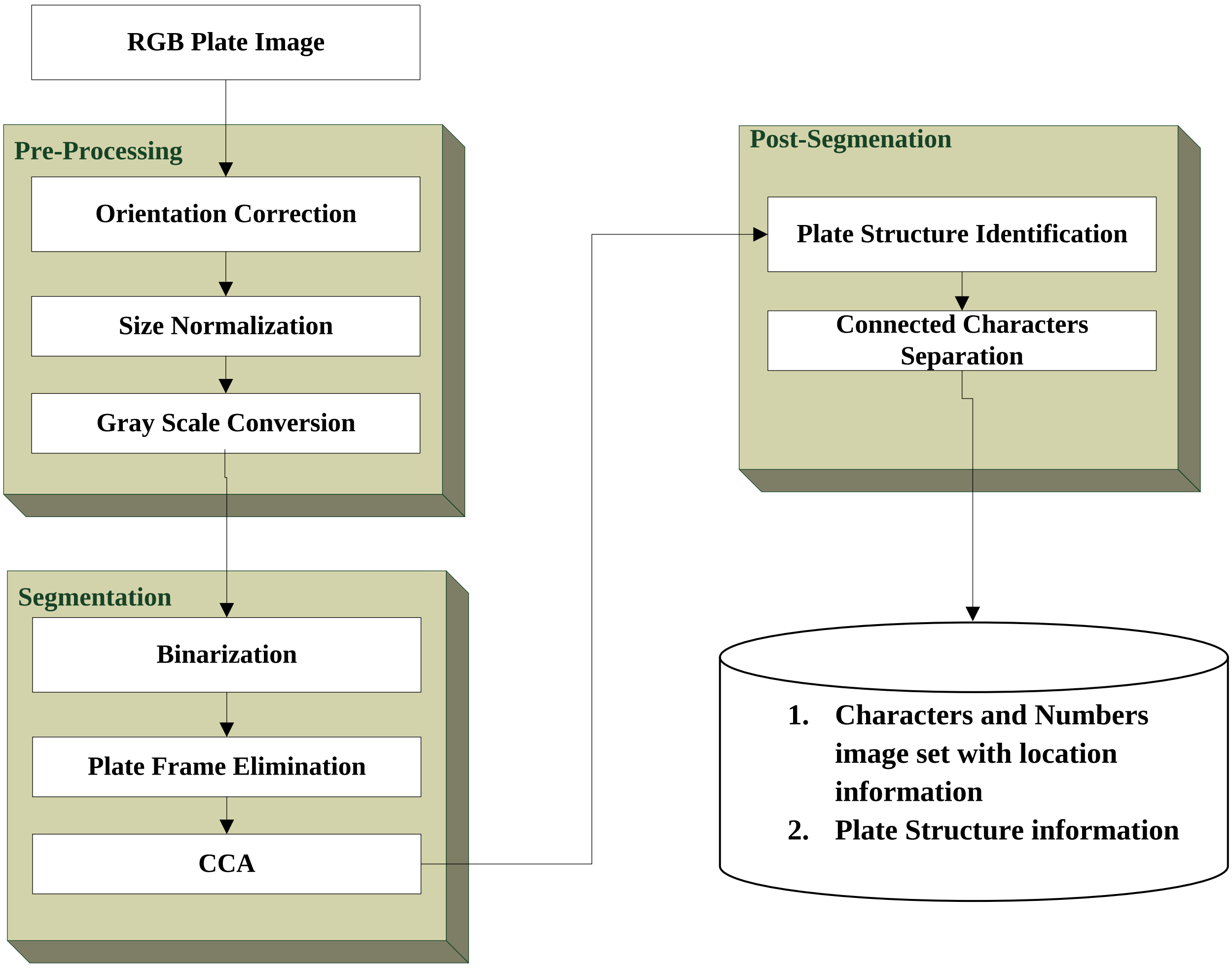


Figure 3.10: Architecture of the character segmentation module

3.2.2.1 Pre-Processing

In this pre-phase of the CS module, the input plate image (RGB) passes through orientation adjustment, gray scale conversion and size normalization processes, for the sake of increasing the accuracy and also decreasing the computation time of the character segmentation process.

Orientation adjustment

The principal Ethiopian plate shape is a rectangle, but quite often the detected plate regions by the PD module are rotated by some angle, which plays a sever role for the failure of character segmentation as well as character recognition processes. Thus, an orientation adjustment process is applied on the plate region image before any further task.

In plate's orientation adjustment two key steps are involved: finding out the slope of the plate and correcting the plate orientation according to the identified slope information.

For this system, linear Hough transform for line (i.e., plate borders) detection, followed by a spatial transformation (i.e., shear transform) that uses the slope of the longest detected as a shear factor is used to correct orientation of plates. Figure 3.11 shows the components of the orientation adjustment module.

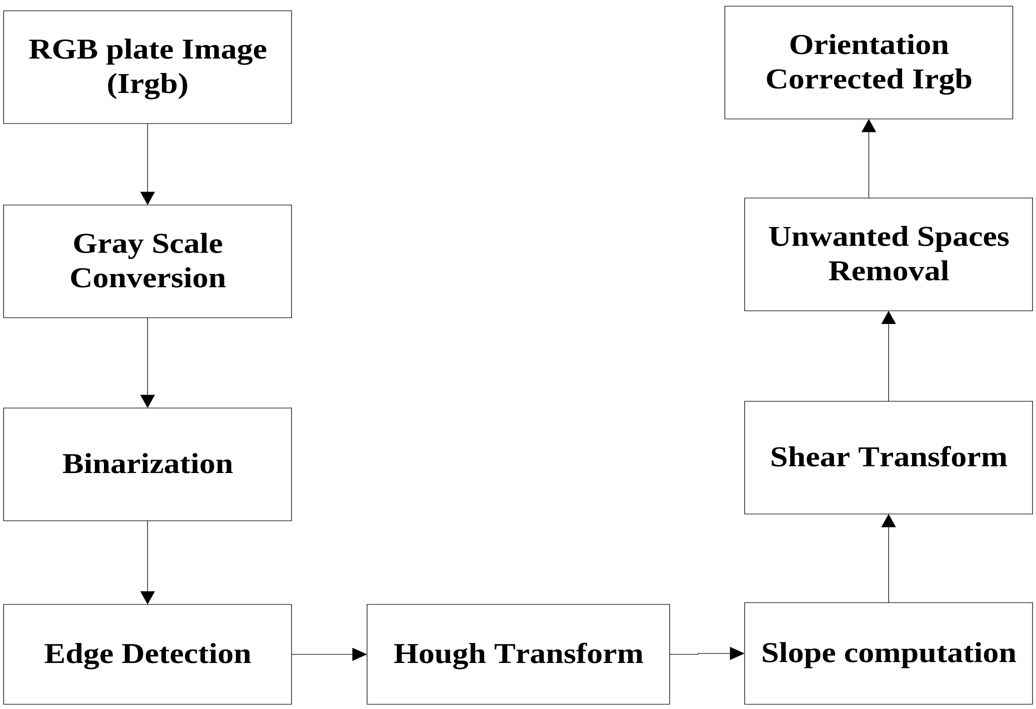


Figure 3.11: Components of the orientation adjustment module

Hough Transform

We started the orientation adjustment process by converting the input RGB plate image into gray scale image. Then canny edge detector algorithm is applied on the gray scale image to keep edge pixels while suppressing the others, and prepare the image to the next stage (i.e., Hough transform). Figure 3.12 shows sample input plate images of the CS module and their corresponding edge image.

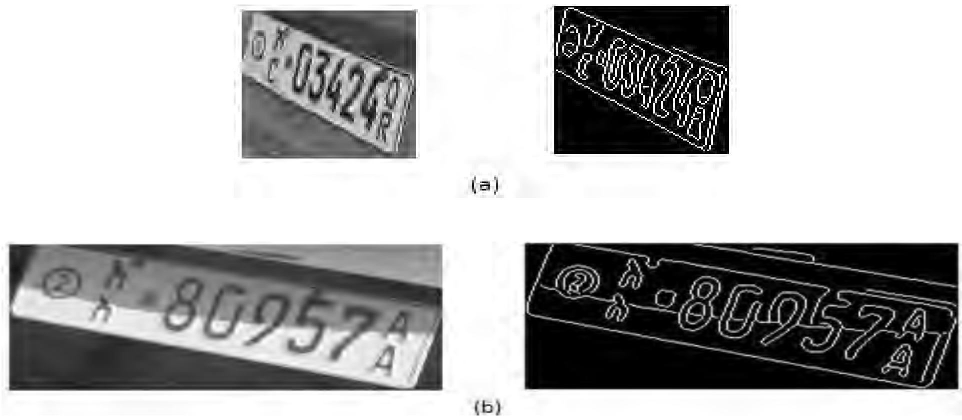


Figure 3.12: Sample gray scale plate images and the response of the canny edge detector

As we discussed earlier in chapter two, the first step towards correcting plate's orientation is, finding out the plate's slope. To achieve this task a lot of methods have been proposed [8, 9]. However, for our work, we used linear Hough transform method to detect plate borders. And then we computed the slope of the longest detected line (often plate borders) from the edge version of the plate region (see Figure 3.12) and considered it as the slope of the plate.

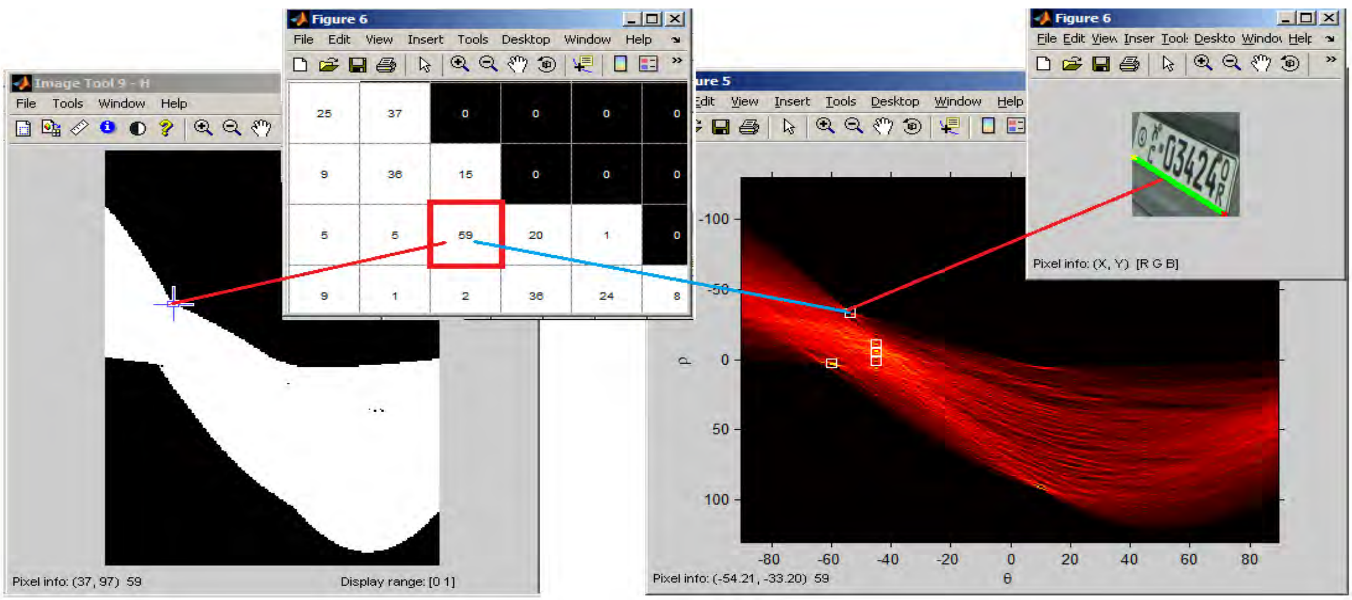


Figure 3.13: The Hough accumulator array (ρ x Θ) generated for *Figure 3.12 (a)*

Figure 3.13 presents the Hough accumulator array generated for the binary image (*Figure 3.12 (a)*); as gray scale image (shown at the left side); as a raster image (on the right side) where the white squares on it shows highest peak values that are potential lines in the input image (shown at top right corner of the figure); and as an array (ρ x Θ) of values (on the top left corner) where value of a cell (ρ‷, Θ‷) represents the number of collinear points/pixels found on the input image, where their curves in the Hough space intersect at (ρ‷, Θ‷). Figure 3.14 shows four strong detected lines from *Figure 3.12 (a)* and *(b)* edge images respectively by linear Hough transform.

Figure 3.14 Response of linear Hough transform, labeling the longest line red.

Spatial Transformation

The last and second step of the orientation adjustment process is, applying spatial transformation on the plate image based on the plate's orientation information. So, first we have to get the actual incline angle or in our case we use the slope of the longest detected line by Hough transform. Slope of a line is computed as:

$$\textbf{Slope}=\frac{y2-y1}{x2-x1} \qquad \textbf{2.6}$$

Where, (x1, y1) and (x2, y2) are the spatial locations of the starting and ending points of the line respectively.

In this work, to adjust plate's orientation we performed vertical shear transform based spatial transformation that uses the slope of the longest detected line as shear factor along the y-axis to change y-position of pixels. On the other hand, no spatial transformation is done along the x-axis. The shear matrix used in the shear transform is presented as:

$$\begin{bmatrix} 1 & -slope \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Figure 3.15: Presents the outcome of the spatial transformation (i.e., Shear transform) on (b) performed on *(a).*

As we can observe from the outcome of the spatial transformation (*Figure 3.15 (b)*), only the y-position of pixels is changed, which results in horizontal orientation corrected plate region.

When spatial transformation is performed, there are often pixels in the output image that are not part of the original input image, which are called a fill value, usually 0. So, to complete the orientation adjustment process the final step is extracting the actual plate from the spatially transformed image (*Figure 3.15(b))*. Thus, to achieve this task, we perform CCL operation on the outcome of the morphological closing operation (*Figure 3.16 (e))* done on the spatially transformed binary image (*Figure 3.16 (d))*, resulting in *Figure 3.16 (f)).*

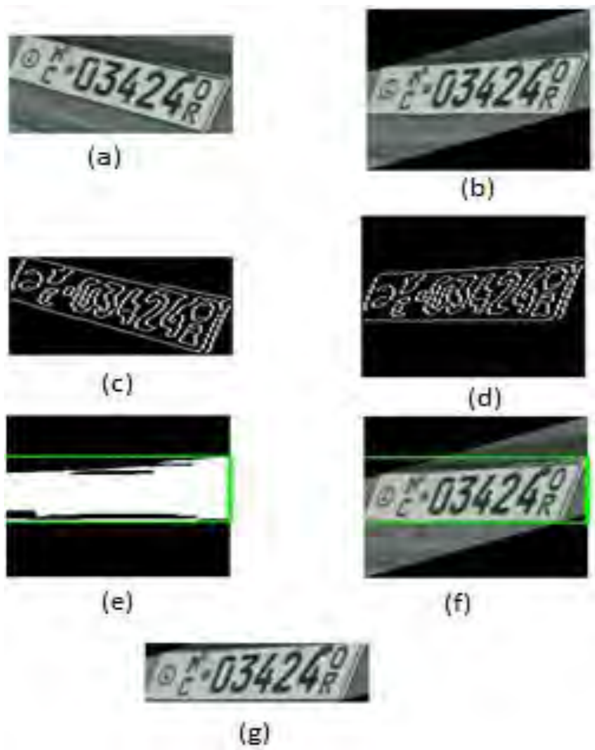


Figure 3.16: Result of the orientation adjustment module

Size normalization

Consistency in dimension of input plate images is an important factor for the performance of the character segmentation process. Especially, in Ethiopian's plate case, where there are a number of primary plate dimensions. Other than the primary dimension difference, plate region dimension difference could occur, because of the distance variation that exists between the camera and the vehicle while taking the pictures. Accordingly, it is crucial to normalize detected plate regions to uniform scale, in order to make the segmentation process less influenced by character's size variation. In this study 160*100 dimension is found to be convenient for the character segmentation process.

3.2.2.2 Segmentation process

The segmentation process has two core phases. In the first phase, the background image is segmented from the characters through binarization process. And then, CCL is used to detect group of pixels that form individual characters. But, before the later phase is performed, plate's frame elimination process is applied on the binarized image in order to increase the accuracy of the CCA process.

Plate's Frame Elimination

Plate‟s frame elimination is one of the critical and common pre-processing tasks of plate characters segmentation. Existence of plate‟s frame has a huge impact on the character segmentation process, especially, if plate characters are connected with the plate‟s frame, which happens almost all the time. Accordingly, we eliminate plate‟s frame by using the MATLAB™ function *‚imclearborder'* that uses morphological reconstruction method to remove borders of images. Figure 3.17 shows sample instance plate images obtained from the last operation and the output of *‚imclearborder'* function performed on them.

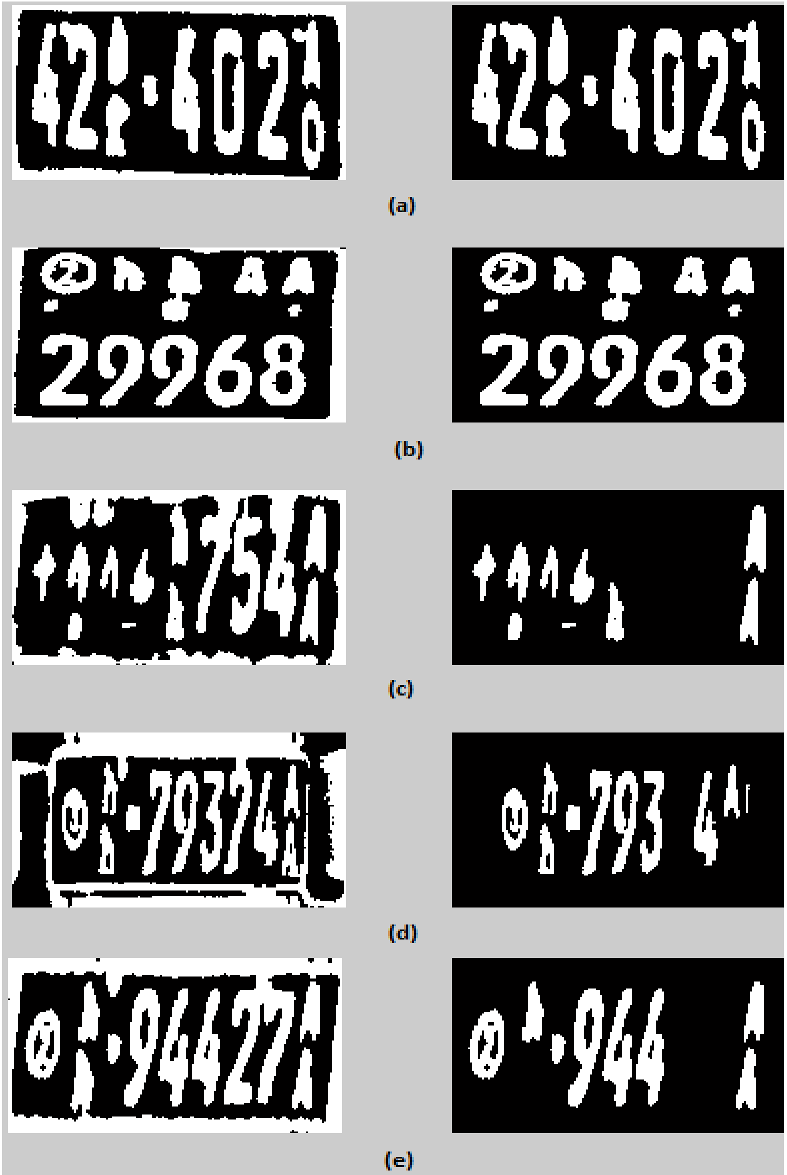


Figure 3.17: Result of the border elimination function *'imclearborder'*

As we can observe from Figure 3.17, the '*imclearborder*' function perfectly eliminates plate's frame/borders. However, there is one problem, if any of the plate's characters are connected with the plate border with noise, rivets (i.e., the usual reason) or any other reasons, then those characters will be considered as part of the border. As a result, as shown in *Figure 3.17 (c, d and e)*, these characters will also be eliminated from the image like the border. Hence, further operations are needed to keep the characters and eliminate the borders as the same time (see Figure 3.18).

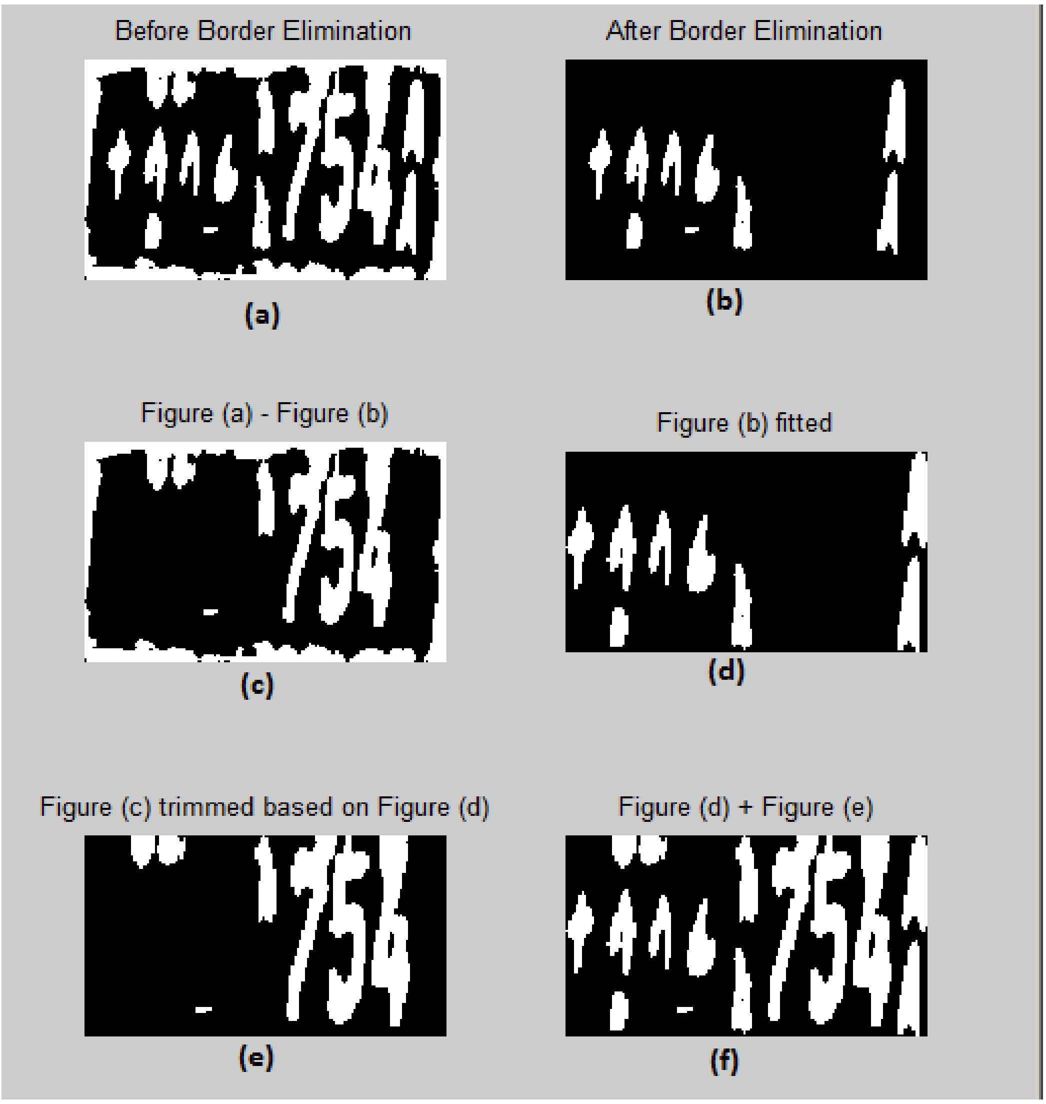


Figure 3.18: The post-frame elimination process done on (a), resulting in (f)

So, this is what we did to keep the characters and eliminate the borders at the same time. First, we subtract (b) from (a) to get the border image (c). Then, we removed the unwanted spaces from (b), which were the location of the borders (d). After that, we trimmed (c) based on the information we get while fitting (b) (i.e., how many number of rows are removed from top and bottom side of the image and as well number of columns removed from the left and right side). This task eliminates the actual border and keeps the characters that are connected to it (e). Finally, we add (d) and (e) to get (f), which contains all the characters and no borders. Figure 3.19 shows more instance of plates that benefit from these post-frame elimination operations.

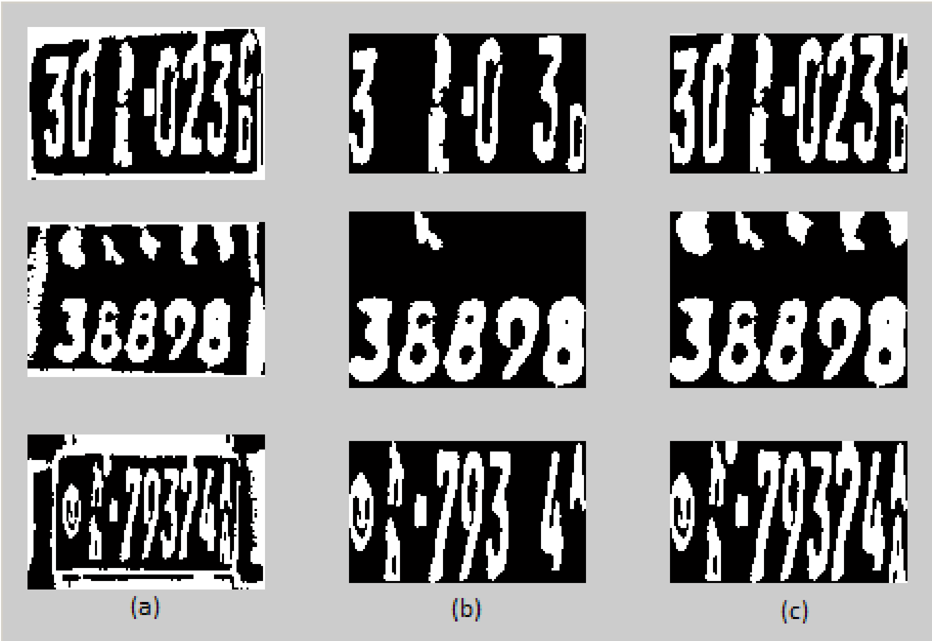


Figure 3.19: (a) instance of input plate images, (b) the output of the '*imclearborder'* function applied on the corresponding images of (a) and (c) the final outcome of the plate's frame elimination module

Character Segmentation

Now, it is time to detect and isolate each characters found on the plate. To accomplish this, we chose CCA method for so many reasons. First, this method is independent of plate's as well as character's orientation, which is a critical issue in other character segmentation methods. In addition, this method is independent of character's location, which is important for our case, because of the fact that Ethiopian plates do not have common number of characters as well as layout. Figure 3.20 shows some of the variety of Ethiopian plate structures.

Figure 3.20: Various Ethiopian car plate characters layout

The CC-detection method used here is similar to the method we previously used in the plate detection process, except the pixels connectivity used here is 4-neighborhood, which is adequate to form a character.

The CCL operation detects every group of connected pixels found on the input image that might be character or not. Accordingly, detected objects filtering process is needed to identify genuine plate character objects. Thus, we used some acceptable plate character's features, as character's width, height and area to filter the right objects (see Figure 3.21).

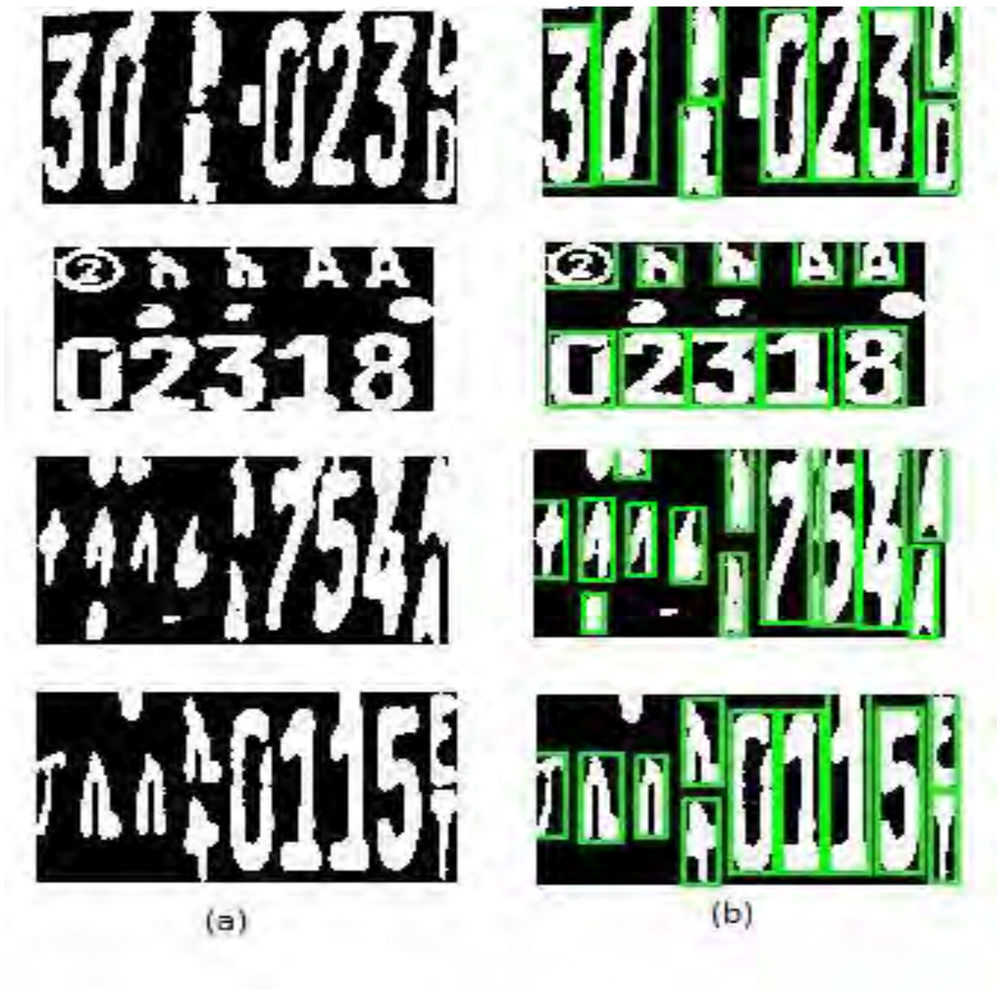


Figure 3.21: The outcome of CCA (b) performed on the corresponding binary images of (a)

3.2.2.3 Post Segmentation

In this post segmentation stage of the character segmentation phase, connected characters and numbers separation is performed with a prior knowledge of plate structure information.

Plate Structure Identification

In order to successfully solve the upcoming issues, first a simple plate structure identification process is done based on the prior knowledge we have about Ethiopian plate's layout. In general, we roughly classify Ethiopian plates, as single or double row and/or circle-coded or circle-un-coded based on the layout of characters and shapes included (see Figure 3.22).

Figure 3.22: On *(a)* circle-coded plates with single and double row plate structure, *(b)* circle-un-coded plates with single and double row plate styles

Structure/type of input plate image is identified using two consecutive techniques. First, to identify whether a plate is single or double row, detected number objects location respect to the origin of the plate is used. And then, to identify whether a plate is circle-coded or circle-un-coded, circle object is searched within the detected objects, found on the top-left corner of the image for single row plates, and in the top part of the image for double row plates.

Connected Characters and Numbers Separation

While using CCL, if objects are connected to each other with group of pixels or a pixel that satisfies the pixels connectivity definition (i.e., 4-neighborhood connectivity), then the objects will be considered as one single object oppose to the truth (see Figure 3.23).

In plate's characters case, the usual reasons for characters as well as number objects connections are rivets, poor image quality, and illumination effects that influence the selection of threshold value in the binarization process. The usual types of connections are number-to-number, character-to-number, circle-to-character, and character-to-character (usually top-bottom connection) (see Figure 3.23).

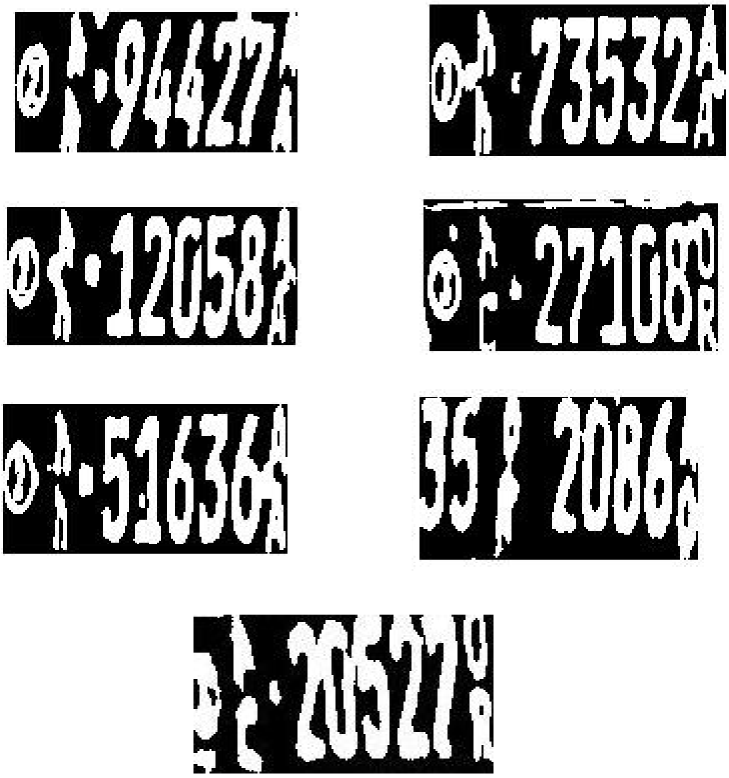


Figure 3.23: Instance of plates with connected characters in different positions

The first step towards solving this problem is, identifying characters/numbers that are detected as one instead of separated objects. This issue is resolved by isolating objects that have a width, way beyond the average width of objects in that particular plate. After identification, the connected object is segmented into two separate parts by identifying whether they are connected side to side or top to bottom using the plate structure information provided (see Figure 3.24).

Figure 3.24: Output of the CS module (b) that uses combination of CCA and connected characters separation module applied on the corresponding images of (a).

3.2.3 Character Recognition

3.2.3.1 Introduction

Character recognition is the process of recognizing characters from a given image, using image processing techniques. As studies show, neural network recognition approaches have higher classification accuracy than other character recognition methods [19]. However, neural network approaches need large amount of data for training, which makes them un-preferable for this system, because of the problem that exists in the data collection process.

So, for the character recognition process, we used correlation based template matching method, which is supported by additional features and techniques (will be explained later) that makes the recognition process perform better and the classification decision not entirely lie on the correlation value. The architecture of the proposed character recognition system is show in Figure 3.26.

Character and number recognition

In this system, recognition of numbers and characters is treated separately, in order to prevent the recognition process from making impractical mistakes like, recognizing number as a character and vice versa. Accordingly, segmented character images obtained from the CS module are separated as character and number image sets. So does the number and character template images.

Templates

Template image sets that constitute plate character and number appearances in real environment are collected from real world data. Some of the issues considered while selecting templates are character‟s orientation and characters with deformed shape, especially caused by rivets. The template images are binarized and normalized to 24x24 and 24x34 pixel sizes for character and number templates respectively. Some template images used in the recognition process are shown in Figure 3.25**.**

Figure 3.25: Some character template images used

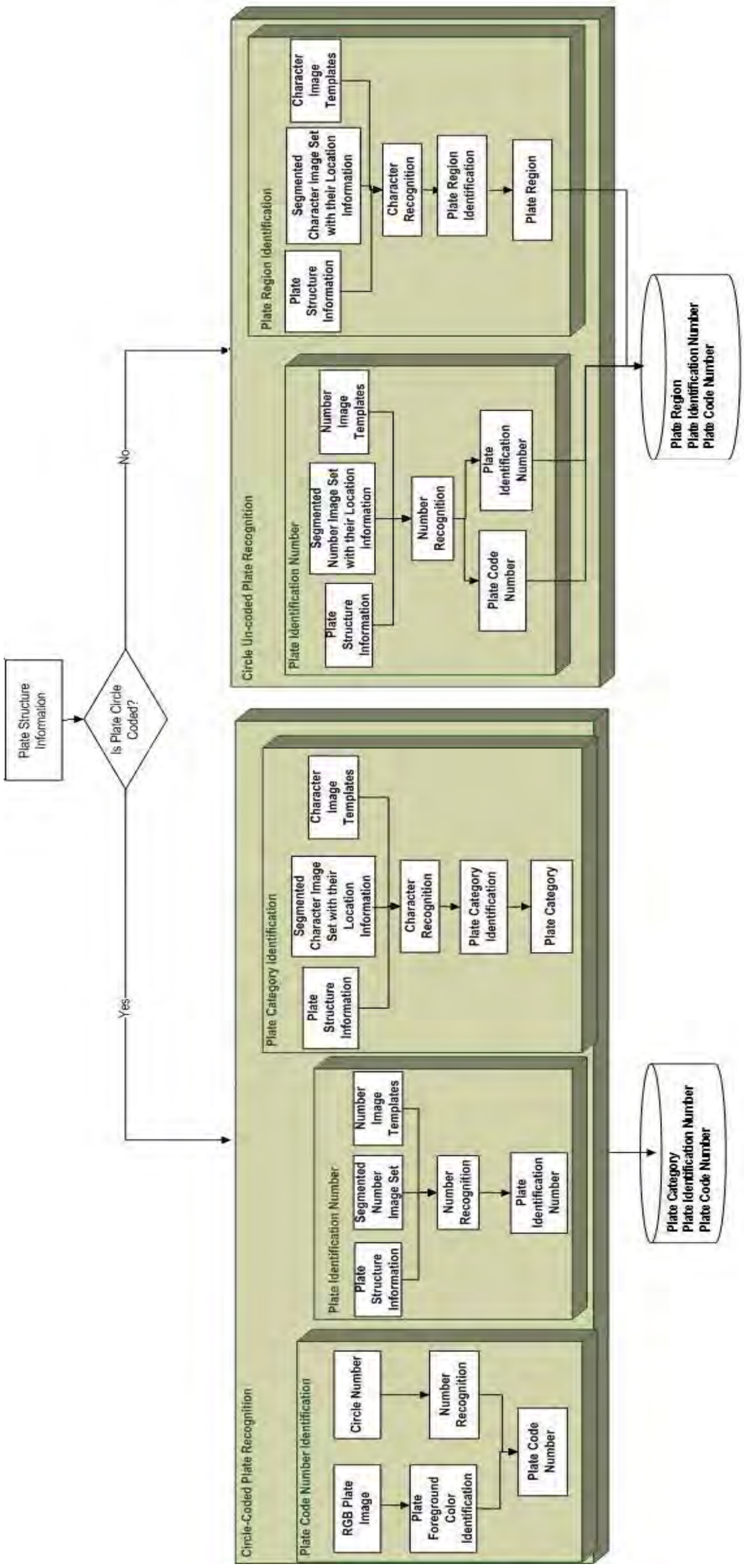


Figure 3.26: Architecture of the character recognition module

To minimize computational cost of the CR module, the recognition process uses plate structure information provided by the CS module. This information prevents unlikely recognition decisions. For instance, characters found on circle-un-coded plates that are authorized by the federal authority do not appear on any of the circle-coded plates. So, by taking this advantage, which not only minimizes computation cost, but also increases accuracy of the recognition process, the template images are classified and stored as numbers, circle coded-Amharic characters, circle coded-English characters, circle-un-coded Amharic characters and circle-un-coded English characters. Thus, at a single execution of the system, only the templates that are related to the input plate structure are loaded and presented to the recognition process.

Number recognition

Segmented number images obtained from the CS module are binary images that have different sizes. So, the first step performed here is, normalizing their size to the size of the number template images (i.e., 24x34). After the normalization process, a fitting process is applied on all of the segmented number images to remove unwanted space. Subsequent to the fitting process, a correlation value is computed between each number source image and the number template images. The template image that scores the higher correlation value (i.e., close to 1.0) is chosen as the number representative for the source image.

Character recognition

As discussed earlier in the beginning of the chapter, Ethiopian plates include both English and Amharic characters. Hence, this situation gives us the chance to identify plate‟s region either from the Amharic or the English characters. However, on some plates like 'ተላላፊ' and 'ልዩ' only the Amharic characters appear. Moreover, because of the reasons explained earlier like image quality, dirt and etc… one of the characters either the Amharic or the English ones or both might be badly defected. Thus, this system takes advantage of the existence of information both in Amharic and English characters in order to get a better character recognition accuracy. Accordingly, first, a correlation value is computed for all detected characters. Then a sub-module cross relates the Amharic and English characters using character‟s location information. Finally, the one with the higher correlation value from the corresponding Amharic and English characters is taken for the identification of plate‟s region/category. This action has two main advantages: first, if both the Amharic and English characters are presented, it gives a higher accuracy of region identification. Second, it gives us the opportunity to identify plate"s region/category even if, only the Amharic or English characters are presented and also only the two opposite (based on their location) characters are presented (i.e., one Amharic and one English).

Circle-code number identification

Under the real world circumstances mentioned earlier that influence input images quality, it is difficult to detect, extract out and recognize the code number found in a circle (i.e., in circle-coded plate types) by using the same methods we used for the other characters. Most of the time, the code numbers are highly connected to the circle, which makes it difficult to separate the numbers from the circle. And some other times, even if we are able to extract the code number from the circle, it is highly distorted. As a result, it is inconvenient to totally rely on a correlation value for recognition.

So, the only way to solve this problem is, by taking advantage of the uniqueness of the plate"s foreground color in each of the circle-coded plates (see Table 3.1). Thus, identifying foreground color of circle-coded plates is equivalent to recognizing/identifying the plate"s code number.

The foreground color identification is done by performing color analysis on the original RGB plate image according to the location information of characters and numbers obtained from the CS module. Table 3.3 shows a partial MATLAB™ code used to identify foreground color of a plate.

Table 3.3: A partial MATLAB™ code used to decide pixel's color

```
For i=1:size(character,2)
 For each (x,y) £ character(i).Image
      r=Iplate(x,y,1);  g=Iplate(x,y,2);  b=Iplate(x,y,3);
                      if g>100 && b>100
                             if b>g && g>r
                              color='blue';
                             end
                      end
                      if r>b && b>g
                          color='red';
                      end
                      if g>90 && b>90 && r>90
                          if g>b | b>r
                              color='green';
                          end
                      end
```

Finally, the code number of circle-coded plates is decided by choosing over the reliable outcome out of the foreground color identification result and the correlation value of the recognized number.

CHAPTER FOUR

EVALUATION

4.1 Evaluation

The developed system is tested on 350 real time pictures that contain car images, where the plate regions are viewed from different angles and distance (see Table 4.1). In order to easily assess where the system strengths are and find out its vulnerabilities, the testing process is done both individually (i.e., on PD, CS and CR) and as a whole. The system is tested on Intel Core i3 2.4GHz processing unit.

In this testing process, success rate of each result is computed using the following formula:

$$\boldsymbol{SuccessRate} = \frac{\boldsymbol{Total\ instance\ of\ Success}}{\boldsymbol{Total\ Number\ of\ Input\ Sample}}\ \boldsymbol{x\ 100}\ \% \qquad \textbf{(4.1)}$$

Table 4.1: Image Database Composition

Samples	Description	Camera Angle	Camera Distance	Degree of Background Complexity	Images Size
Set 1	Day light, Moving Cars	Up to-30^o	High	Average-High	153
Set 2	Day light, Moving and Non Moving Cars	Up to-15^o	Average	Average	124
Set 3	Day light , Moving and Non Moving Cars	Up to-15^o	Low	Low	35
Set 4	Night, Moving Cars	Up to-15^o	Average	Average	16
Set 5	Shadow Packing , Moving Cars	Up to-15^o	Low - average	Low- Average	14
Set 6	Day light and Night, Moving Cars , Blurred	Up to-15^o	Low – average	Low	8

Camera Distance

High → 3-5 meters

Average → 1-3meters

Low → < 1 meter

The following tables (Table 4.2-4.5) present the testing result of each individual module and the system performance as a whole:

Table 4.2: The testing result of the plate detection module

Sample Set	**Input Size**	**Successfully extracted Plates**	**Success rate**
Set 1	153	146	95.4 %
Set 2	124	115	92.7 %
Set 3	35	31	88.5%
Set 4	16	9	56.3 %
Set 5	14	6	42.8%
Set 6	8	4	50%
Total	350	311	88.9%

Table 4.3: The testing result of the character segmentation module

Sample Set	**Input Size (Successfully extracted Plates)**	**Number of plates, where CS module succeed**	**Success rate**
Set 1	146	124	84.9%
Set 2	115	102	88.7%
Set 3	31	29	93.5%
Set 4	9	3	33.3%
Set 5	6	3	50 %
Set 6	4	0	0%
Total	311	261	83.9%

Table 4.4: The testing result of the character recognition module

Sample Set	**Input Size (Number of plates, where CS module succeed)**	**Number of identified/recognized plates**	**Success rate**
Set 1	124	102	82.25 %
Set 2	102	89	87.25 %
Set 3	29	25	86.2%
Set 4	3	2	66.6 %
Set 5	3	3	100%
Set 6	0	-	-
Total	261	221	84.67%

Table 4.5: The overall system success rate

Number of input images	Number of plates extracted successfully	Number of plates segmented successfully	Number of plates recognized successfully	Success rate
350	311	261	221	63.14 %

Since we were unable to get the program code of [4] (i.e. the existing Ethiopian car plate recognition system) to evaluate our system performance and being able to compare to it, we managed to implement its PD module only, and tested it on the same data we just used. The testing result is shown at Table 4.6.

Table 4.6: The comparison testing result of our PD module with PD of [4]

Number of input images	Successfully extracted plates by our system	Successfully extracted plates by PD of [4]
350	311 (88.9)%	96 (27.42%)

As of the CS and CR modules, [4], uses connected component analysis and template matching techniques respectively. So, since we do not even know the character features used and also the template images, we were unable to implement these modules. However, still if were able to implement the CS module in [4], which considers all plates as if they have 10 alpha numeric characters, it will not perform well on the data at hand, because of the fact that the data set contains different structure of Ethiopian car plates, opposed to the data set used in [4] that only contain 10 characters per plate.

As we can see from the testing result, the system in [4] is able to detect only 96 plate regions out of 350, which is a very low accuracy rate that dramatically degrades the whole recognition process, since execution of both CS and CR modules depends on the successful completion of the PD module. So, from this we can conclude that our system got a better recognition accuracy rate.

4.2 Component Evaluation

In this section the testing result of the system and points where the system fails are discussed widely.

4.2.1 Plate Detection

The developed PD module is tested on 350 car images that constitute real environment car pictures that are taken under different circumstances (Table 4.2) and it got 88.9 (%) success rate by detecting 311 plate regions out of 350 (see Figure 4.1). The module performed a good job in detecting plate regions from car images taken under a camera distance of 1-5 meters and camera angle of up to 30° (i.e., set 1, set 2 and set 3). On the other hand, from the testing result, we noted that the module poorly performs in low contrast input images like, images taken at night time (i.e., set 4 and set 6). In addition, in set 6 blurred input images also highly influenced the accuracy of the PD module. Moreover, the system also poorly performs on sample set 5, because of the shadow effects, which indicates that there is a technique used in the PD process that is easily affected by shadow (discussed later). In this section, some of the reason and points where, this module fails to detect plate regions are discussed.

Figure 4.1: Sample successful plate detection instances

Effect of shadow on the PD module

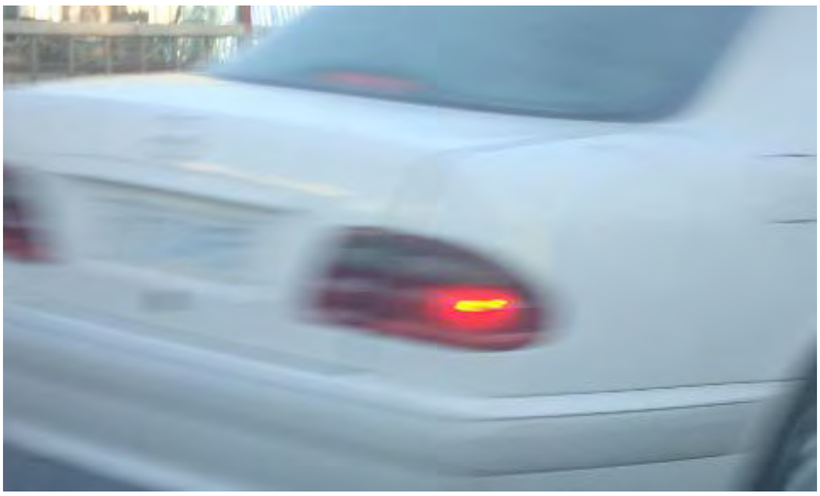
Because of illumination variations that exist while taking pictures of cars, shadows could be created. Appearance of shadow, especially in plate region area of input images has severe impact on the outcome of the plate detection module. This happens because of the reason that appearance of shadow directly affects the selection of a threshold value in the binarization process of the PD module. This action is capable of making parts of plate region to be treated as different class (background) and/or make a plate region to be connected with other regions that might make the plate region to be rejected by the plate candidates filtering sub module since it does not satisfy the aspect ratio feature of plate regions. Figure 4.2 shows instance of an input image where its plate region area packed a shadow. As we can observe from the output of the binarization process of the PD module (Figure 4.2 (b)) the plate region is connected to other unwanted regions because of the appearance of a shadow.

Figure 4.2: An instance where the PD module fails to detect plate region, because of shadow effects

Blurred input images

Blurry images usually occur if the input picture is taken, while the object is in a very fast motion. A deblurring operation was considered to solve this case, but it was not sufficient enough for the plate detection operation. Even if, the input image makes it through the plate detection stage, it could not pass the character segmentation step. Due to the reason that it is difficult to segment the plate characters from the background image, in the binarization process of CS module. Figure 4.3 shows instances of blurred input images where, on (a) the PD module was able to detect plate region while on (b) it fails to do so.

(a)



(b)

Figure 4.3: The output of the PD module on blurred input images

4.2.2 Character Segmentation

The developed CS module is tested on 311 plate regions that are successfully detected by the PD module from the total input images of the system (i.e., 350). The module was able to successfully segment plate characters of 261 plate regions and also identify their plate structure, which is the key input for the CR module. As we observed from the testing result, this module in particular got a higher character segmentation performance on input images that are taken under day light, a camera distance of up to one meter and a camera angle of up to 15^o (i.e., set 3). In addition, the module also got higher segmentation rate in image set 1 and 2. However, the module badly performs in image sets that contain pictures taken at night time (i.e., 4 and 6). Nevertheless, the CS module got a total of 83.9% success rate, which is achieved through the contribution of the plate‟s frame elimination, plate‟s orientation adjustment, and post segmentation sub modules. Although, failure in one of these sub modules can lead to unsuccessful outcome of the CS module.

4.2.3 Character Recognition

This module results in 84.67% of success rate. From the testing process, we noted that this module can identify plate‟s region/category, even though all of the region/category code characters are not presented. In addition, the usage of characters location information on the plate recognition process has made a great difference in the performance as well as the accuracy of the recognition process. Overall, all the techniques used in the CR module such as fore-ground color identification, Amharic and English characters location and correlation result cross checking, usage of plate structure information that helps to decide which template images to load (helpful in cutting the percentage of misclassification by half) greatly influence the performance as well as the accuracy of the CR module.

On the other hand, from the testing process, we observed that this module misclassifies numbers like 0 for 8, characters ኪ for ከ, and ከ for ኬ. As well characters and numbers affected by plate‟s rivet have got higher percentage of misclassification.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this study, a system that recognizes Ethiopian car plates is developed. The system has three major modules: plate detection, character segmentation and character recognition. Each of the modules uses various image processing techniques to perform the required task.

The system starts by accepting RGB image of a car. Then the PD module takes up the input and applies a series of image processing techniques, in order to locate a plate region. Before starting the process of detection, first the input image passes through some preprocessing tasks, namely, gray scale conversion and size normalization. Then the subsequent outcome passes through Gabor filter, binarization, morphological closing operation, and CCA processes, to detect potential plate candidates. Finally, candidate plates filtering is performed, to identify the genuine plate region.

Once the plate extraction is done, the character segmentation module continues to identify and segment out each alpha-numeric characters found in the extracted plate image. This module has three consecutive stages: preprocessing, segmentation and post segmentation. In preprocessing stage, the input plate image is normalized into manageable size and also Hough transform based orientation adjustment process is performed. In the segmentation process, plate‟s frame elimination process is performed on the extracted binary plate image prior to the CCA that is used to detect and extract each characters of the plate. In the last stage of character segmentation module (i.e., post segmentation) identification and separation of connected characters with a prior knowledge of input plate structure is performed on the output of CCA.

In addition to character segmentation, this module provides functionalities like, identifying structure/type of input plates, and providing location information of detected characters to the CR module.

The last module of the system is character recognition. This module accepts character and number image sets along with location and plate structure information from the character segmentation module. In addition to these inputs, it also takes template character and number image sets according to the plate structure information provided. The developed character recognition module uses a correlation based template matching method, which uses a correlation value computed between the source and template image set for recognition decision. However, in this system, the recognition decision is supported by location information of characters, plate structure information of input plates and color processing techniques in addition to correlation values.

At the end, the proposed system is developed using MATLAB™, then it is tested on 350 car pictures that are taken under various illumination conditions, car motion, camera distance and angle. The testing process is done both in groups and individually on each of the major modules (i.e., PD, CS and CR) of the system. The PD, CS and CR modules got a success rate of 88.9%, 83.9% and 84.67% respectively. All these modules perform good in input images that are taken under daylight, 1-5 meters of camera distance and camera angle of up to 30^o. On the other hand, the modules are found to be sensitive to blurred input images, to images that have low contrast, and to input images that contained shadows, especially in the plate region area.

Overall, the developed system got an accuracy rate of 63.14% by correctly detecting and recognizing 221 plate images out of 350 input car images. The system is able to detect and recognize a plate between 2-5 seconds depending on whether post-processing operations are needed or not.

In conclusion, in this work, a CPR system that recognizes more varieties of Ethiopian car plates than the existing CPR system for Ethiopian car plates [4] is developed with better performance and accuracy rate.

5.2 Recommendations

In order to enhance the performance and also accuracy of the system the following points could help in the future.

- Adding contrast enhancement technique, as a preprocess in the plate detection module
- Using local binarization techniques than global threshold values
- Applying vertical angle adjustment on the detected plate region before the character segmentation process

REFERENCES

[1] Bremananth, R.Chitra, A. Seetharaman, V. Nathan, V.S.L., A robust video based license plate recognition system, IEEE Proceedings of ICISIP, 2005, Page(s): 175 – 180.

[2] Zhenxue Chen, Guoyou Wang, Jianguo Liu and Chenyun Liu, Automatic License Plate Location and Recognition Based on Feature Salience, International Journal of Computational Cognition, Vol. 5, No. 2, June 2007.

[3] I-Chen Tsai, Jui-Chen Wu, Jun-Wei Hsieh, and Yung-Sheng Chen, Recognition of Vehicle License Plates from a Video Sequence, IAENG International Journal of Computer Science, 36:1, IJCS_36_1_04.

[4] Huda Zuber Ahmed, Design and Implementation of Car Plate Recognition System for Ethiopian Car Plates, http://www.aau.edu.et/dspace/

[5] Ali Tahir, Hafiz Adnan Habib, M. Fahad Khan, License Plate Recognition Algorithm for Pakistani License Plates, Canadian Journal on Image Processing and Computer Vision Vol. 1, No. 2, April 2010.

[6] Mahmood Ashoori-Lalimi, Sedigheh Ghofrani, An Efficient Method for Vehicle License Plate Detection in Complex Scenes, Scientific Research Circuits and Systems, 2011, 2, 320-325.

[7] H.S. Kim, et al., 1991, Recognition of a Car Number Plate by a Neural Network, in the Proceedings of the Korea Information Science Society Fall Conference, Vol. 18, pp. 259-262, 1991.

[8] Lin Liu, Sanyuan Zhang, Yin Zhang, and Xiuzi Ye, Slant Adjustment of Vehicle License Plate Image, F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 237 – 244, 2005.

[9] Wen C-Y, Yu C-C, Hun Z-D, A 3-D transformation to improve the legibility of license plate numbers, Journal of Foresensic Sciences, 2002, 47(3): 578-585.

[10] Hadi Sharifi, Asadollah Shahbahrami, An Evaluation of License Plate Recognition Algorithms, International Journal of Digital Information and Wireless Communications (IJDIWC) 1(1): 281-287 The Society of Digital Information and Wireless Communications, 2011.

[11] Phalgun Pandya, Mandeep Singh, Morphology Based Approach To Recognize Number Plates in India, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-3, July 2011.

[12] S. Hamidreza Kasaei, S. Mohammadreza Kasaei, S. Alireza Kasaei, New Morphology-Based Method for Robust Iranian Car Plate Detection and Recognition, International Journal of Computer Theory and Engineering, Vol. 2, No. 2 April, 2010 1793-8201.

[13] T. Duan, D. Duc, T. Hong Du, Combining Hough Transform and Contour Algorithm for detecting Vehicles‟ License-Plates, Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing , October 20-22, 2004 Hong Kong.

[14] 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.

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

[16] S. Zheng and H. Lee, Detection and recognition of license plate characters with different appearances, Proc. of 16th International Conference on Pattern Recognition, vol.3, 2003, pp. 979-983.

[17] Hakan Caner, H. Selcuk Gecim, and Ali Ziya Alkar, Efficient Embedded Neural-Network-Based License Plate Recognition System, IEEE Transactions on Vehicular Technology, Vol. 57, No. 5, September 2008.

[18] Ying Wen, Yue Lu, Jingqi Yan, Zhenyu Zhou, Karen M. von Deneen, and Pengfei Shi, Senior, An Algorithm for License Plate Recognition Applied to Intelligent Transportation System, IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 3, September 2011.

[19] V. Koval 1, V. Turchenko 1, V. Kochan 1, A. Sachenko 1, G. Markowsky, Smart License Plate Recognition System Based on Image Processing Using Neural Network , IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 8-10 September 2003, Lviv, Ukraine.

[20] Guangmin Sun, Gang Li, Lei Xu and Jing Wang, The Location and Recognition of Chinese Vehicle License Plates under Complex Backgrounds, Journal of Multimedia, Vol. 4, No. 6, December 2009.

[21] Vibha S. Vyas and Priti Rege, Automated Texture Analysis with Gabor filter, GVIP Journal, Volume 6, Issue 1, July 2006.

[22] Jaedo Kim, Youngjoon Han, and Hernsoo Hahn , Character Segmentation Method for a License Plate with Topological Transform, World Academy of Science, Engineering and Technology 56 2009.

[23] Yaregal Assabie and Josef Bigun, Ethiopic Character Recognition Using Direction Field Tensor, The 18th International Conference on Pattern Recognition (ICPR'06) 0-7695-2521-0/06 2006 IEEE.

[24] Million Meshesha and C. V. Jawahar, Optical Character Recognition of Amharic Documents, Center for Visual Information Technology International Institute of Information Technology

[25] Md. Mahbubul Alam Joarder, Khaled Mahmud, Tasnuva Ahmed, Mohsina Kawser, and Bulbul Ahamed, Bangla Automatic Number Plate Recognition System using Artificial Neural Network, Asian Transactions on Science & Technology (ATST ISSN: 2221-4283) Volume 02 Issue 01.

[26] Eskinder Desta, The Automotive Industry and Trend Analysis in Ethiopia, July 2007.

[27] Kripa Iyer, Smitha S. , An Automatic & Robust Vehicle License Plate Recognition System, 978-1-4244-1674-5/08, 2008 IEEE.

[28] Vidushi Sharma Sachin Rai Anurag Dev, A Comprehensive Study of Artificial Neural Networks, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 10, October 2012.

[29] Ehsan Nadernejad, Sara Sharifzadeh, Edge Detection Techniques:Evaluations and Comparisons, Applied Mathematical Sciences, Vol. 2, 2008, no. 31, 1507 – 1520.

[30] Pavlos Stathis, Ergina Kavallieratou, Nikos Papamarkos, An Evaluation Technique for Binarization Algorithms, Journal of Universal Computer Science, vol. 14, no. 18 (2008), 3011-3030.

[31] Varun Manchikalapudi, Skew Correction and Localisation of Number Plate Using Hough Rectangular Transform, International Journal of Computer Science and Telecommunications Vol. 2, ISSue 3, September 2011.

[32] P. Kruizinga, N. Petkov and S.E. Grigorescu, Comparison of texture features based on Gabor filters, Proceedings of the 10th International Conference on Image Analysis and Processing, Venice, Italy, September 27-29, 1999, pp.142-147.

[33] Amalorpavam.G, Harish Naik T, Jyoti Kumari, Suresha M, Analysis Of Digital Images Using Morphlogical Operations, International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 1, February 2013

[34] Richard O. Duda, Peter E. Hart, Use of The Hough Trasformtion to Detect Lines And Curves in Pictures, Comm. ACM, Vol 15, No.1, pp. 11-15 , January 1972.

[35] Love Jain, Devesh Mahor, Application of Hough Transform for finding Parametric Curves, International Journal of Computer Applications in Engineering Sciences, Vol I, Issue II, June 2011.

[36] J. G. Daugman, Uncertainty Relation for Resolution in Space, Spatial Frequency, And Orientation Optimized By Two-Dimensional Visual Cortical Filters, Journal of the Optical Society of America, vol. 2, pp. 1160-1169, July 1985.

[37] S.Kranthi, K.Pranathi, A.Srisaila, Automatic Number Plate Recognition, International Journal of Advancements in Technology, Vol 2, No 3 ,July 2011.

[38] Serkan Ozbay, and Ergun Ercelebi, Automatic Vehicle Identification by Plate Recognition, World Academy of Science, Engineering and Technology , 2005.

[39] D. G. Bailey, C. T. Johnston, „Single Pass Connected Components Analysis‟, Proceedings of Image and Vision Computing New Zealand 2007, pp. 282–287, December 2007.

[40] Nadir Nourain Dawoud, Brahim Belhaouari Samir, Josefina Janier, Fast Template Matching Method Based on Optimized Metrics for Face Localization, Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol I, March 14 - 16, 2012.

[41] Dimitrios Chrysostomou, Antonios Gasteratos, Comparison And Evaluation Of Similarity Measures For Vergence Angle Estimation Proceedings of 16th Int. Workshop on Robotics, June 7-9, 2007

APPENDICES

APPENDIX A: GUI of the system

The graphical user interface of the developed CPR system is shown in Figure A.1. The system starts, when an input image is provided using the "Load Image" button. The input image has to be either RGB or intensity image. Then, the next step is clicking on the "Extract Plate" button, which detects and extracts the plate region from the loaded image. Finally, the "Recognition" button is used to recognize the individual characters on the extracted plate and displays and stores the code number, plate number and region/category of the plate (see Figure B.1).

Figure A.1: GUI of the developed CPR system

APPENDIX B: Sample Snapshots

Figure B.1: Sample snapshots of, the GUI of the system, filled with input image and the system response

APPENDIX C: Sample MATLAB^{TM} Codes

In this section sample/partial MATLAB^{TM} codes of the system from each module are presented.

1. The plate detection function sample code

```
function[firstCandidates]= plateDetection(inputImage, conn)

%****** SIZE NORMALIZATION AND RGB To GRAY SCALE CONVERSION ********

Ir=imresize(inputImage,[480 640]);
    if size(Ir,3)==3
        Ig=rgb2gray(Ir);
    else
        Ig=Ir;
    End

%********************** GABOR FILTER********************************

[G,gabout] = gaborfilter(Ic,2,4,6,pi/3);
gab=uint8(gabout);

%******* BINERIZATION AND MORPHOLOGICAL CLOSING OPERATION **********

Ib=im2bw(gab,graythresh(gab));
Ib=bwareaopen(Ib,20);
SE=strel('rectangle',[1 conn]);
Ib=imclose(Ib,SE);
Ib=bwareaopen(Ib,300);

%************************* CONNECTED COMPONENT LABELING ************

firstCandidates=struct('image',{});
regions=PlateCCL(Ib);
if size(regions,1)==0 || size(regions,2)==0
    return;
else
    for i=1:size(regions)
      x=regions(i).BoundingBox(1);
      y=regions(i).BoundingBox(2);   % POSITION OF CANDIDATE
      w=regions(i).BoundingBox(3);   % PLATE REGIONS
      h=regions(i).BoundingBox(4);

        Icr=imcrop(Ir, [x y w h]); %CROPPING CANDIDATE PLATE REGIONS
        firstCandidates(i).image=Icr;
   end
end
```

2. Character segmentation

```
%*********************************************************************
%PRE-PROCESSING OPERATIONS FOR CHARACTER SEGMENTATION MODULE
%********************************************************************

function [Ir,Ig]=csPreprocessing(RGBplateimage)

Iorg=RGBplateimage;

%% ORIENTATION CORRECTION

Ig=rgb2gray(Iorg);
Ib=edge(Ig,'canny');
[H,theta,rho]=hough(Ib);     %Generate the hough matrix

p=houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))))  %Find 5 peak values in H

lines=houghlines(Ib,theta,rho,p);    %Search lines

%Spatial Transformation
slopes = vertcat(lines.point2) - vertcat(lines.point1);
slopes = slopes(:,2) ./ slopes(:,1)

if slopes(1)~=Inf
    TFORM = maketform('affine', [1 -slopes(1) 0 ; 0 1 0 ; 0 0 1]);
    I=imtransform(Iorg,TFORM);
    Ib=imtransform(Ib,TFORM);
else
    I=Iorg;
end

%Cropping the corrected plate region from Iorg
SE=strel('rectangle',[1 20]);
Ib=imclose(Ib,SE);
plateRegion=PlateCCL(Ib);  %Extracting the actual location of the plate

if size(plateRegion,2)==0
    plate=Iorg;
else
    for i=1:size(plateRegion)
        x=plateRegion(i).BoundingBox(1);
        y=plateRegion(i).BoundingBox(2);
        w=plateRegion(i).BoundingBox(3);
        h=plateRegion(i).BoundingBox(4);
        plate=imcrop(I, [x y w h]);
    end
end

%% SIZE NORMALIZATION and GRAY SCALE CONVERSION

Ir=imresize(plate,[100 160],'bilinear');
Ig=rgb2gray(Ir);
```

3. Characters Recognition

```
function []=CRHome(pType,num,char,RgbObjects,circle,Ir,t,l)

[num,amhChars,engChars,circleData,pType]=CRpre(num,char,pType,circle);
    %% NUMBER RECOGNITION
            [num_temp]=loadNumTemplates();
            [Numbers]=CorrRecogntion(num,num_temp,[34 24]);

                for j=1:size(Numbers,2)
                disp(Numbers(j).id);
                end


    %% CHARACTERS RECOGNITION
           if strcmp(pType,'circled')
                   [amh_temp]=loadCodedAmhCharTemplates();
                   [eng_temp]=loadCodedEngCharTemplates();

               [AmharicChars]=CorrRecogntion(amhChars,amh_temp,[24 24]);

               [EnglishChars]=CorrRecogntion(engChars,eng_temp,[24 24]);

            elseif strcmp(pType,'un-circled')
                    [amh_temp]=loadunCodedAmhCharTemplates();
                    [eng_temp]=loadunCodedEngCharTemplates();

               [AmharicChars]=CorrRecogntion(amhChars,amh_temp,[24 24]);

               [EnglishChars]=CorrRecogntion(engChars,eng_temp,[24 24]);

            end


    %% REGION IDENTIFICATION
          if strcmp(pType,'circled')
            [region]=regionIdentification(AmharicChars,EnglishChars);
            disp(region);

          elseif strcmp(pType,'un-circled')

      [region]=uncodedPlateCodeIdentification(AmharicChars,EnglishChars);


           end
```

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work performed under the supervision of Dr. Yaregal Assabie, has not been presented as a thesis for a degree program in any other university and all sources of materials used for the thesis are duly acknowledged.

Name: Seble Nigussie Ayalew Signature:_____________________

Place: Addis Ababa

Date of submission: April, 2013

This thesis has been submitted for examination with my approval as a university advisor.

Advisor's Name: Dr. Yaregal Assabie Signature: _____________________