12-2015

Signage Recognition Based Wayfinding System for the Visually Impaired

Abdullah Khalid Ahmed
Western Michigan University

SIGNAGE RECOGNITION BASED WAYFINDING SYSTEM FOR THE VISUALLY IMPAIRED

by

Abdullah Khalid Ahmed

A thesis submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Master of Science in Engineering (Electrical)
Electrical and Computer Engineering
Western Michigan University
December 2015

Thesis Committee:

Ikhlas Abdel-Qader, Ph.D., Chair
Raghvendra Gejji, Ph.D.
Azim Houshyar, Ph.D.
Osama Abudayyeh, Ph.D.

SIGNAGE RECOGNITION BASED WAYFINDING SYSTEM FOR THE VISUALLY IMPAIRED

Abdullah Khalid Ahmed, M.S.E.

Western Michigan University, 2015

Persons of visual impairment make up a growing segment of modern society. To cater to the special needs of these individuals, society ought to consider the design of special constructs to enable them to fulfill their daily necessities. This research proposes a new method for text extraction from indoor signage that will help persons of visual impairment maneuver in unfamiliar indoor environments, thus enhancing their independence and quality of life.

In this thesis, images are acquired through a video camera mounted on glasses of the walking person. Frames are then extracted and used in an integrated framework that applies Maximally Stable Extremal Regions (MSER) to detect alphabets along with a morphological dilation operation to identify clusters of alphabets (words). Proposed method has the ability to localize and detect the orientation of these clusters. A rotation transformation is performed when needed to realign the text into a horizontal orientation and allow the objects to be in an acceptable input to any of the available optical character recognition (OCR) systems. Analytical and simulation results verify the validity of the proposed system.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor, Dr. Ikhlas Abdel-Qader, for her guidance, suggestions and support throughout the development of this thesis. She introduced me to the world of research and encouraged me to develop my own ideas for the problem while supporting me at each step with her knowledge, attentiveness and advice. Working with her has been a valuable experience for me and my continued education.

I would like to extend my thanks and appreciation to each member of my thesis committee, Dr. Raghvendra Gejji, Dr. Azim Houshyar, and Dr. Osama Abudayyeh, for reviewing this thesis and valuable suggestions to its development with their insights and feedback.

I would like to express my deepest gratitude to my family and friends, who are always standing by me, and for their endless support and understanding.

Abdullah Khalid Ahmed

TABLE OF CONTENTS

Table of Contents - continued

CHAPTER

Table of Contents - continued

LIST OF TABLES

LIST OF FIGURES

List of Figures- Continued

List of Figures- Continued

List of Figures- Continued

CHAPTER 1

INTRODUCTION

In 2010, researchers concluded that of the 285 million visually impaired people worldwide, 39 million are believed to be blind. People who suffer from impaired vision encounter difficulties and limitations in society, negatively impacting their quality of life, and seriously affecting their ability to conduct a productive life [1]. Finding employment to support themselves and their families, therefore, becomes an obvious challenge, even in industrialized and developed countries. For example, the Centers for Disease Control and Prevention (CDC) in the United States reported that there are 12 million visually impaired Americans, more than one million of whom are legally blind [2].

Text is considered one of the most meaningful avenues through which people communicate and disclose information. Therefore, the ability to extract text from complex background images becomes essential for visually impaired and blind people to be able to access this information. Recent availability of cameras in many portable devices, such as mobiles phones and tablets, has allowed researches to focus on developing methods of extraction to provide solutions to this problem [3]. The high performance of these cameras coupled with the ergonomic interface of these devices has allowed development of computer vision assistive technologies that grant blind persons the ability to access text in signage. Extracting this text information from signage, however, has many technical challenges such as variation in the alignment, color, size,

and edge quality of the text paired with the additional challenges of varying camera angles.

Any text extraction algorithm must overcome these challenges as accuracy is of paramount necessity in devices aiming to aid the blind. In this proposed work for an indoor signage reader system challenges will be identified and addressed. For example, the color of text characters is always the same within each sign throughout a building, making it easier to be recognized against sign background. Furthermore, the text edges are typically strong in images of signage because the text is carefully designed to be easily read. Nevertheless, distortion artifacts continue to affect the performance of current extraction technologies, and the search for better algorithm thus continues [3].

A system for text detection and recognition in general may consist of three major stages, as it is illustrated in Figure 1.1:

1) Text detection and localization

2) Text extraction and enhancement

3) Text recognition such as in Optical Character Recognition (OCR)

In the first stage, text regions are detected from still images, grouped into text strings/objects, and finally bounded into boxes. The second stage extracts text by applying cropping processes to the bounding box and enhancing the extracted text. This enhancement, however, increases the likelihood of generating noise, which may negatively influence the efficiency of subsequent stages output. Finally, the enhanced

extracted text is ready to be recognized by an OCR system to get the machine-encoded text as audio signal for the blind person ear piece.

The purpose of this research is to investigate the recent advances in text detection and to evaluate the performance of algorithms in applications of text extraction. In the process of this evaluation, special attention will be placed on indoor use of signage readers to assist the blind and visually impaired persons with guidance. Focus will be placed on the first two stages of text extraction process, since modern OCR systems are highly accurate in recognizing extracted texts.

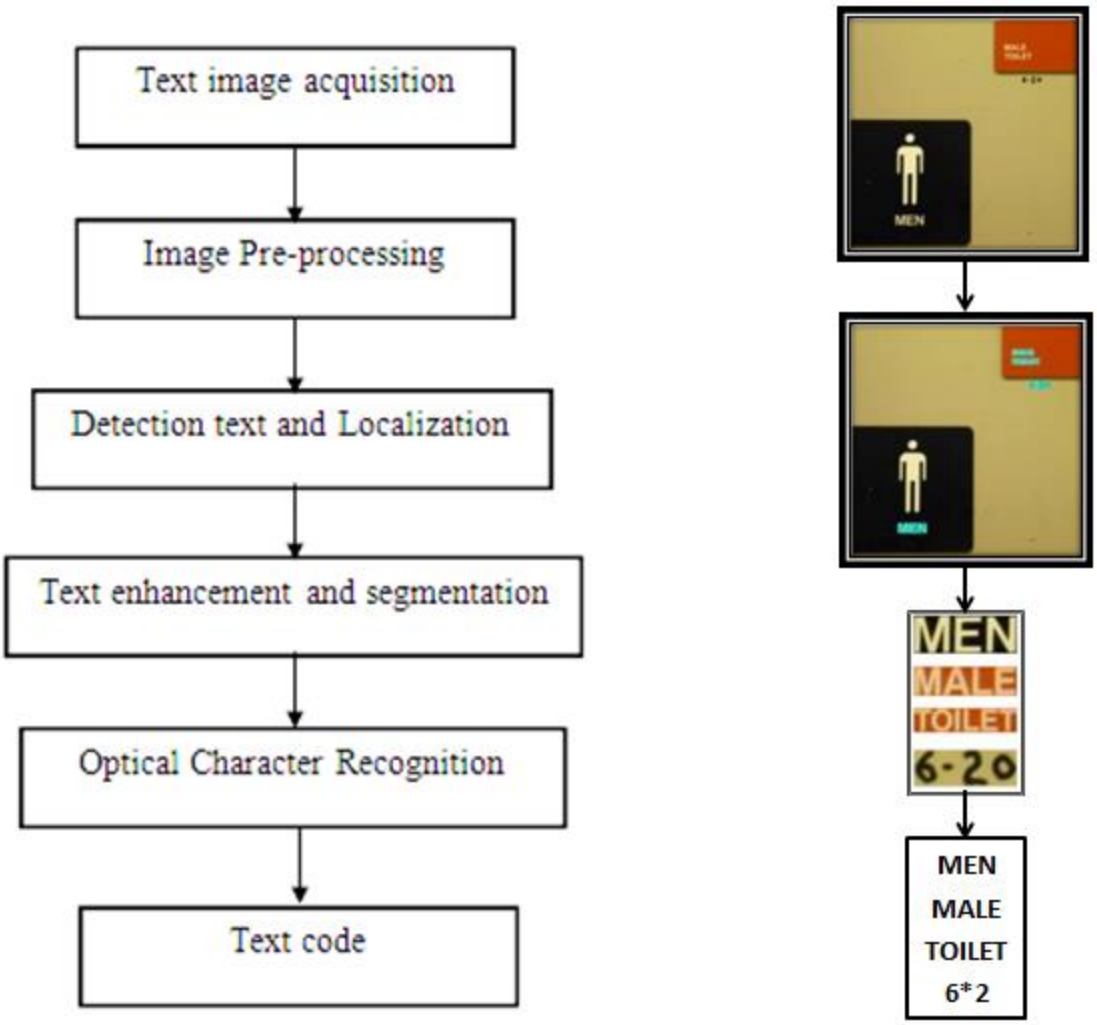


Figure1.1: A flowchart of the processes of text detection and recognition [11].

1.1 Thesis statement

Text is classified as one of the most important resources that enable people to communicate and gather information from their surroundings. Therefore, extracting text from complex backgrounds, or images in nature, is helpful for visually impaired and blind persons. The availability of cameras in many portable devices, such as mobiles and tablets, has allowed researchers to draw new focus to the challenges that face current text

extraction methods. Nowadays, cameras are easy to use and have high performance, which leads to the development of computer vision assistive technologies for using these devices in order to improve the quality of living for visually impaired and blind persons.

This thesis is focused on investigating algorithms that are specifically related to text detection and localization and presenting a new method that identifies text boxes. It also computes the angle of the unaligned text using its geometrical properties and realigns it into the horizontal axis. This alignment will enhance the results of the OCR reader/system.

1.2 Objective of this work

The goal of this work is designing a system to assist the visually impaired in recognizing text from indoor signage using a camera mounted on their glasses. Therefore, the focus will be on developing a system that has the following capabilities:

1. Utilizing an efficient algorithm to detect text from signage.
2. Providing horizontally-aligned and text-only objects to enhance OCR translation from signage images into text file.
3. Converting the output text to speech.

1.3 Thesis outline

The thesis consists of five chapters. Chapter 1 provides an introduction and background into the importance of text, the challenges faced by visually impaired and

blind persons, and the stages involved in current text extraction and recognition methods. Chapter 2 evaluates pertinent literature detailing techniques that measure the performance of algorithms. The proposed system is described in detail in chapter 3. In chapter 4, the simulation results are presented. Finally, chapter 5 includes a summary and conclusion of this work including recommendations for future work.

CHAPTER 2

LITERATURE AND BACKGROUND

2.1 Text detection and recognition methods

Text detection and recognition have been widely proposed because embedded text in images provides important information to the reader. Each method gives robust results for specified set of images. However, results may vary due to image size, orientation, contrast, and color. The methods of text detection and localization explored can be classified with respect to their base, such as edge-based, stroke-based, MSERs/ERs based and other methods [3]. Table 2.1 lists the performance measures of methods proposed and carried out in the International Conference on Document Analysis and Recognition (ICDAR) 2003/2005/2011 datasets.

Precision, recall, and F-measure are quantities that measure performance. Precision measures the confidence of the algorithm while recall measures the sensitivity of the algorithm. F-measure is the harmonic average of precision and recall. Precision is defined by the number of true positives divided by the sum of all true positives and false positives. True positives are represented by true predicted elements with false positives representing falsely predicted elements. Recall is the number of true positives divided by the sum of the true positives and false negatives [5]. A false negative occurs when an element is missed altogether. The equations that define the aforementioned performance metrics are shown in equations 2.1-2.3.

$$Precision = \frac{Tp}{Tp + Fp} \tag{2.1}$$

$$Recall = \frac{Tp}{Tp + Fn} \tag{2.2}$$

$$F - measure = \frac{2 * Precison * Recall}{(Precison + Recall)} \tag{2.3}$$

Where:

Tp: True positives

Fp: False positives

Fn: False negatives

When a system obtains high recall but is low in precision, it returns many results. However, most of its predicted elements are incorrect when compared to the actual elements. On the other hand, when the system obtains high precision but low recall, it returns very few results, yet most of its predicted elements are correct when compared to the actual elements. The ideal system would be capable of obtaining high precision and high recall, returning a high percentage of the elements correctly.

Table 2.1: A list of scores that related to detection and localization methods carried on ICDAR 2003/2005/2011 datasets.

Year	Author	Precision (P)	Recall (R)	F-measure	Average time (s)	Feature
2010	Epshtein et al.[8]	0.73	0.6	0.66	0.94	Text detection, (SWT)
2011	Chen et al. [9]	0.73	0.6	0.66	0.2 (for MSER extraction)	Text detection, (MSERs)
2012	Neuman, & Matas [10]	0.647	0.731	0.687	1.8	Text detection, Extremal regions (ERs), gradient magnitude
2012	Yi, & Tian [13]	0.56/ (0.54)	0.69/(0.68)	0.6/ (0.58)	10.36/ (1.54) for reduced size image	Text localization, stroke orientation, edge distribution
2013	Yi et al. [11] ICDAR 2005/2013	0.56/ (58.09)	0.69/(67.22)	0.6/ (62.32)	1.87 for text reading	Text localization, gradient, stroke orientation, edge distribution
2014	Rajkumar et al. [12]	-	-	-	-	Text localization, Haar Cascade Classifier Algorithm, edge distribution and stroke orientations
-	Wang et al.[14]	0.71	-	-	-	Text localization, color, spatial layout

2.1.1 Edge based methods

Edges are considered reliable features for text detection regardless of color, intensity, orientation, or layout. Edge-based methods are simple and have good performance in text extraction from natural scene images, namely because natural scene images tend to exhibit strong edges. Nevertheless, it is still difficult for these methods to obtain strong edge profiles in images affected by highlights and shadows [6].

Liu and Samarabandu have proposed an edge-based algorithm to extract text for indoor mobile robot navigation [6]. Continuing, they propose a multi-scale edge-based method for extracting text from documents, as well as indoor and outdoor scene images [7]. Strength, density, and the variance of edge orientations are distinctive characteristics of text embedded in images that can be used as main features of detection for the identification of text in scene images. Those features are utilized to output a gray scale image that has the same size as the original image, differing in that the pixel intensity now represents possible text candidates. Once possible candidates are identified by pixel intensity, clustering, filtering, and boundary boxes operations are applied to localize text regions (see Figure 2.1). In the end, accurate binary characters are extracted from the localized text to be fed into an OCR engine. This method is capable of extracting text effectively with respect to color/intensity, style, orientation, font size, and illumination; even in the presence of reflections, perspective distortion, shadows, and complex image backgrounds. Furthermore, edge based algorithms are efficient in extracting text features from real scenes. Therefore, they can be applied in different application fields such as

mobile robot navigation, object identification, vehicle license detection/recognition, document retrieving, and page segmentation. These two methods obtain (91.8%) precision and (96.6%) recall [6, 7].

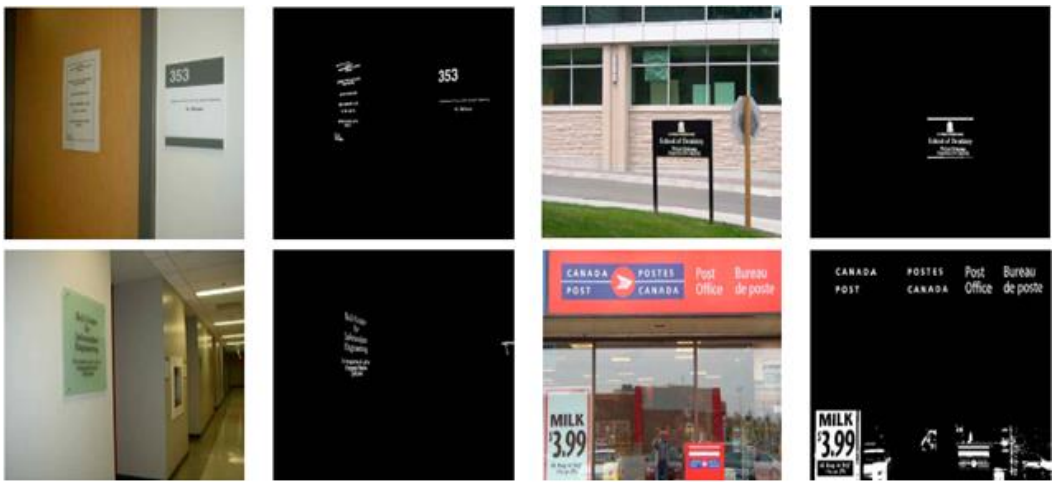


Figure 2.1: Indoor nameplate and outdoor sign images with their associated extracted text grey scale images. Note the differences in font size, perspective distortion, color, and strong reflection between each case [6].

2.1.2 Stroke based methods

The strokes in natural scene images give robust text detection features. The integration of orientations coupled with features of the text that contain stroke components provides an opportunity to model the text. Stroke width is the one feature that separates text from other elements of a natural scene because it typically remains constant. Therefore, stroke width can be used to recover the regions of the natural scene that are likely to contain text. Stroke based methods are easy to implement in certain applications because of their anticipatory nature and simplicity. These methods usually

extract text stroke candidates by segmentation, verify them by feature extraction and classification, and group them by clustering. However, it becomes difficult to segment and verify text strokes in complex backgrounds [8].

Epshtein et al. provides a suitably fast and strong operator, utilizing a combination of geometric reasoning to obtain reliable detection of text [8]. A flowchart of their described method is shown in Figure 2.2 below.

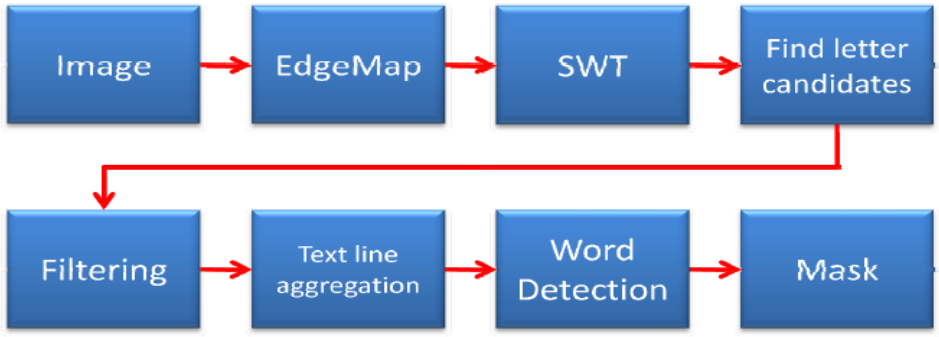


Figure 2.2: The flowchart of the algorithm described by Epshtein et al. [8].

The operator utilized in stroke width methods is called the Stroke Width Transform (SWT), which transforms image data from color values per pixel to most likely stroke width. Since the stroke width of the text is typically constant in scene images, stroke width based text detection systems typically operate independent of the scale, font, language, and direction of text within images. These operations put pixels of similar stroke width into connected components in order to detect letters in a range of scales within the same image. As a result, the need to utilize a scanning window approach is not required. Language filtering techniques also do not need to be utilized, allowing

stroke width detection algorithms the ability to extract multilingual text. Additionally, stroke based methods retain enough information about the text, by way of accurate segmentation and good mask detection, to enhance the recognition by an OCR engine, as depicted in Figure 2.3.

Figure 2.3: OCR results on the original image and on the recovered text segmentation masks. Columns, from left to right: original image, OCR output on the original image, text segmentation mask (superimposed on gray level versions of original images), OCR output on the masks [8].

2.1.3 The Extremal Regions (ERs)/ Maximally Stable ERs (MSERs)

The MSERs based text localization is a widely proposed method in which MSERs are chosen as character candidates. However, due to its sensitivity to blur, large numbers

of components are subject to repetition. To mitigate this issue, MSERs algorithms are combined with pruning algorithms to more accurately select proper MSERs as character candidates [3].

Chen et al. have proposed a method that merges the properties of Maximally Stable Extremal Regions (MSERs) with canny edges, as shown in the flowchart in Figure 2.4 below [9].

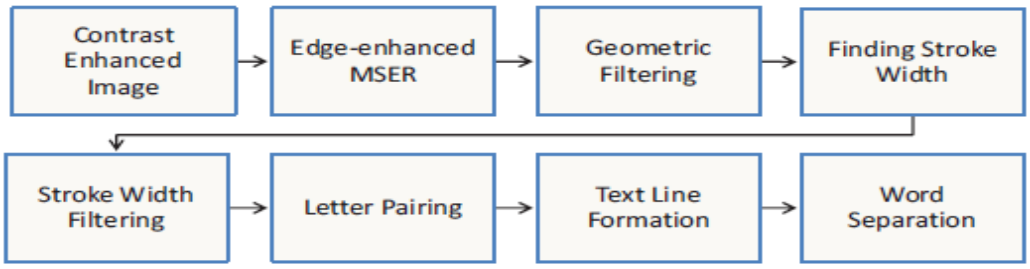
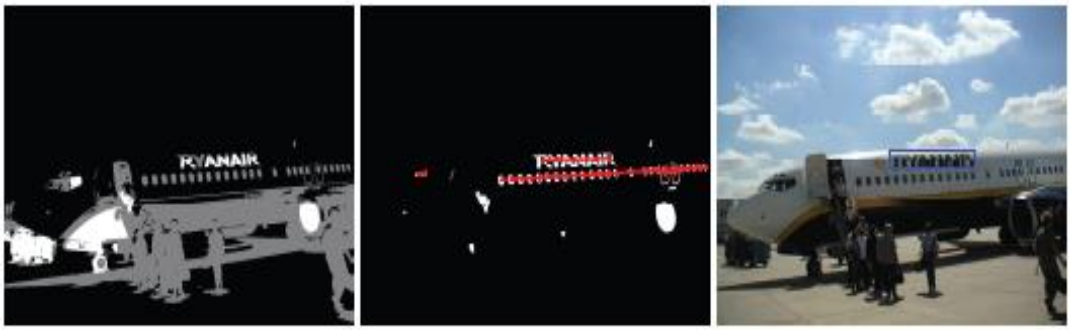


Figure 2.4: The system flowchart described by Chen et al. [9].

Canny edge detectors are applied to identify the boundaries, removing MSER pixels that fall outside the boundaries along the gradient directions. This is done to detect small letters in images with limited resolution. In contrast to the method described by Epshtein's et al., this stroke width method is based on the Euclidean distance transform, or simply, the distance transform. This allows the stroke width transform and geometric image of MSER to give more reliable results. To identify lines of text, letter candidates are clustered. Clusters of letter candidates that are considered “accepted text lines” are then filtered to exclude false positives. At this stage, the text lines are binarized letter

patches which can be fed directly into the text recognition system. Figure 2.5 shows the result of this text extraction method from a natural image.



(a) Detected MSER (b) Text candidates (c) Detected text

Figure 2.5: Extraction of text from a natural image (a) Detected MSER for dark objects on bright background. (b) After geometric and stroke width filtering, text candidates are pairwise grouped to form text lines. The red lines on the image show the identified text lines. (c) Text line verification rejects false positives and the detected text is highlighted by the blue box [9].

Neumann and Matas propose an end-to-end real -time text localization and recognition system [10]. In contrast to the method previously described that utilized a subset of MSERs, this method tests all ERs while reducing the memory footprint, keeping the same computational complexity, and retaining real-time performance. This system is able to select appropriate ERs in real time via a sequential classifier that operates on the basis of a set of features specific to character detection. The ERs are manually extracted from the ICDAR2003 training dataset, which was used to train this method. The method was then evaluated with an ICDAR2011 dataset and a Street View Text-dataset (SVT) (see Figure 2.6). This method suffers performance loss against images containing noise and low contrast characters.

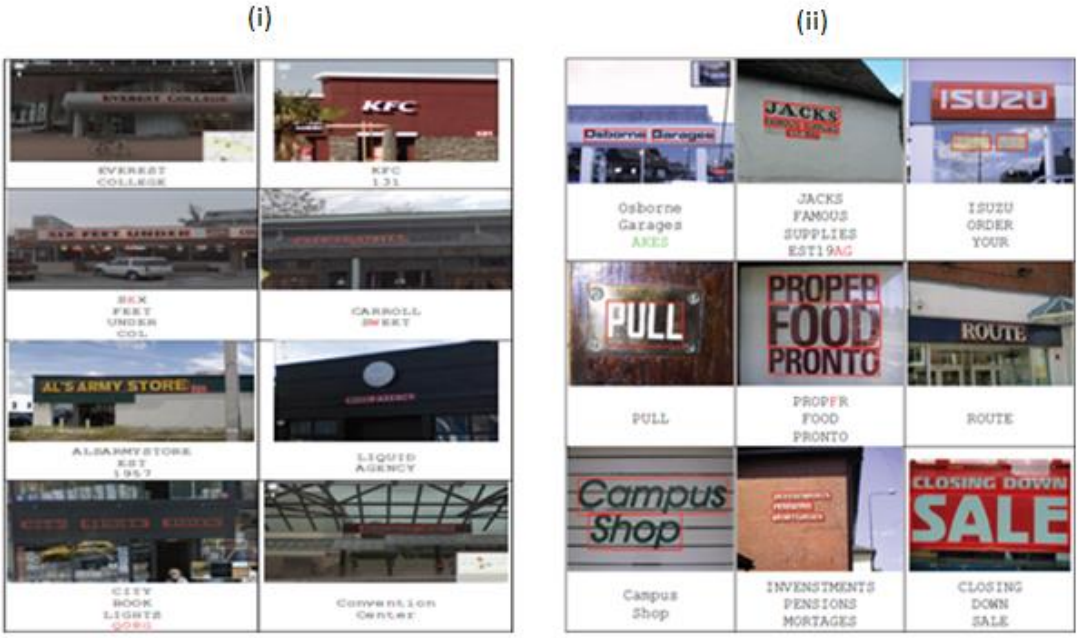


Figure 2.6: Text localization and recognition examples (i) on the ICDAR 2011 dataset and (ii) from the Street View Text Detection dataset (Red letters are recognized incorrectly) [10].

2.1.4 Other approaches

Since text varies so significantly from application to application, single approaches often fail under one or more of a variety of conditions. To improve the robustness through various text categories, researchers have begun developing new methods that utilize a combination of different approaches. As a result, many papers have proposed methods that assist visually impaired or blind people in understanding labels or nearby signage via an audible output [11, 12, 13, and 14].

Yi, C et al. propose a camera-based assistive text reading framework to help blind persons read text labels and product packaging in their daily lives. The flowchart of their described system is shown in Figure 2.7 below [11, 13].

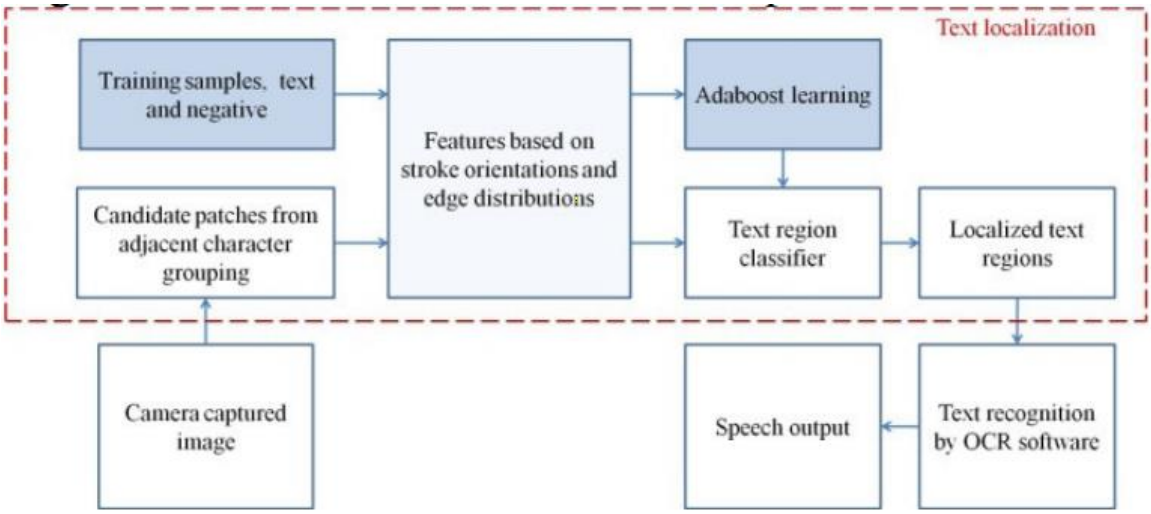


Figure 2.7: The flowchart of the system proposed by Yi, C et al. [13].

The region of interest (ROI), which is the moving part of the frames, is obtained by applying a mixture of Gaussian based background subtraction (BGS) methods. The gradient of stroke orientation and distribution of edge pixels features then are learned through an Adaboost model to recognize the text regions from within the regions of interest (ROIs). In contrast to prior work, this method combines rule-based layout analysis and learning-based text classifier training in order to obtain a text localization algorithm that can deal with extracting text information from complex backgrounds with different text patterns. The proposed algorithm can handle extracting text information from handheld objects as well as nearby signage (see Figure 2.8). With respect to the

orientation of the text, this method assumes that all text strings are kept in a horizontal alignment.

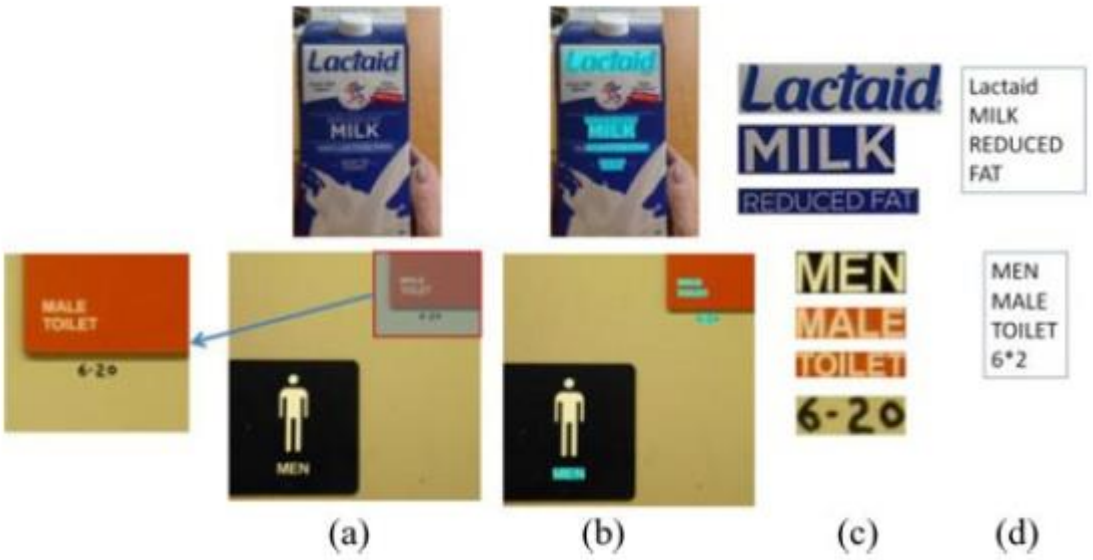


Figure 2.8: Two examples of text localization and recognition from camera captured images. (Top) Milk box. (Bottom) Men's restroom signage. (a) Camera captured images. (b) Localized text regions (marked in blue). (c) Text regions cropped from image. (d) Text codes recognized by OCR. Text at the top-right corner of bottom image is shown in a magnified callout [11].

Rajkumar, N. et al. [12] have proposed another framework that is similar to the work described by Yi, C et al. but differing in which classifier is used. The classifier, the Haar Cascade, was trained on features pertaining to the gradient of stroke orientation and distribution of edge pixels in order to recognize the text regions from regions of interest (ROIs).

Wang, S. et al. [14] have proposed an automatic computer-vision method for helping blind or visually impaired people to understand restroom signs using both text and symbols. For text detection and recognition, it begins with extracting text strings by analyzing the similarity of the color and spatial layout. Then, an open-source OCR engine is applied on the detected text regions to obtain the encoded text (see Figure 2.9).

Figure 2.9: Detected text regions are marked in blue masks in the images (left column). The detected text regions and the corresponding text codes recognized by off-the-shelf OCR are displayed in the right column [14].

This method is also capable of handling symbols with variations in scale, rotation, view angle, perspective projection, and illumination. In the end, the obtained text code and the detected symbols are converted into an audio output for the blind individual to interpret. This system is able to operate on symbolic and text restroom signs. Moreover, it can detect multiple, and even rotated signs. However, large shape distortion, blur, and

low resolution images cause this method to suffer performance loss, and ultimately fail, in detecting and recognizing signage (see Figure 2.10).

Figure 2.10: Examples of images in which the method described by Wang, S et al. fails due to large perspective projections (a) and the small size of signage (b and c) [14].

CHAPTER 3

PROPOSED METHODOLOGY: SIGNAGE RECOGNITION BASED WAYFINDING SYSTEM FOR THE VISUALLY IMPAIRED

3.1 Introduction

The proposed algorithm is a method that extracts text from indoor scene signage images with complex backgrounds by searching them for semantic objects, then extracting identified candidate regions that contain text characters which satisfy our predefined geometrical constraints. These are defined as the candidate character components, since we reject all other non-text background outliers, and are represented by connected components or bounding boxes.

In most cases, text characters of signage will appear in the form of connected components when in uniform color and contrast to its surrounding. This also brings in high edge density to contrast the dominant plain regions within an image. Based on these two layout characteristics of text, the MSERs method is well suited for extracting candidate character components from images and thus our choice for using it. Since the OCR software utilized operates using horizontally aligned text images, detecting and correcting the orientation of text is very essential and must be accomplished. This pre-processing stage, in which aligned and complete text from cluttered background is obtained, is followed by an input into Optical Character Recognition (OCR) and text to speech engines to deliver the audio output to the visually impaired users.

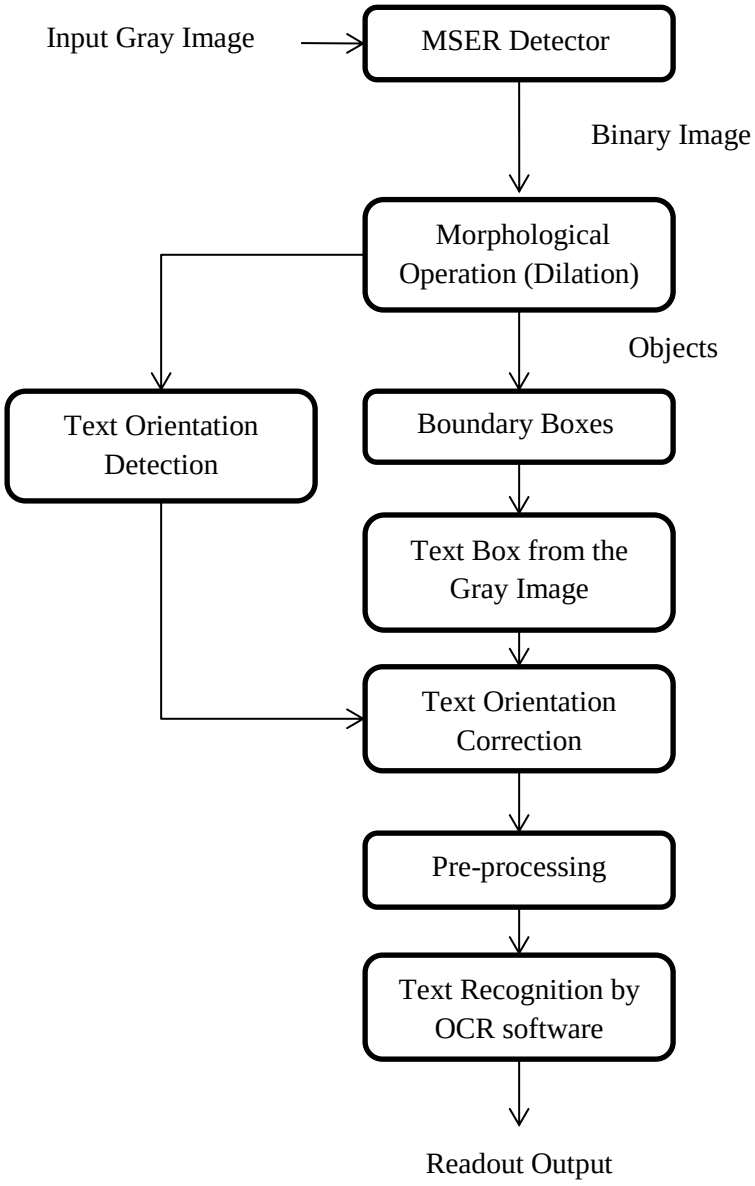


Figure 3.1: The proposed system.

3.2 Maximally Stable Extremal Regions (MSERs) detector

Because Maximally Stable Extremal Regions (MSERs) algorithm only operates on two-dimensional matrices, the three color-coding components (Red-Green-Blue) of matrices that contain a third dimension of color must be summarized into a single value. This process is more commonly known as gray-scaling, where the resulting image only holds different intensities of gray.

The algorithm of Maximally Stable Extremal Regions (MSERs) is a new methodology of extracting stable connected components of some gray-level sets of the image, proposed by Matas et al. [15]. The regions are defined according to their extremal property of the intensity function within and on the outer boundary of the region. Basically, it is an iterative process that gradually joins the pixels of an image together. By nature, it can detect regions of arbitrary geometry making it a desirable tool for general object recognition. Introduction to MSER concept and the necessary auxiliary definitions are given in the following subsection.

3.2.1 MSER- The algorithm

The basis of our proposed algorithm is to identify regions that stay nearly the same through a wide range of thresholds, where the set of gray levels within the image *I* are the possible range of thresholds. In MSER, every extremal region is a connected component of a thresholded image and all possible thresholds are applied to an input image and the stability of extremal regions is evaluated to find MSERs. By using all of

the possible thresholds, each intensity image is able to represent one frame in a sequence of binary images. This concept is shown in Figure 3.2 below.

Figure 3.2: MSER procedure: A sequence of binary images (I_l) at different threshold values (l).

As given in the formal definition proposed by Matas et al., all pixels with intensities that are below a given threshold are set to black and all those above or equal are declared to be white. The algorithm is therefore generally implemented to detect dark MSERs. However, the intensity of input image can be inverted to detect bright MSERs. In Figure 3.2, a sequence of thresholded images I_l or $I(x)$ with frame l corresponding to threshold t is shown. Notice that the first thresholded image is totally white, with black spots corresponding to local intensity minima appearing and growing larger in each subsequent thresholded image. These black spots will keep merging until the whole image becomes black. The full set of connected components in all frames represents the full set of maximal regions. Minimal regions can be obtained by inverting the intensity of the image.

3.2.2 MSER definition

Figure 3.3 shows the extremal region R_l of an image. It is a connected component (see section 3.4) of the level set $S_l = \{x: I(x) \leq l\}$, $l \in \mathbb{R}$. Figure 3.4 describes a situation in which discontinuities can cause multiple extremal regions. The threshold l spans a finite number of values, $l = \{0, \ldots, M - 1\}$, where M is the sampling range of the image.

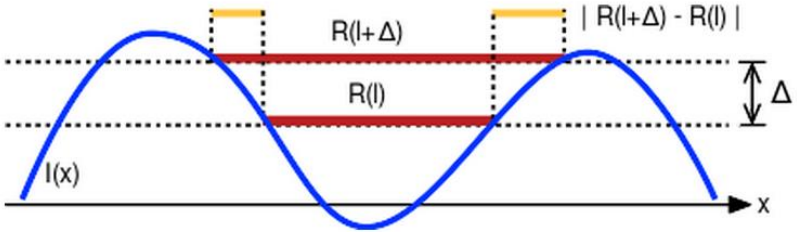


Figure 3.3: Extremal regions are connected components of level sets. Extremal region variation is used to compute stability [16].

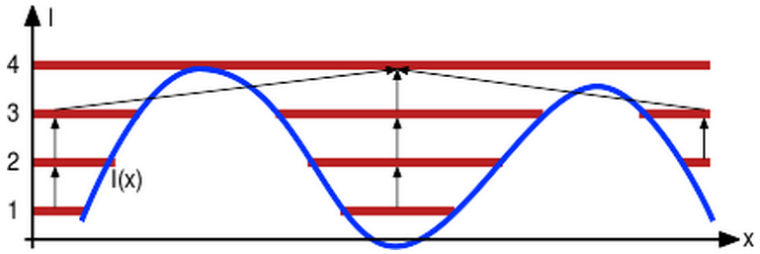


Figure 3.4: Discontinuities can cause multiple extremal regions to coincide [16].

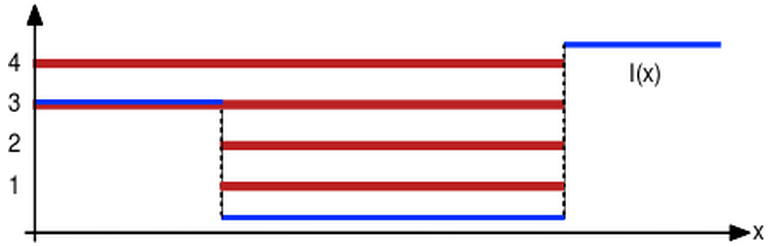


Figure 3.5: Extremal regions are arranged in a tree of nested regions [16].

In each frame, an operation is performed to connect the pixels into regions (see section 3.4). A family tree of regions is represented utilizing arrows in Figure 3.4. This family tree can be obtained by connecting two regions R_l (the children) and R_{l+1} (the parent) if and only if the condition $R_l \subset R_{l+1}$ is satisfied. The extremal region becomes a maximally stable extremal region (MSER) once certain stability criteria are satisfied, described in equation (3.1). The criterion used in this work is not identical to the criterion proposed by Matas et al. [15]. According to Vedaldi and Fulkerson, this algorithm is faster and simpler to understand [16].

Let $B(R_l) = (R_l, R_{l+1}, \ldots, R_{l+\Delta})$ be the branch of the tree rooted at R_l with stability score defined as:

$$v(R_l) = \frac{|R_{l+\Delta} - R_l|}{|R_l|} \tag{3.1}$$

The goal is to select maximally stable branches (corresponding to a low score), which are obtained when regions exhibit similar area and shape. To start, all branches are

assumed to be maximally stable. Then, each branch $B(R_l)$ and its parent branch $B(R_{l+1})$: $R_l \subset R_{l+1}$ are considered (notice that, due to the discrete nature of the calculations, they might be geometrically identical). The one that is less stable is marked as such, i.e.:

- If $v(R_l) < v(R_{l+1})$, mark R_{l+1} as unstable;
- If $v(R_l) > v(R_{l+1})$, mark R_l as unstable;
- Otherwise, do nothing.

For a general definition of MSER see the definition described below by Matas et al. [15].

3.2.3 Region properties

Regions that are stable during local binarization over a large set of thresholds possess the following interesting properties [15]:

• Invariance to affine transformation of image intensities.

• Closed under continuous geometric transformation T: $D \rightarrow D$ on the image domain.

• Stability, since only extremal regions that remain virtually unchanged over a range of thresholds are selected.

• Multi-scale detection. Since no smoothing is involved, both very fine and very large structures are detected.

This operation produces a structure in which the area (size) of each connected component is stored as a function of intensity. In MSER detection, this operation seeks a range of thresholds that leaves the variation in region area effectively unchanged. In the output, each MSER is represented by position of a local intensity minimum (or maximum) and a threshold. To achieve a binary result in the next stage, the index of these

subsets, which contain a threshold, are then taken and given a value of "1." These binarized indexed subsets are placed on a black background as it is shown in Figure 3.6 below.

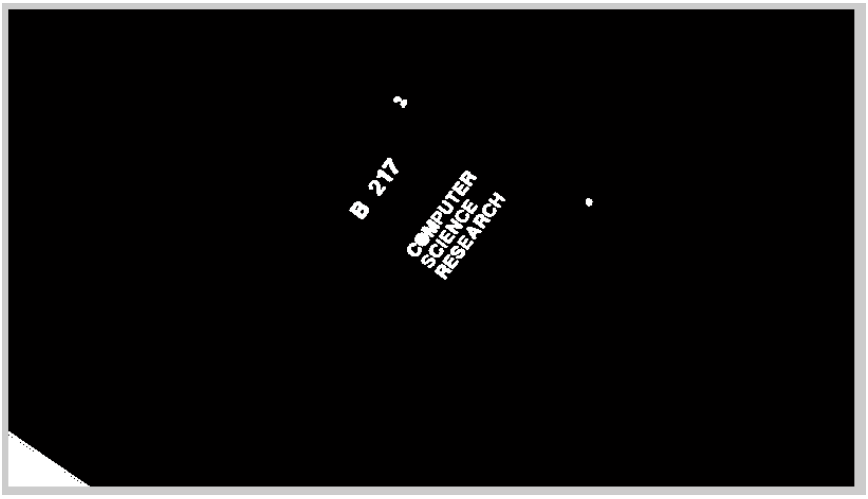


Figure 3.6: A binary image of the MSERs pixels.

3.3 Binary images

Binary images can contain only one of two values at each index, in this case black or white. Binary images are easier to acquire, process, and store than the images that contain many levels of brightness (gray levels). Consider the "wanted regions" in an image as objects in the field of view, and the background as everything else. The objects (foregrounds) are brighter (or darker) than the background as it is shown in Figure 3.7, where a value of "1" is given to all of the images pixels that belong to the objects, and a value of "0" is given to all of the images pixels that belong to the background. The characteristic function is defined as $b(x, y)$. Binary images can also be

obtained via thresholding operators that define the characteristic function to assign that index a value of "1" where the brightness is less than some threshold level and a value of zero where it is not. In these types of images, there are often restrictions as to which processes can be performed and which cannot be performed. For example, various geometric and topological properties that label individual objects allow geometrical computations to be performed separately, simplifying binary images for further processing, can be performed with binary images. On the other hand, despite high-speed hardware computations, some restrictions on binary images should be kept in mind such as the high contrast between the foreground and the background and the fact that the obtained pattern has to be two- dimensional. The characteristic function *b(x, y)* contains a value for each pixel in the image, called the continuous binary image [17].

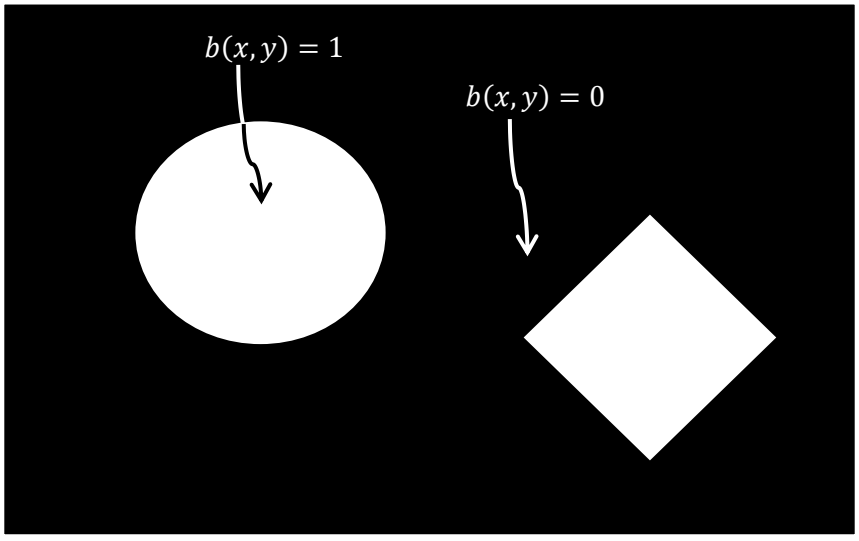


Figure 3.7: Binary image using a characteristic function *b(x, y)* [17].

3.4 Connected Components (CCs) analysis

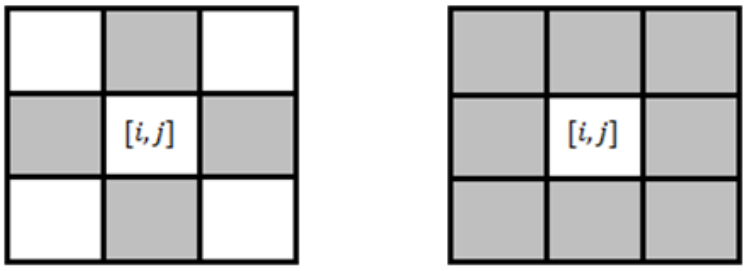
Connected components are defined as a set of pixels in which each pixel is connected to all other pixels. In binary images, the connected component algorithm is used to segment the object pixels from background pixels. Then, labeling and geometric properties can be applied to obtain the properties of each object.

3.4.1 Neighbors and foreground

In image processing, a digital image can be represented as *B[i, j]*, with [*i*, *j*] representing the coordinates of the pixels in the image. For formulating the adjacency criterion for connectivity, the notation of a *neighborhood* is introduced. For a pixel *p* with the coordinates $[i, j]$, the set of pixels considered the *4-neighbors* and *8-neighbors* of pixel *p*, respectively, are given by Jain et, al. as:

4- Neighbors $[i+1, j], [i-1, j], [i, j+1], [i, j-1]$

8- Neighbors $[i+1, j+1], [i+1, j-1], [i-1, j+1], [i-1, j-1]$, including also all of the 4- Neighbors.



(a) 4-Neighbors (b) 8-Neighbors

Figure 3.8: The 4- and 8- neighborhoods for a rectangular image where pixel [*i, j*] is located in the center of each neighborhood [18].

The foreground is another term should be mentioned before defining the connectivity. Foreground is the set of all pixels with value (1) in an image, denoted as *S.*

3.4.2 Connectivity

A pixel $p \in S$ is connected to $q \in S$ if there exists a path from p to q consisting entirely of pixels of *S.*

The connectivity is an equivalence relation. There are several relationships between pixels that should be considered in order to connect these pixels. For any three pixels *p, q,* and $r \in S$, the following properties should be fulfilled [18]:

1. The pixel p is connected to p (reflexivity)
2. If p is connected to q, then q is connected to p (commutativity).
3. If p is connected to q and q is connected to r, then p is connected to r (transitivity).

3.4.3 Component labeling

Component labeling is a common operator utilized in machine vision to find the connected components in an image. Each of the connected components is marked with a distinctive label. The points of each connected component represent an object. It is a necessity when performing this operation in situations when there is more than one object to find properties and locations of each object individually. Figure 3.9 shows an example of an image with its associated labeled and connected components.

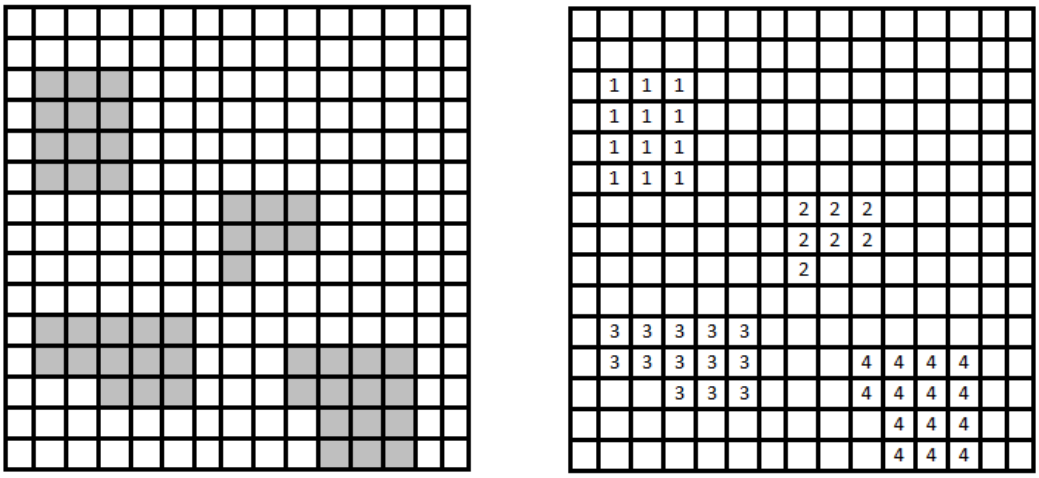


Figure 3.9: An image (on the left) and its connected component image (on the right) [18].

3.4.4 Sequential algorithm

Sequential algorithms are involved in the labeling of connected component algorithms. The processes of these algorithms operate on two rows within an image at a time. These processes are split into two different passes: assigning labels and aggregation (see Figure 3.10). Assigning labels begins by scanning an image pixel-by-pixel from top to bottom, and left to right, to identify connected pixel regions, *i.e.* regions of adjacent pixels which share the same set of intensity values *B. B,* in the example of a binary image, is assigned a value of "1." The algorithm then looks at the neighbors of a pixel and tries to assign pre-existing labels to each pixel valued "1." However, if we find two different labels in the neighborhood of any given pixel, an equivalency table should be constructed to keep track of all labels that are considered equivalent. The second pass utilizes this table to assign a unique label to all of pixels in a component [18].

The algorithm looks at only two pixels of the 4- neighbors (the ones above and to the left of the pixel of interest). The algorithm has already scanned these two 4-neighbor pixels. There exist three cases of interest during the progression of these scans. The pixel assigned as with a new label if there exist no "1" valued pixels in the neighborhood. If only one of the two neighboring pixels is valued "1" and labeled with the label "*L*", the pixel will be assigned a label of "*L*" too. If two neighboring pixels are valued "1" and have both the same label "*L*", the pixel will be assigned a label of "*L*". However, if two neighboring pixels are valued "1" and have different labels, "*M*" and "*N*," these two labels should be merged to represent one connected component. The pixel is then assigned the smaller label and both pre-existing labels are then considered as equivalent in the equivalency table.

All of the information utilized in the assigning of unique labels for each connected component is recorded in the equivalency table. In short, the first pass records all labels that belong to one connected component, while the second pass selects one label from an equivalent set, assigning that label to all pixels of a component. The lowest label is usually assigned to a component. As a result, a distinctive label is assigned to each component [18].

3.4.4.1 Sequential Connected Component Algorithm using 4-connectivity

1. Scan the image left to right, top to bottom.
2. If a pixel is valued "1," then

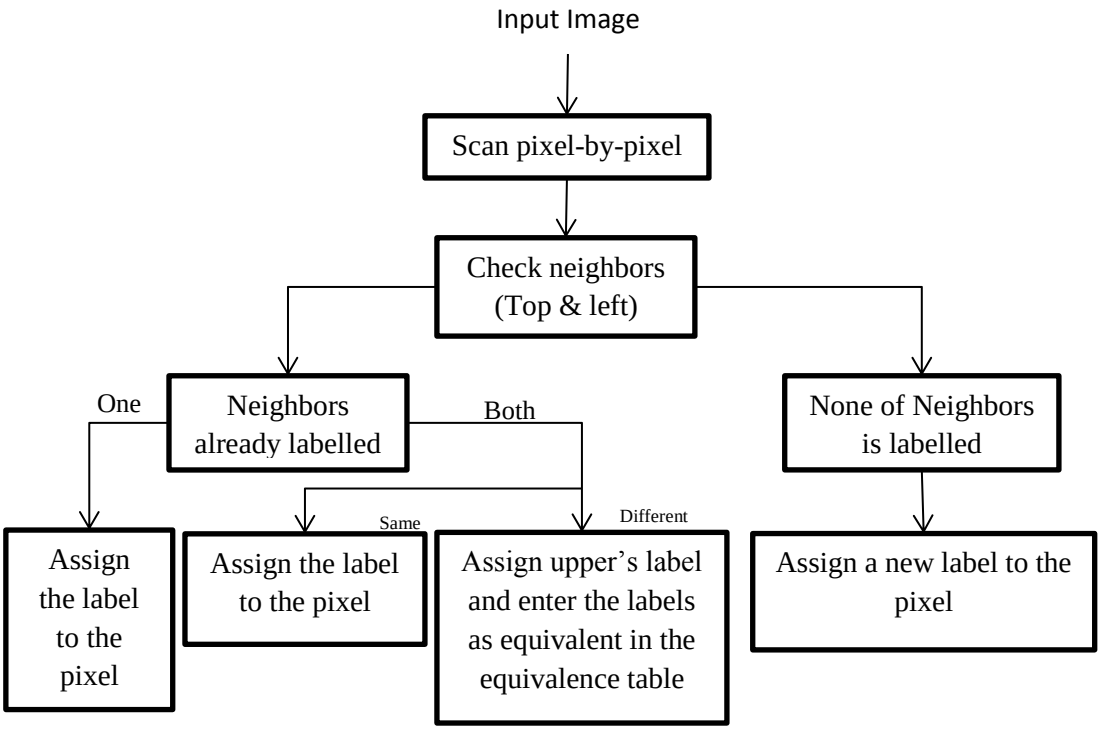
a) If only one of its upper and left neighbors has a label, then copy the label.

b) If both have the same label, then copy the label.

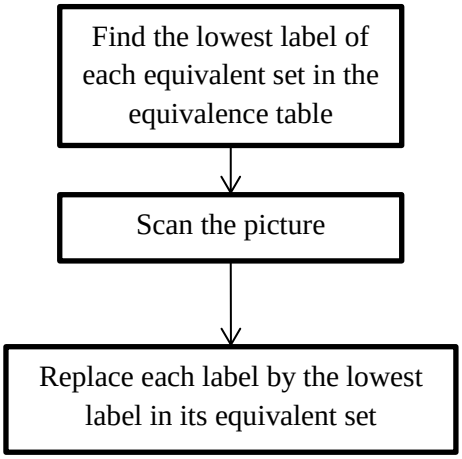
c) If both have different labels, then copy the upper's label and enter the labels in the equivalency table as equivalent labels.

d) Otherwise, assign a new label to this pixel and enter this label in the equivalency table.

3. If there are more pixels to consider, then go to step 2.
4. Find the lowest label for each equivalent set in the equivalence table.
5. Scan the picture. Replace each label by the lowest label in its equivalent set.



(a) The first pass



(b) The second pass

Figure 3.10: The flowcharts of the two passes of
the sequential connected component algorithm.

3.5 Region properties

After labeling the regions, the geometrical properties of regions such as area, position, and orientation can be computed. The analysis of binary regions turns out to be one of the simpler tasks for which many efficient algorithms have been developed and implemented in reliable applications that utilized every day.

The geometrical properties are scalar quantities that describe a function and capture its significant features. They are essential in the field of pattern recognition because they describe objects and characters regarding their position, size and orientation.

3.5.1 Size (area)

Assume for now that a two-dimension digital image, *I*, has a single object with characteristic function $b(x, y)$. In binary images, the size or the area of the object can be obtained, as described by Horn [17]:

$$A = \iint_I b(x, y)dx\ dy \tag{3.4}$$

where the integral is over the entire image *I*. If there exists more than one object, the formula will compute the total area. This equation is at the zeroth- order moment of *b(x, y)*, which counts the white pixels in the binary image.

3.5.2 Position

The representative point that describes the position of an object can be defined as the center of area of that object. The center of area is the center of mass in a scenario where the entire mass of the object can be concentrated without changing the first moment of the object about any axis. In other words, the position of an object defines its spatial location as shown in Figure 3.11. To calculate the position of the object in a binary image, we use methods explained by Horn [17]:

$$\bar{x} \iint_I b(x,y)dx\,dy = \iint_I x\, b(x,y)dx\,dy \tag{3.10}$$

$$\bar{y} \iint_I b(x,y)dx\,dy = \iint_I y\, b(x,y)dx\,dy \tag{3.11}$$

Where $\bar{x}$ and $\bar{y}$ are the coordinates of the center of the area. We notice that the integrals of the left side of the equations 3.10 and 3.11 define the area *A*. Therefore, the area cannot be zero if one wishes compute $\bar{x}$ and $\bar{y}$. These equations represent the first-order moments of $b(x,y)$.

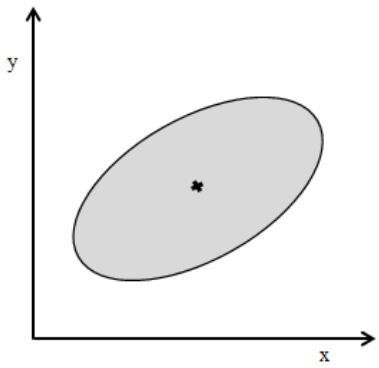


Figure 3.11: The position in a region in a binary image [17].

3.5.3 Orientation

It is complex to calculate the orientation of an object. For instance, orientation is not unique for some shapes, such as circles. The region must first be elongated in order to get a unique orientation. Then, the orientation of the axis of elongation can be utilized to calculate the orientation of the object, as seen in Figure 3.12. This axis of elongation can be defined by choosing the axis of least second moment, which is equivalent to the axis of least inertia. The axis of second moment for a region is a line for which the integral of the squared distances between points in the object reaches a minimum. Once this criterion is met, the resulting line is the axis about which it takes the least amount of energy to spin an object.

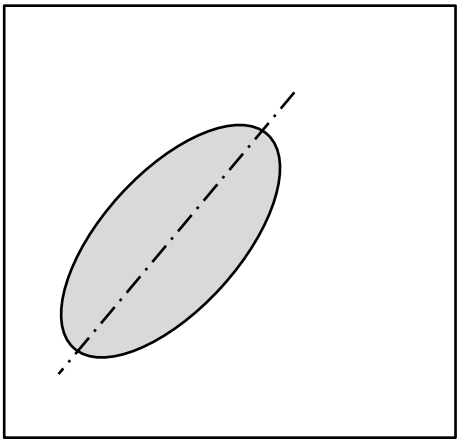


Figure 3.12: The orientation of an object in an image is represented by the direction of the axis of the least inertia [17].

The sum of the squared distances can be defined in terms of integrals as shown in the equations described by Horn [17]:

$$E = \iint_I r^2 b(x, y) dx\, dy \tag{3.12}$$

where (r) is the perpendicular distance from any given point (x, y) to the line (see Figure 3.14).

Selecting a particular line in the plane requires two specified parameters. These convenient parameters are (ρ) and (θ), where (ρ) is the distance from the origin to the closest point on the line and (θ) is the angle between the x-axis and the line, measured counterclockwise (see Figure 3.13).

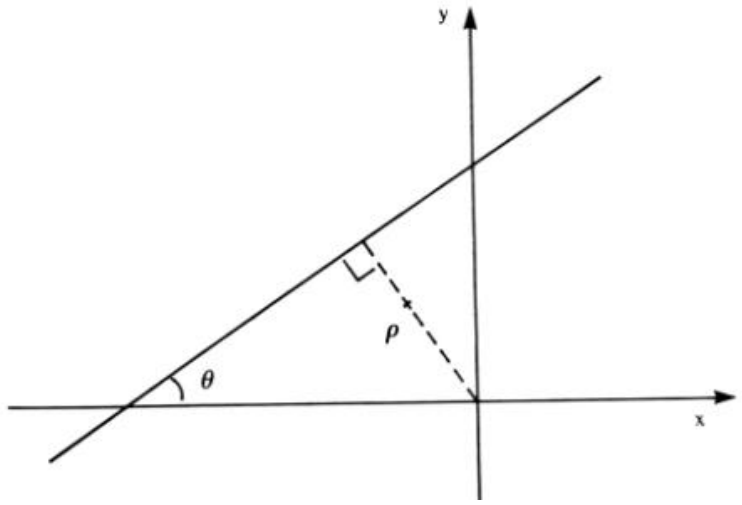


Figure 3.13: The two parameters (ρ and θ) are useful to identify a particular line in the plane [17].

These two parameters are preferred because they change continuously once the coordinate system is rotated or translated. This method is robust in that no problems arise if the line is parallel or nearly parallel to either of the axes. The equation of the line defined by the two parameters (ρ and θ) is given by Horn as [17]:

$$x \sin\theta - y \cos\theta + \rho = 0 \tag{3.13}$$

Geometrically, the line intersects the x-axis at the point $(-\rho/\sin\theta)$ and the y-axis at the point $(+\rho/\cos\theta)$. The point placed on the line nearest to the origin is $(-\rho\sin\theta\,, +\rho\cos\theta)$. Parametric equations can now be written for the points on the line described by Horn as [17] :

$$x_0 = -\rho\sin\theta + s\cos\theta \qquad and \qquad y_0 = +\rho\cos\theta + s\sin\theta \tag{3.14}$$

where (s) is the distance along the line from the point nearest to the origin.

The point (x, y) is a given point on the object. The closest point (x_0, y_0) on the line should be obtained to compute the distance (r) between the point and the line (see Figure 3.14). So, the distance (r) as described by Horn, will be [17]:

$$r^2 = (x - x_0)^2 + (y - y_0)^2 \tag{3.15}$$

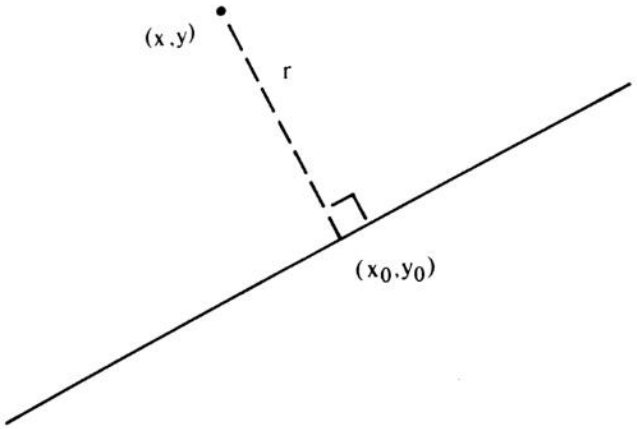


Figure 3.14: The perpendicular distance (r) from a point (x, y) on the object to a line can be found easily, once the closest point on the line (x_0, y_0) is defined [17].

Substituting the equations in (3.14) in equation (3.15) results [17]:

$$r^2 = (x^2 + y^2) + \rho^2 + 2\rho(x \sin\theta - y \cos\theta) - 2\, s\, (x \cos\theta + y \sin\theta) + s^2 \qquad (3.16)$$

Deriving with respect to s and setting the result equal to zero gives:

$$s = x \cos\theta + y \sin\theta \qquad (3.17)$$

Substituting equation (3.17) in the equations in (3.14) results [17]:

$$x - x_0 = +\sin\theta\, (x \sin\theta - y \cos\theta + \rho),$$

$$and \qquad y - y_0 = -\cos\theta\, (x \sin\theta - y \cos\theta + \rho) \qquad (3.18)$$

Substituting the equations in (3.18) in equation (3.15) results [17]:

$$r^2 = (x \sin\theta - y \cos\theta + \rho)^2 \qquad (3.19)$$

Comparing equation (3.19) to the line equation given in (3.13) shows that the line is the points for which *r*=0 in equation (3.19), which means the way of choosing the line gives the distance from the line directly.

In the end, equation (3.19) is substituted in equation (3.12) to address the minimization, described by Horn as [17]:

$$E = \iint_I (x \sin\theta - y \cos\theta + \rho)^2 \, b(x,y) dx \, dy \tag{3.20}$$

Deriving equation (3.20) with respect to ρ and setting the result to zero gives:

$$A \, (\bar{x} \sin\theta - \bar{y} \cos\theta + \rho) = 0, \tag{3.21}$$

where $(\bar{x}, \bar{y})$ is the center of area. Thus, the axis of least second moment passes through the center of area. This suggests a change of coordinates to $x' = x - \bar{x}$ and $y' = y - \bar{y}$, for:

$$x \sin\theta - y \cos\theta + \rho = x' \sin\theta - y' \cos\theta \tag{3.22}$$

Substituting in equation (3.20) and simplifying results, described by Horn as [17]:

$$E = a \sin^2\theta - b \sin\theta \cos\theta + c \cos^2\theta \tag{3.23}$$

where a, b, and c are the second moments given by Horn as [17]:

$$a = \iint_{I'} (x')^2 \, b\,(x,y)dx'dy',$$

$$b = 2\iint_{I'} (x'y')\, b\,(x,y)dx'dy', \qquad (3.24)$$

$$c = \iint_{I'} (y')^2 \, b\,(x,y)dx'dy',$$

The formula for E can be written, as described by Horn [17]:

$$E = \frac{1}{2}(a+c) - \frac{1}{2}(a-c)\cos 2\theta - \frac{1}{2}b\sin 2\theta \qquad (3.25)$$

Deriving equation (3.25) with respect to θ and setting the result to zero gives equation 3.26, described by Horn as [17]:

$$\tan 2\theta = \frac{b}{a-c} \qquad (3.26)$$

Note that if b=0, and c=a, the object does not have a unique axis of the orientation, where the orientation of the axis is given by Horn as [17]:

$$\sin 2\theta = \pm\frac{b}{\sqrt{b^2+(a-c)^2}}, and \quad \cos 2\theta = \pm\frac{a-c}{\sqrt{b^2+(a-c)^2}} \qquad (3.27)$$

From the equations in (3.27), the positive result equation leads to the desired minimum for E. Conversely, the negative result equation leads to the desired maximum for E (which can be shown by observing the second derivative for E with respect to θ).

Finally, the ratio of the maximum to the minimum of *E* gives information about how rounded the object is, with a value of one corresponding to a circle and a value of zero corresponding to a straight line.

3.6 Mathematical morphology

Mathematical morphology is a method that deals directly with the shape and structure of image components such as skeletons, boundaries, and convex hull. It is an effective technique in image processing and can be applied to gray scale or binary images. Because of the simplistic nature of binary images, mathematical morphology is still more easily applied to binary images. This is because the black and white pixels are naturally form two sets, the foreground and the background. These morphological techniques are beneficial also for pre- and post-processing, such as in morphological thinning, pruning, and filtering. Two basic morphological operations are dilation and erosion. These operations enlarge or reduce an object in an image, based on another object called the "structure element" described by Gonzalez et al. [19]. In this context, dilation will be given consideration as it applies to the method proposed in this thesis. The objective in utilizing this technique is to combine the letters in each word such that they are represented by one region. This is done in order to generate bounding boxes on each individual letter in the combined region, and also to detect the text orientation.

3.6.1 Structuring Element (SE)

Reflection and translation operations are applied to a set of pixels (*B*) in a binary image (white or black depends on the convention) in order to create the structuring

element, where its values are members of the 2-D integer space Z^2. Each pixel of *B* has (*x*, *y*) coordinates. $\hat{B}$ denotes the reflection of *B*, which is defined:

$$\hat{B} = \{w | w = -b, \quad for\ b \in B\} \tag{3.28}$$

$\hat{B}$ is the points of set *B* with replacing their coodinates to $(-x, -y)$

w: is an element of the set *B*, as it is shown in Figure 3.15.

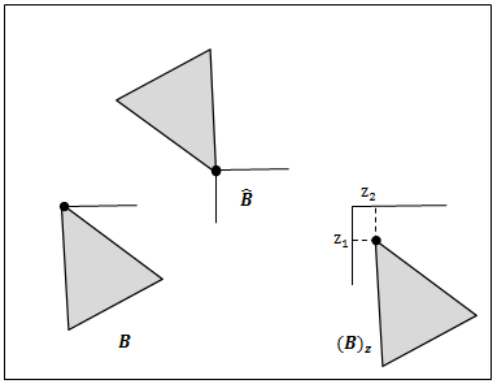


Figure 3.15: *B* set with its associated reflection, $\hat{B}$ and translation, $(B)_z$ as described by Gonzalez et al. [19].

Translation is another operation that replaces a set at a certain point. Consider replacing the set *B* to the point $z = \{z_1, z_2\}$ to get the replaced set $(B)_z$ as it is defined by Gonzalez et al. as follows [19]:

$$(B)_z = \{c \mid c = b + z, \ for\ b \in B\} \tag{3.29}$$

$(B)_z$ are the points of set *B* that are replaced in their coodinates to $(x + z_1, y + z_2)$, as shown in Figure 3.15.

The structuring element is defined as an array, or small image, with logical values that are used as a moving window. It is generally a square dimension in size; they can be 3x3, 5x5, and sometimes larger depending on the application. Each logical value can take on the value 0, 1; the shaded square is not a member of the SE. The 1's make the shape of the structuring element, *i.e.* lines, circles, diamonds, and rectangles as shown in Figure 3.16.

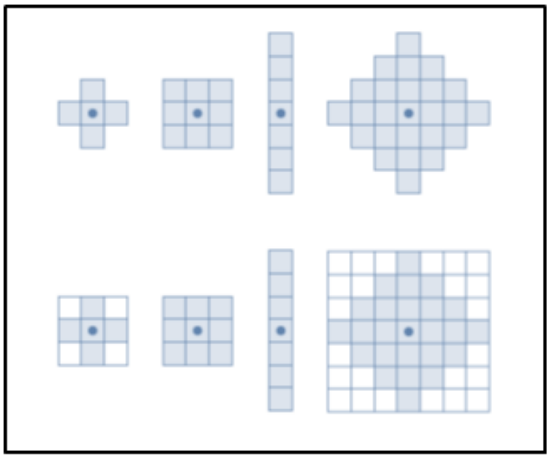


Figure 3.16: 1st row: some shapes of SE, 2nd row: SE converted to rectangular array [19].

The structuring elements (SEs) are converted to rectangular arrays to work with images. The dot in the middle represents the center of symmetry.

3.6.2 Dilation operation

The dilation operation is one of the bases of morphological processing, known as the operation of "thickening" objects in a binary image. It is controlled by the shape of

the structuring element. Mathematically, dilation is defined as a set of operations. A is dilated by B, written as A⊕B, is given as described by Gonzalez below [19]:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \tag{3.30}$$

$\emptyset$ denotes the empty set, B denotes the structure element, and $\hat{B}$ denotes the reflection of set B.

In short, after flipping the set B about the origin and then displacing it, it slides over the image A, to indicate A is dilated by B. Therefore, when the center point of the SE touches a pixel that is a part of A, and another pixel of the SE overlaps with the background, the operation is set to ON (set 1 to the background pixels that are coincided with the SE pixels). This operation adds a layer of pixels around the periphery of all the regions, which results in dimension increment. This may cause images to merge or expand, as shown in Figure 3.17.

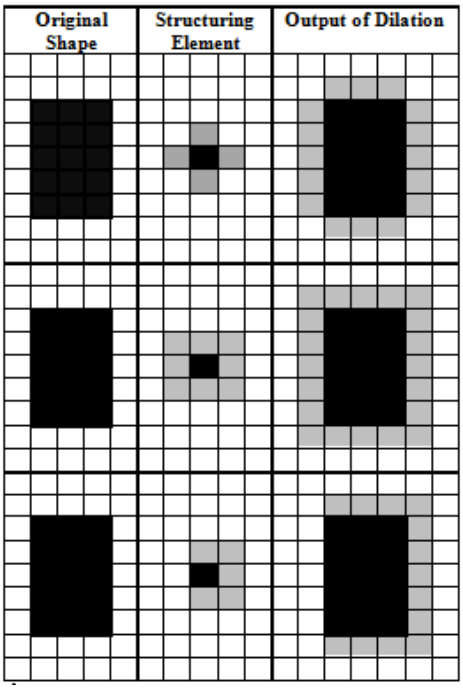


Figure 3.17: Dilation operation as described by Soriano [20].

3.7 Boundary boxes

Boundary boxes are the smallest rectangles that can contain a region, or a set of points, such that all the points are inside it or lie exactly on one of the boundaries sides. In two-dimensional images a bounding box is defined by all of (*x*, *y*) coordinates that satisfy $x_{min} \leq x \leq x_{max}$ and $y_{min} \leq y \leq y_{max}$; these are specified by the extreme points P_{min} (x_{min}, y_{min}) and P_{max} (x_{max}, y_{max}). The four sides of the boundary rectangle are always parallel to the x or y axis, either vertical or horizontal. Since this work is dealing with labeled binary regions, it is easy to detect the maximum and the minimum coordinates for each region. Then, the height and the width of the region can be obtained according to the points P_{min} and P_{max}. This allows detection of the rest of the corners of

the triangle. In brief, the procedure essentially identifies the four corners of the rectangle after detecting the max and the min points above as:

$$P_1\,(x_{min}, y_{min})\ ,\ P_2\,(x_{min}, y_{max})\ ,\ P_3\,(x_{max}, y_{max}),\ P_2\,(x_{max}, y_{min})$$

3.8 Text orientation detection

In analysis of text images and recognition systems, orientation detection and correction each play an important role. Unaligned text severely degrades the performance of text recognition systems. Since text recognition analysis cannot proceed when the text is unaligned, even at a small angle, orientation detection and correction techniques are added to the proposed system in order to achieve a more reliable and adaptive system.

After dilating the binary image, words will be merged into objects; the orientation of each object will be detected (see section 3.5.3) in this stage in order to compute the angle. These angle ranges from -90 to 90 in counter-clockwise direction.

3.9 Text orientation correction

The Affine transformation is one of the most common spatial coordinate transformations. Rotation transformations are used to correct the rotation of the text from the detected angle of each text (as an object). It is a circular transformation around an axis, or a point. In this proposed method, the rotation is considered around an axis defined by the normal vector to the image plane located at the center of the image, as described by Gonzalez et al. [21]. The process starts with translating the center of the image to its origin in order to rotate it, and then translate it back to its original location.

The form of rotation matrix is described by Gonzalez et al. as [21]:

$$[x \quad y \quad 1] = [v \quad w \quad 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.31}$$

The form of translation matrix is described by Gonzalez et al. as [21]:

$$[x \quad y \quad 1] = [v \quad w \quad 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \tag{3.32}$$

Where (v, w) are pixel coordinates in the original image and (x, y) are the corresponding pixel coordinates in the transformed image. And (t_x, t_y) is the coordinate of the translating point.

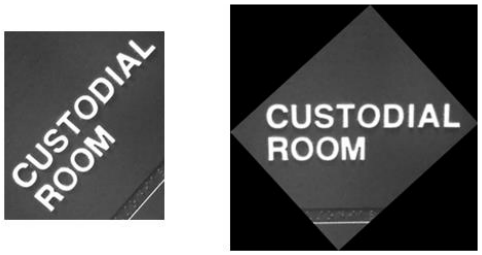


Figure 3.18: Left: original image, right: rotated image by 45 degree clockwise.

3.10 Post-processing

Post-processing stages are required based on the objectives of this work. Since this work aims for a system to be used by visually impaired and blind persons, the output

should be clear, completed text. Therefore, binarization, dilation (see section 3.6.2), and projection techniques are all applied to reduce the high probability of error from the OCR results.

3.10.1 Binarization

Binarization is the process that converts an image of up to 256 gray levels into an image that contains only two values: “0” and “1,” black and white respectively. It is a useful step in the pre-processing of text before putting it through an OCR engine. In fact, most OCR packages on the market today can only operate on bi-level (black and white) images. Thresholding is a simple way to get a binary image from a gray style image. Thresholding is achieved by classifying all pixels with values above a defined threshold as white, and all other pixels as black. Selecting the right threshold that is compatible to the entire image, however, is not easy. Therefore, different adaptive binarization methods are needed.

3.10.2 Projections

Image projections are represented as one-dimensional vectors of the image contents. They are not unique. In fact, many images may have the same projection. There is still, however, useful information in the projections. Projections of binary images are the number of “1” pixels along the vertical and horizontal directions as shown in Figure 3.19 below. The equations of the projections of a binary image are given by Jain et al. as [18]:

$$H[i] = \sum_{j=1}^{m} B[i,j] \quad and \quad V[j] = \sum_{i=1}^{n} B[i,j] \tag{3.33}$$

Where $B[i,j]$ is the discrete binary image with rows (i) and columns (j), where[i = 0, 1,…. n-1] and [j =0, 1, …… m-1], and $H[i]$ and $V[j]$ denote the horizontal and vertical projections respectively.

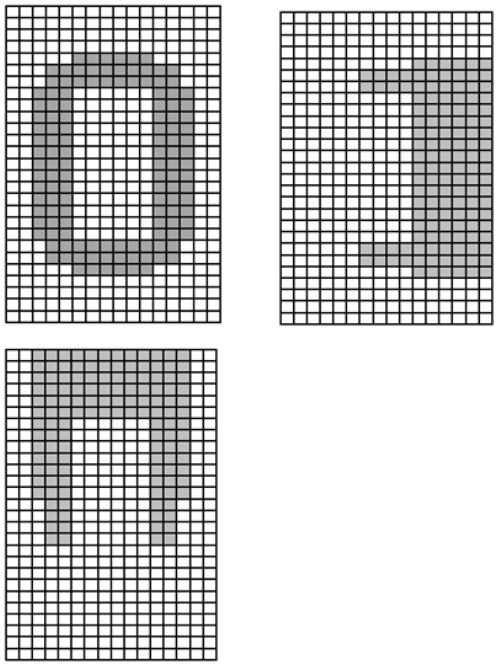


Figure 3.19: A binary image with its horizontal and vertical projections [18].

The pre-processing stage begins with dilating the binarized image vertically to connect the words from different lines and get a complete horizontal projection of the text. An indicator is placed in the middle of the projection and goes up, and down, to save the location at which the summation of the row equates to zero. Next, the top and the bottom “saved locations” are used to crop the binarized image horizontally. The resultant

image from this step is then dilated horizontally to connect the letters to achieve a connected vertical projection. Then, same process is applied to save the right and the left locations so that the image can be cropped vertically to identify the region of interest, which consists entirely of text (see chapter4).

3.11 Optical Character Recognition (OCR)

Optical character recognition (OCR) is an important research area in pattern recognition. It is known as the electronic or mechanical translation of scanned images of handwritten, typewritten, or printed text into machine-encoded text. Many applications of OCR have been used, such as enabling the editing of a text document in a physical form, or enabling the ability to search for terms in a document that was scanned into a computer from a printed form. The engines of OCR are commonly used to digitize text documents so that they can be digitally stored for remote access, mainly for websites. This facilitates the instantaneous availability of these priceless resources, no matter the location of the end user. However, OCR applications often suffer performance losses in situations such as distortion, or dimmed light, which make it difficult for the OCR engine to correctly recognize text. Therefore, pre- and post-processing stages are required in order to present a more suitable image to the OCR engine to get reliable results. OCR deals with the recognition of characters obtained by optical mechanism, such as a camera or a scanner. These engines are even capable of recognizing both handwritten and printed text. Their performance, however, depend on the quality of the input document. OCR systems are designed to work on images that consist almost entirely text, with very little

non-text clutter [22]. The primary function of OCR system is to recognize numbers, alphabetic letters, or other characters, from digital images, without any human involvement.

3.11.1 Tesseract OCR engine

The Tesseract OCR engine is considered the most accurate open source OCR engine available. It is capable of reading and converting a wide variety of image formats to text in over 60 languages. Tesseract was released by HP as an open source technology for free use. In 1995, it was one of the top 3 engines ranked by the UNLV Accuracy test. Between 1995 and 2006, it garnered steady improvement, but since has been developed extensively by Google. Tesseract is available from Google at https://code.google.com/p/tesseract-ocr/ [23].

3.11.2 Architecture of Tesseract

Tesseract has independently developed page layout analysis technologies. It works on binary images, where the text is either white on a black background or black on a white background. It stores the outlines of component on connected component analysis and nests the outlines together to form a blob. Such blobs are organized into text lines. Then, the lines and regions are analyzed to search for fixed pitch and proportional text. The lines are segmented into words by analysis based on the character spacing. Fixed pitch is chopped into character cells, and proportional text is broken into words by definite spaces and fuzzy spaces. The word recognition of Tesseract consists of two passes. The first pass tries to recognize words, with satisfactory words being passed along

to an Adaptive Classifier as training data, which can then recognize the text more accurately. Through the second pass, words which have not been recognized in the initial pass are analyzed again through a second run over the page (see Figure 3.20). In the end, Tesseract resolves fuzzy spaces. To locate capital and small text, Tesseract checks alternative hypotheses for x-height [24].

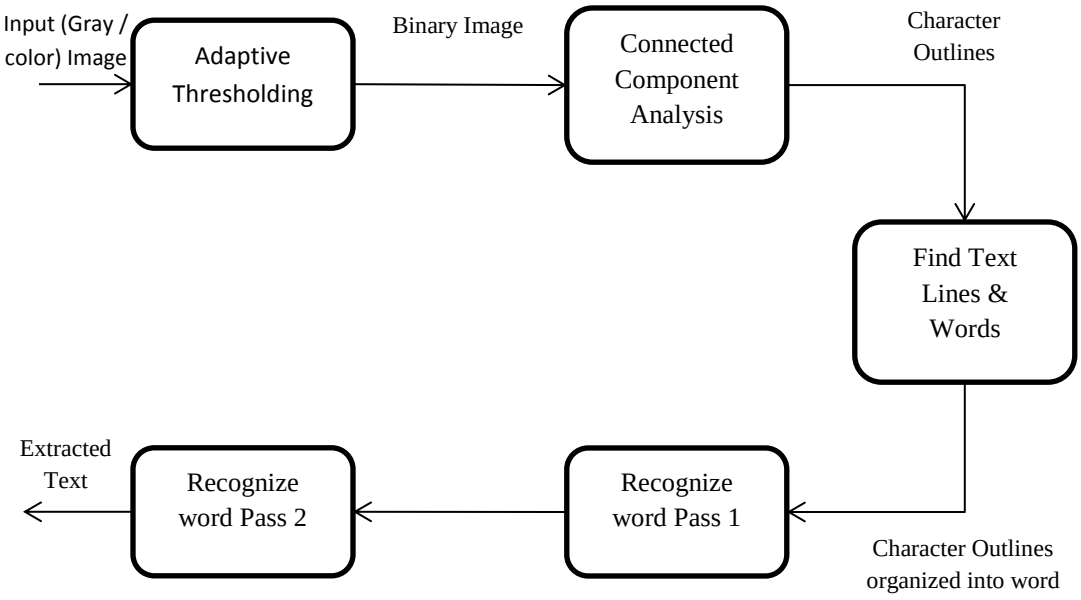


Figure 3.20: Architecture of Tesseract [25].

3.12 Text to speech

Speech is used as the main source of communication between humans. Speech synthesis is an automatic generation of speech waveforms that can be implemented via

software or hardware systems. The text- to- speech (TTS) synthesizer is able to convert the extracted text by an OCR into speech automatically. Its process is broken down into two main parts: text analysis and the generation of speech wave forms. In text analysis, the received text is translated into a phonetic representation, including correct text-pre-processing, pronunciation, and prosodic information. The generation of speech waveforms is the end product from the text analysis information, as it is shown in Figure 3.21 [21]. Today, synthesized speech systems are software based. These software systems are easy to navigate and modernize while typically being less expensive than their hardware system counterparts. However, hardware devices offer the benefit of better portability.

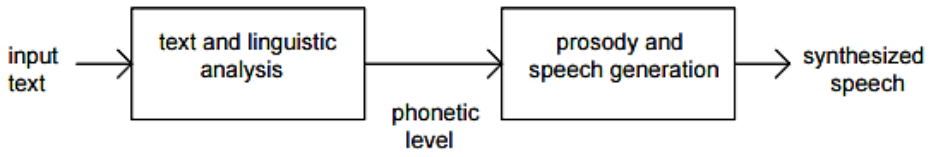


Figure 3.21: Simple text-to-speech synthesis procedure [26].

3.12.1 Speech Application Programming Interface (SAPI)

SAPI is an interface between applications and speech technology engines, which includes both text-to-speech and speech recognition. The interface allows multiple applications to share the available speech resources on a computer without having to program the speech engine itself. Currently, SAPIs are available for several environments, such as MS-SAPI for Microsoft Windows operating systems. This SAPI is

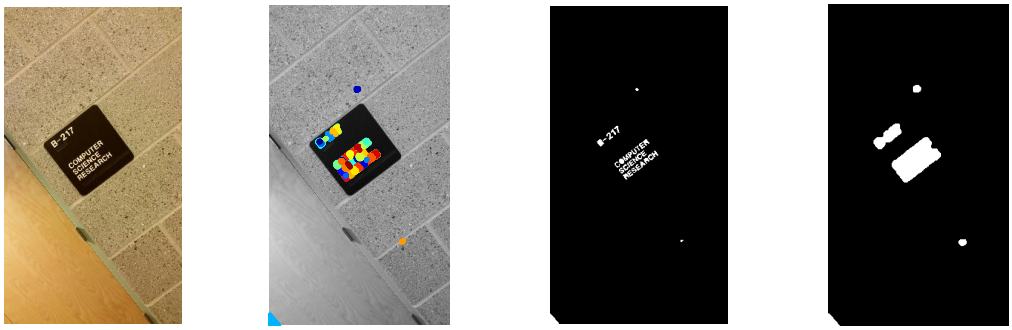
known as Software Developer's kit, which is available for American English as well as German, Dutch, Spanish, Italian, and Korean. In other words, this software can work in harmony with other programs that are utilized to output speech [26].

CHAPTER 4

SIMULATION RESULTS AND DISCUSSION

This chapter discusses the results of the simulation. A computer, with 8GB memory RAM and 2.40 GHz processor Intel(R) core(TM) - i5 CPU was used to execute the simulation along with MATLAB. The dataset consisted of 120 images of the indoor signs inside College of Engineering and Applied Sciences campus. Images taken include a variety of orientations, camera angles and sizes, to test the proposed system. These images were resized into (720x1280). To evaluate the MSER algorithm, we have calculated the precision and recall according to the equations in chapter 2 (2.1 and 2.2). Precision and recall have been obtained from calculating the true positives (Tp), false positives (Fp), and false negatives (Fn) where Tp is the summation of the correctly detected text characters, Fp is the summation of the falsely detected elements, and Fn is the summation of the missed text characters. The results of the detection yielded 0.92 precision and a 0.97 recall value, both values are indicative of high level of performance. The system outputs 82% of the images perfectly as speech. The average time to read text is 0.95s for each word and 3.46s for each image. It takes 3.25s for each word and 5.52 s for each image before the text is spoken. The results of the system will be explained step-by-step using figures 4.1, 4.2, & 4.3.

Figure 4.1 shows the original color image in part (a) and the gray image where the detected pixels of the maximally Stable Extremal Regions (MSERs) are replaced on it in part (b). Those pixels that are within the MSERs are given a value of "1" and placed on a black background to obtain the binary image, seen in part (c). Finally, a dilation operation is applied on the binary image in part (c) in order to cluster the letters into a group, confining each of the words as one object. Figure 4.1 part (d) shows that the words have elongated because of the dilation. Therefore, detecting text orientation is performed by computing the axis of elongation where it is placed in the middle of the object and passes through its center, as explained in chapter 2. Furthermore, cropping text from the gray image is obtained from the shown image (d) in figure 4.1. Figure 4.1 shows the system results from the beginning until dilating the text for the subsequent processes.



(a) (b) (c) (d)

Figure 4.1: Text detection and dilation (a) original image, (b) MSERs placed on gray image, (c) Binary image of MSERs pixels, (d) dilated image.

Figure 4.2 shows the tilting correction using the affine transformation which uses the detected orientation in the previous step. In order to provide a clean, complete, text to

input into the Optical Character Recognition (OCR) engine, the extracted image needs to be thresholded to obtain a binary image for the pre-processing steps. The threshold for converting the gray image into binary image is selected adaptively. Since the signage in CEAS building have text in white on black background, we have calculated the histogram of the images in part (b), as shown in figure 4.2, and ignored the dark half of the histogram which is between the gray levels (0) to (256/2). Then, the mean of the bright half of the histogram, which is between (256/2 to 256), is computed to select which gray level is desired as a threshold. Figure 4.2 part (c) shows the results of thresholding the images of part (b). The selected threshold gives clear text regions with reducing the unwanted regions such as the non-text regions for better recognition by the OCR engine (see part (c) in Figure 4.2).

Figure 4.2: Text extraction and preparation for the pre-processing stages (a) Extracted text, (b) Orientation correction, (c) Binarization.

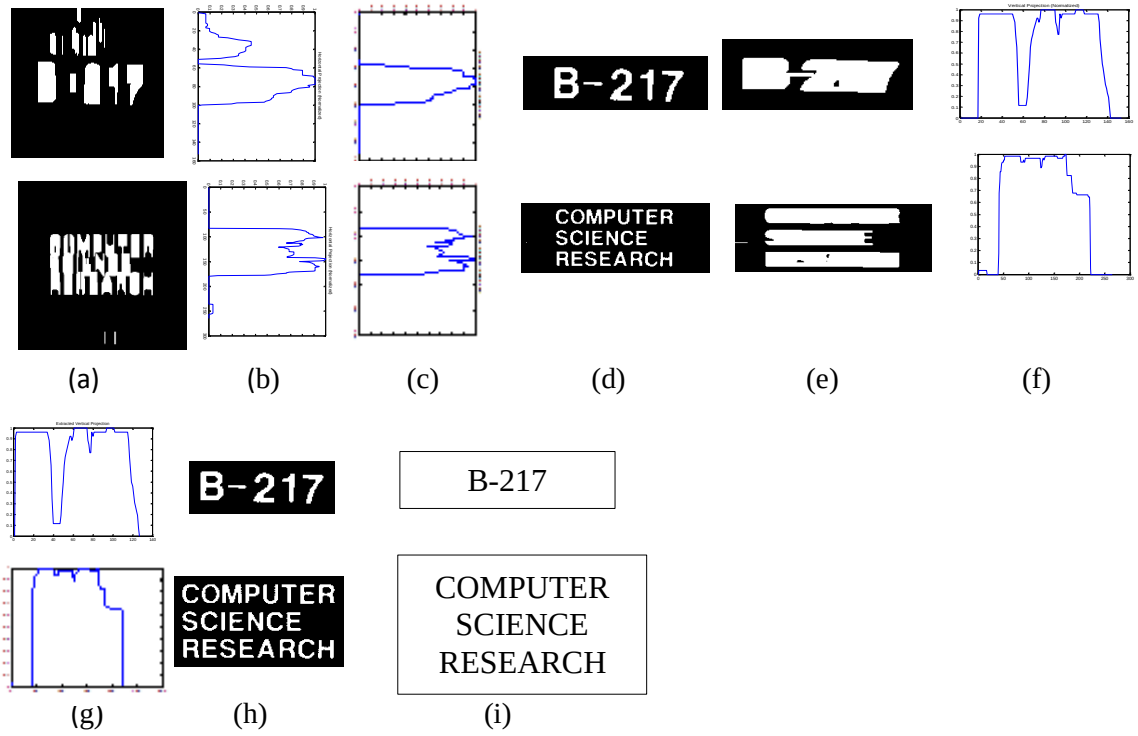


Figure 4.3: The pre-processing and OCR results (a) Dilation vertically, (b) Horizontal projection, (c) Extracted projection, (d) Extracted image horizontally, (e) Dilation horizontally, (f) Vertical projection, (g) Extracted projection, (h) Text of interest, (i) Output of OCR.

Figure 4.3 is continuing the steps in Figures 4.2 and 4.1 to explain the results of the whole system. Part (a) in Figure 4.3 shows the dilation operation which elongated the regions of the binary image in Figure 4.2 (c) vertically to combine the lines together. Obtaining the horizontal projection is shown in part (b) of the same figure. Then, part (c) illustrates the cropping of the middle section of the projection where the text is and the neglecting of the upper and the lower parts of the projection. The results of cropping the binary image horizontally by utilizing the horizontal project in part (c) is shown in part (d). To combine the letters together, part (e) shows the results of dilating the image in

part (d). Vertical projections of images in part (e) are obtained and shown in part (f). Part (g) shows the results of the indicator that is used to get only the middle section of the vertical projection and neglecting the right and the left sides of it. Finally, part (h) shows the results of the preprocessing step which is the text-only images. However, the encoded text results are shown in part (i) which is the outputs of the OCR to be converted into audio for the user.

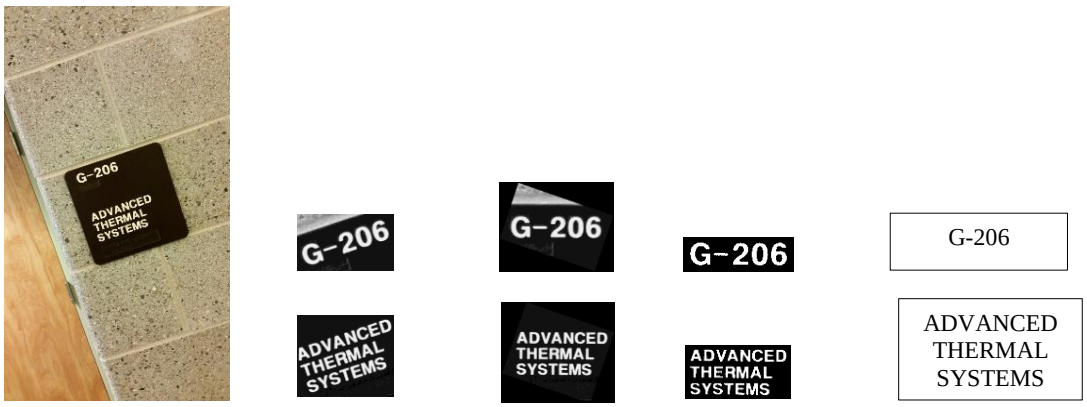
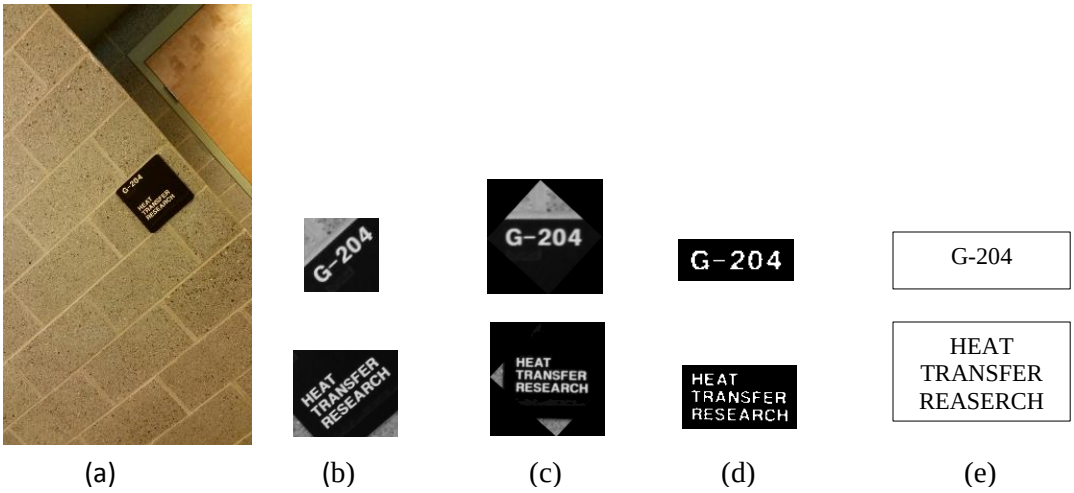


Figure 4.4: Example of the proposed system of close distance image and positive angle (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

Figures 4.4- 4.8 show examples of tested images with different text orientations and sizes. These examples show the results of four stages of our system which are text extraction, text orientation correction, preprocessing, and OCR outputs.

Figure 4.4 shows the results of an image was captured from short distance from the signage where the text is tilted with positive angle from the horizontal axis. The output of the system is the spoken words in part (e).



(a) (b) (c) (d) (e)

Figure 4.5: Example of the proposed system of far distance image and positive angle (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

Figure 4.5 shows an example of an image was captured from a far distance from the signage where the text is tilted with a positive angle from the horizontal axis. Part (b) shows the cropped text from the gray image. The result of text orientation correction of images in part (b) is shown in part (c) which consists of the text and non-text regions in the background. Part (d) illustrates the output of the preprocessing stage where the non-text regions are filtered out. The output of the preprocessing step in part (d) are fed to the OCR to result the encoded text as shown in part (e) to output it as an audio for the visually impaired.

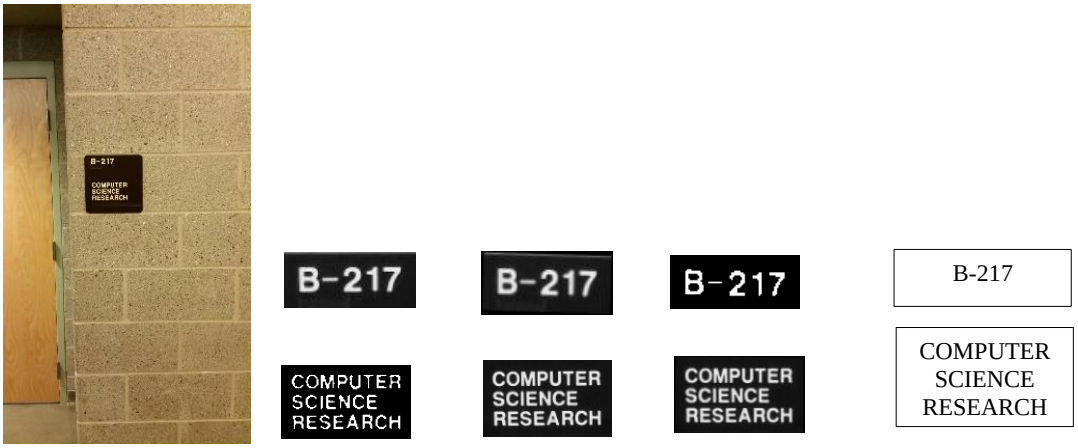


Figure 4.6: Example of the proposed system of far distance image with straight sign (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

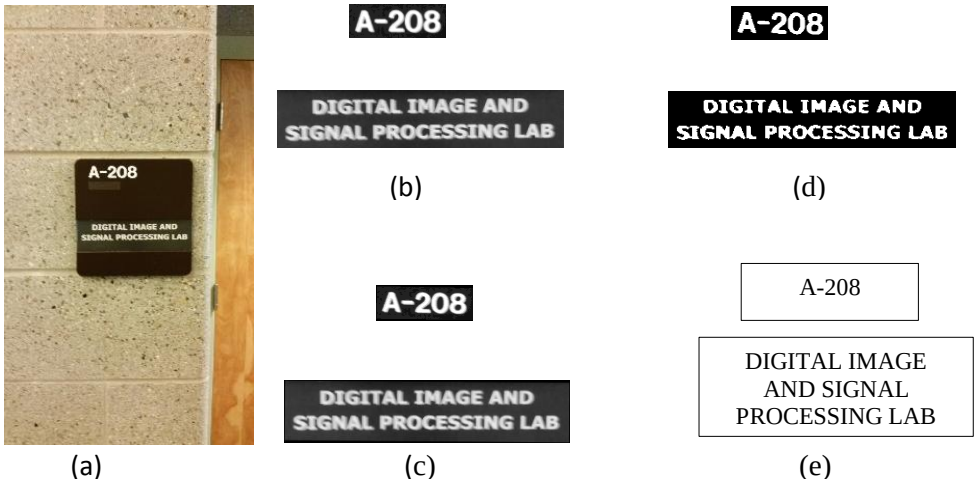


Figure 4.7: Example of the proposed system of close distance image with straight sign (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

Figures 4.6 and 4.7 show examples of images with horizontal text to test the system on both tilted and non-tilted text images. The tested images were captured from

close and far distances from the signage. The system gives right encoded text as shown in part (e) of each figure.

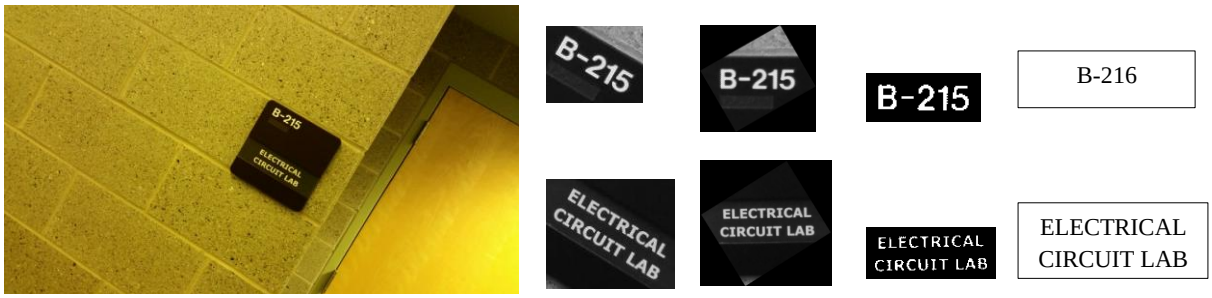


Figure 4.8: Example of the proposed system of close distance image with negative angle (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

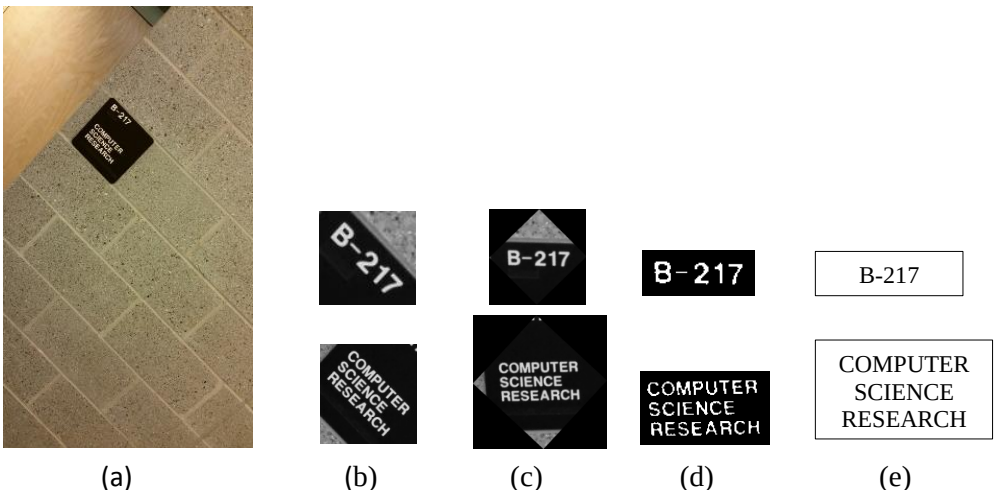


Figure 4.9: Example of the proposed system of far distance image with negative angle (a) original image, (b) Extracted text images, (c) Corrected text orientation, (d) Results of the preprocessing, (d) Encoded text machine from the OCR.

Two more examples of the tested images are shown in Figures 4.7 and 4.8. These two examples illustrate the results of four stages of our system, namely 1) text extraction, 2) text orientation correction, 3) preprocessing, and 4) the OCR outputs. The input images were captured from varying distances, where the close distance is less than one meter and the far distance is less than 3 meters from the signage, and the text is tilted with negative angles from the horizontal axis. The results of the system are shown in part (e) of each figure. The system outputs correct text.

CHAPTER 5

THESIS SUMMARY, CONCLUSIONS, AND FUTURE WORK

5.1 Summary

The work presented in this thesis proposed, developed, and tested a system that is used to assist visually impaired and blind individuals by allowing them access to indoor signage text. The system was implemented using MATLAB and tested using indoor images acquired at constant distance and text size and orientation. It was found that a range of image acquisitions up to 3 meters in maximum distance is acceptable as the system adapted to such variations. The system consists of six components. First, it utilizes an efficient algorithm to detect text from images. The algorithm initiates the text extraction using Maximally Stable Extremal Regions (MSERs). Second, a morphological dilation operation is used to elongate the text as objects in preparation for the following step which needs objects to be identified. Third, detection of object orientation was achieved according to the geometrical properties of the elongated shape, such as area, center, and orientation, of each group of words (object). This realignment allows the system to maximize accuracy at the Optical Character Recognition (OCR) engine output. Fourth, any text orientation detected in the previous step is corrected via the affine transformation. Fifth,, final processing steps are included to filter any non-text regions in order to generate text-only images. This step consists of thresholding gray corrected text images, dilating vertically and horizontally, and finally projecting horizontally and

vertically searching for discontinuities. Sixth, in the last stage, the system uses an open source OCR engine and a Speech Developer Kit (SDK) to convert the encoded-text to speech for audible output. In real life implementation/ deployment, this would be accomplished through a headphone/earpiece mounted on the visually impaired user's glasses. The experimental results show that this system can accurately output speech from the text extracted from indoor signs with different text orientations, sizes, and distances up to three meters.

5.2 Conclusions

The proposed system successfully reads text aloud from signs regardless of variations in acquired image size or orientation. It has been tested on 120 images of the indoor signs found within the engineering building at Western Michigan University. The proposed stages, including text orientation detection with associated pre-processing steps, have been shown to successfully enhance the performance of OCR engines in reading text images of signage. The system was able to output 82% of the tested images perfectly into speech. The average time to read text was 0.95s for each word and 3.46s for each image, while it takes 3.25s and 5.52s respectively for word and image to be spoken aloud.

5.3 Future work

Some recommendations for future work may include addressing real time challenges since processing time is important for real life applications. Further recommendations include the need to develop a system that will allow OCR to recognize symbols and icons found in indoor signage that usually do not contain text information

such as bathroom signs, and the expansion of the current system for use in outdoor signage recognition.

BIBLIOGRAPHY

[1] World Health Organization. (2012). Global data on visual impairments 2010. URL: http://www. who. int/blindness/GLOBALDATAFINALforweb. pdf [accessed 2013-02-28][WebCite Cache].

[2] Blindness and Vision Impairment. (2011, February 8). Retrieved from http://www.cdc.gov/healthcommunication/ToolsTemplates/EntertainmentEd/Tips/Blindness.html

[3] Ye, Q., & Doermann, D. (2014). Text detection and recognition in imagery: A survey.

[4] Wang, K., Babenko, B., & Belongie, S. (2011, November). End-to-end scene text recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 1457-1464). IEEE.

[5] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[6] Xiaoqing Liu and Jagath Samarabandu, *An Edge-based text region extraction algorithm for Indoor mobile robot navigation*, Proceedings of the IEEE, July 2005.

[7] Liu, X., & Samarabandu, J. (2006, July). Multiscale edge-based text extraction from complex images. In *Multimedia and Expo, 2006 IEEE International Conference on* (pp. 1721-1724). IEEE.

[8] Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2963-2970). IEEE.

[9] Chen, H., Tsai, S. S., Schroth, G., Chen, D. M., Grzeszczuk, R., &Girod, B. (2011, September). Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on* (pp. 2609-2612). IEEE.

[10] Neumann, L., & Matas, J. (2012, June). Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 3538-3545). IEEE

[11] Yi, C., Tian, Y., &Arditi, A. (2013). Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons

[12] Rajkumar, N., Anand, M. G., & Barathiraja, N. Portable Camera-Based Product Label Reading For Blind People.

[13] Mladen Kezunovic, *Fundamental of Power System Protection*. Texas, The United Stats of America: Academic Press., 2005. Yi, C., & Tian, Y. (2012). Assistive text reading from complex background for blind persons. In *Camera-Based Document Analysis and Recognition* (pp. 15-28). Springer Berlin Heidelberg.

[14] Wang, S., Yi, C., & Tian, Y. Detecting and Recognizing Signage for Blind Persons to Access Unfamiliar Environments.

[15] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*,*22*(10), 761-767.

[16] A. Vedaldi and B. Fulkerson. Api reference for mser. www.vlfeat.org, September 2005.

[17] Horn, B. (1986). *Robot vision*. MIT press.

[18] Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine vision* (Vol. 5). New York: McGraw-Hill

[19] R. Gonzalez and R. Woods. (1992). *Digital Image Processing*, Addison-Wesley Publishing Company, Chap. 9.

[20] M. Soriano. "Morphological Operations". Applied Physics 186 2013. University of the Philippines

[21] Gonzalez, R. C., & Woods, R. E. (2004). *Digital image processing*. Pearson Education India.

[22] Mori, S., Suen, C. Y., & Yamamoto, K. (1992). Historical review of OCR research and development. *Proceedings of the IEEE, 80*(7), 1029-1058

[23] Tesseract Open-Source OCR: http://code.google.com/p/tesseract-ocr.

[24] Smith, R. (2007, September). An overview of the Tesseract OCR engine. In*icdar* (pp. 629-633). IEEE.

[25] Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition.*International Journal of Recent Technology and Engineering (IJRTE) Volume, 2,* 72-75

[26] Lemmetty, S. (1999). Review of speech synthesis technology. *Helsinki University of Technology*