TEXT LOCALIZATION AND EXTRACTION FROM BACKGROUND WITH TEXTURE AND NOISE IN DIGITAL IMAGES USING ADAPTIVE THRESHOLDING AND CONVOLUTIONAL NEURAL NETWORK

Miss Pukjira Pattaranuprawat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Applied Mathematics and Computational Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019

การระบุและคัดแยกข้อความออกจากพื้นหลังที่มีพรรณลักษณ์และสัญญาณรบกวนในภาพดิจิทัลโดยใช้การขีดแบ่งปรับตัวและโครงข่ายประสาทแบบสังวัฒนาการ

น.ส.ภัคจิรา ภัทรานุประวัติ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Thesis Title	TEXT LOCALIZATION AND EXTRACTION FROM BACKGROUND WITH TEXTURE AND NOISE IN DIGITAL IMAGES USING ADAPTIVE THRESHOLDING AND CONVOLUTIONAL NEURAL NETWORK
By	Miss Pukjira Pattaranuprawat
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Associate Professor RAJALIDA LIPIKORN, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

ภัคจิรา ภัทรานุประวัติ : การระบุและคัดแยกข้อความออกจากพื้นหลังที่มีพรรณลักษณ์และสัญญาณรบกวนในภาพดิจิทัลโดยใช้การขีดแบ่งปรับตัวและโครงข่ายประสาทแบบสังวัฒนาการ. (TEXT LOCALIZATION AND EXTRACTION FROM BACKGROUND WITH TEXTURE AND NOISE IN DIGITAL IMAGES USING ADAPTIVE THRESHOLDING AND CONVOLUTIONAL NEURAL NETWORK) อ.ที่ปรึกษาหลัก : รศ. ดร.รัชลิดา ลิปิกรณ์

ในช่วงไม่กี่ปีที่ผ่านมางานวิจัยเกี่ยวกับการหาตำแหน่งของข้อความได้รับความสนใจเพิ่มมากขึ้นเนื่องจากยังมีปัญหาเกี่ยวกับการหาตำแหน่งข้อความอีกหลายปัญหาที่ยังไม่สามารถแก้ไขได้ ในงานวิจัยนี้นำเสนอวิธีใหม่ในการหาตำแหน่งของข้อความในภาพ ขั้นตอนแรกของวิธีที่นำเสนอคือการปรับปรุงรูปภาพ โดยการแปลงภาพสีให้เป็นภาพระดับสีเทา จากนั้นใช้ตัวกรองค่าเฉลี่ยเพื่อปรับปรุงภาพ โดยตัวกรองค่าเฉลี่ยจะทำให้พื้นหลังกลมกลืมและทำให้สัญญาณรบกวนลดลง หลังจากนั้นจะใช้วิธีขีดแบ่งปรับตัวเพื่อแปลงภาพระดับสีเทาให้เป็นภาพขาวดำ ขั้นตอนที่สองจะเป็นการใช้ส่วนประกอบที่เชื่อมติดกันแบบแปดเพื่อนบ้านเพื่อสร้างกล่องสี่เหลี่ยม เพื่อหาตำแหน่งของข้อความและใช้ขนาดของกล่องข้อความเพื่อจำแนกส่วนประกอบในกล่องสี่เหลี่ยมว่าส่วนประกอบที่น่าจะเป็นข้อความหรือส่วนประกอบที่ไม่ใช่ข้อความ ขั้นตอนที่สามจะใช้อัตราส่วนความกว้างต่อความสูงและวิธีขีดแบ่งแบบออสซุ เพื่อแยกส่วนประกอบที่น่าจะเป็นข้อความให้เป็นส่วนประกอบที่น่าจะเป็นตัวอักษร จากนั้นใช้โครงข่ายประสาทแบบสังวัตนาการเพื่อทำการจำแนกส่วนประกอบว่าใช่ตัวอักษรหรือไม่ใช่ตัวอักษร ข้อมูลที่ใช้ในการประเมินวิธีที่นำเสนอเป็นข้อมูลที่ได้มาจาก ICDAR 2013 ซึ่งประกอบไปด้วยภาพที่ใช้สำหรับสอนโครงข่าย 229 ภาพและภาพที่ใช้เป็นตัวทดสอบ 233 ภาพ ผลจากการจำแนกแสดงให้เห็นว่าวิธีที่นำเสนอให้ค่าการเรียกคืนเท่ากับ 71.87% และค่าเที่ยงเท่ากับ 36.59% จากผลลัพธ์ค่าการเรียกคืนและค่าเที่ยงเท่า งานวิจัยนี้สามารถหากล่องข้อความของตัวอักษรได้อย่างมีประสิทธิภาพแต่ยังไม่สามารถลบกล่องข้อความที่ไม่ใช่ตัวอักษรได้ดีพอ

สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อนิสิต ....
ปีการศึกษา	2562	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6071976923 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORD: text localization, text extraction, adaptive thresholding, CNN, Machine Learning

Pukjira Pattaranuprawat : TEXT LOCALIZATION AND EXTRACTION FROM BACKGROUND WITH TEXTURE AND NOISE IN DIGITAL IMAGES USING ADAPTIVE THRESHOLDING AND CONVOLUTIONAL NEURAL NETWORK. Advisor: Assoc. Prof. RAJALIDA LIPIKORN, Ph.D.

For the past few years, research topics on finding position of text have received more and more attention from researchers because there are still lots of problems that are needed to be solven. We propose a novel method to find the position of text in an image. The first step of the proposed method is to adjust an image by converting a color image to a gray scale image and then use an average filter to improve an image. An average filter is used to make the background smooth and reduce noise. After that an adaptive thresholding is used to convert a gray scale image to a binary image. In the second step, the 8-neighbor connected component is used to generate bounding boxes to find the position of text candidates and the size of a bounding box is used to classify a bounding box as a text candidate or a non-text candidate. In the third step, the width-to-height ratio and Ostu's thresholding are used to divide text candidate into character candidates. Then convolutional neural network is used to categorize a character candidate as character or non-character. The proposed method was evaluated on a data set from ICDAR 2013 which contains 229 training image data and 233 test image data. The classification revealed that the proposed method yields 71.87% of recall and 36.59% of precision. This research can find bounding box of character effectively but it can not delete bounding box of non-character well.

Field of Study:	Applied Mathematics and Computational Science	Student's Signature ....
Academic Year:	2019	Advisor's Signature ....

ACKNOWLEDGEMENTS

For my thesis, I would like to express my gratitude for my Master degree thesis advisor, Associate Professor Rajalida Lipikorn, for her advice about knowledgement including image processing, codes, English language and others. In addition, I would like to thank her for her support and useful suggestion when I have a problem about my thesis or other things that affect to my thesis progression. My thesis will not be success without her support.

I also would like to thank you Assistant Professor Dr. Khamron Mekchay, Associate Professor Dr. Nagul Cooharojananone, Assistant Professor Dr. Kitiporn Plaimas and Dr. Suriya Natsupakpong for helping suggestion about thesis.

Moreover I would like to thank to my scholarship, Science Achievement Scholarship of Thailand (SAST), for the financial support for this Master degree. Further more, my friend whose always help me with the thesis.

Finally, I would like to thank my family for any supports and understanding me.

Pukjira Pattaranuprawat

TABLE OF CONTENTS

Page

.... iii

ABSTRACT (THAI) iii

.... iv

ABSTRACT (ENGLISH) iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

TABLE OF FIGURES viii

LIST OF TABLE xi

CHAPTER I Introduction 1

1.1 Objectives 5

1.2 Scopes and assumptions 5

CHAPTER II Background Knowledge 6

2.1 Background knowledge 6

2.1.1 RGB color 6

2.1.2 Gray scale 7

2.1.3 Average filter 8

2.1.4 Binary image 9

2.1.5 Ostu thresholding 9

2.1.6 Adaptive thresholding 10

2.1.7 Connected component labeling 11

2.1.8 Erosion 13

2.1.9 Convolutional neural network 13

CHAPTER III Proposed Method 17

3.1 Adjusting image 18

3.2 Text candidate position and classification 20

3.3 Character candidate position and classification 20

CHAPTER IV Results and Discussion 23

5.1 Future work 32

REFERENCES 33

VITA 35

TABLE OF FIGURES

Page

Figure 1.1 (a) Different sizes, (b) distortion of texts, (c) different sizes, (d) special font, (e) contiguous characters....2
Figure 1.2 (a) Blurred image, (b) blurred image, (c) low illumination image, (d) high illumination image 3
Figure 2.1 RGB color space....6
Figure 2.2 Gray scale 7
Figure 2.3 RGB and gray scale images from ICDAR 2013 8
Figure 2.4 (a) An image that is superimposed by a 3x3 average filter, (b) pixels of an image under an average filter centered at (x,y), (c) an average filter 9
Figure 2.5 Global thresholding 11
Figure 2.6 Adaptive thresholding using local mean 11
Figure 2.7 (a) The 4-neighbor, (b) the 8-neighbor connected components.... 12
Figure 2.8 (a) 4-neighbor, (b) 8-neighbor connected components..... 12
Figure 2.9 (a) An input image, (b) a structuring element.... 13
Figure 2.10 Output from erosion.... 13
Figure 2.11 Structure of CNN.... 14
Figure 2.12 (a) Input image, (b) sliding window (filter), (c) result 14
Figure 2.13 Filtering that uses stride = 1 15
Figure 2.14 Padding of an image 15
Figure 2.15 An image after using max pooling with 2x2 window and stride = 2 16
Figure 2.16 An image after using mean pooling with 2x2 window and stride = 2.... 16

Figure 3.1 The process of the proposed method 17

Figure 3.2 (a) RGB image, (b) grayscale image, (c) result after adjusting by average filter 18

Figure 3.3 (a) Gray scale image, (b) binary image, (c) negative of binary image 19

Figure 3.4 A structuring element of size 5x2 19

Figure 3.5 (a) The result after using 8 neighbor connected component to find bounding boxes, (b) the result after deleting bounding boxes that have area smaller than 250 pixels. 20

Figure 3.6 Bounding boxes of character candidates. 21

Figure 3.7 The design of AlexNet 22

Figure 4.1 Sample images from ICDAR 2013 training set 23

Figure 4.2 Sample images from ICDAR 2013 test set 24

Figure 4.3 Character train image from ICDAR 2013 train set 24

Figure 4.4 Non-character training images from ICDAR 2013 train set 24

Figure 4.5 The accuracy of training data and the loss of training data 25

Figure 4.6. The results of text that can be localized and extracted 100%. 26

Figure 4.7 The results of text localization with different text sizes 27

Figure 4.8 The result of occluded text localization 27

Figure 4.9 The results of text localization with special characters (a) red special characters (b) white special characters 28

Figure 4.10 The result of text localization with blurred text 28

Figure 4.11 The result of low illunmination 29

Figure 4.12 The result of high illunmination 29

Figure 4.13 The result of text that cannot localize and extract character (a) RGB image, (b) gray scale image 30

Figure 5.1 The results from the proposed method..31

Figure 5.2 Samples of training data that look similar (a) character, (b) non-character. 32

LIST OF TABLE

Page

Table 1 show experimental of ICDAR 2013 26

CHAPTER I
Introduction

At present, social media is one of the important thing for people life. When people want to know some data, they can find data from social media. Normally data that people find from social media contain 2 main part, texts and images. Images on social media come from people when they go to some places or do somethings and upload them on social media. The collection of these images becomes a large repository that is very helpful when people want to search for information from an image. The easiest way to retrieve information from an image is to localize and extract texts from an image. Furthermore, text localization can be used in many other applications, for example, a translator application that can translate text from one language to other languages by using a camera to take a picture of text, localizing text in an image, and localize and classify each character from text. Recently, many researches on text localization and classification have been proposed but there are still problems with the accuracy of results. The problems of text localization and extraction can be divided into two cases.

1. The problems on text localization due to text properties

There are many problems caused by the properties of text. The first problem is the differences of fonts, which includes the size and the typeface because each character has many different shapes, styles, and sizes. Furthermore, some characters have shape that are similar to other characters or background. The second problem is the distortion of text. The third problem is the sizes of characters. If a character's size is too small, it will be very difficult to localize. The last problem is the contiguous characters which it is difficult to identify and separate these characters. Figure 1.1 shows example images with problems from text properties.

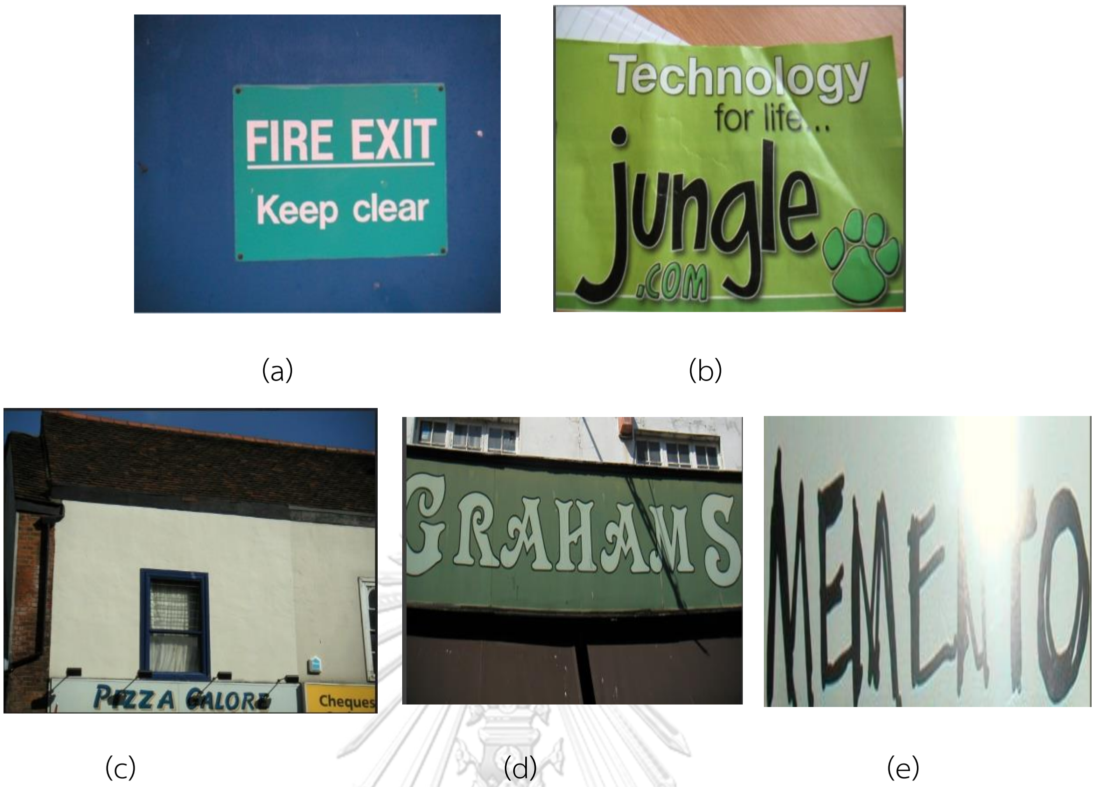


Figure 1.1 (a) Different sizes, (b) different sizes, (c) distortion of texts, (d) special font, (e) contiguous characters

2. The problems on text localization due to external factors

There are many problems due to external factors. The first problem is a motion-blurred image. Sometimes, people take a picture while they are moving or the camera quality is poor. The second problem is the illumination. When people take a picture in low or high illumination, it makes texts in a picture to be obscured and difficult to see. Figure 1.2 shows example images with problems from external factors.

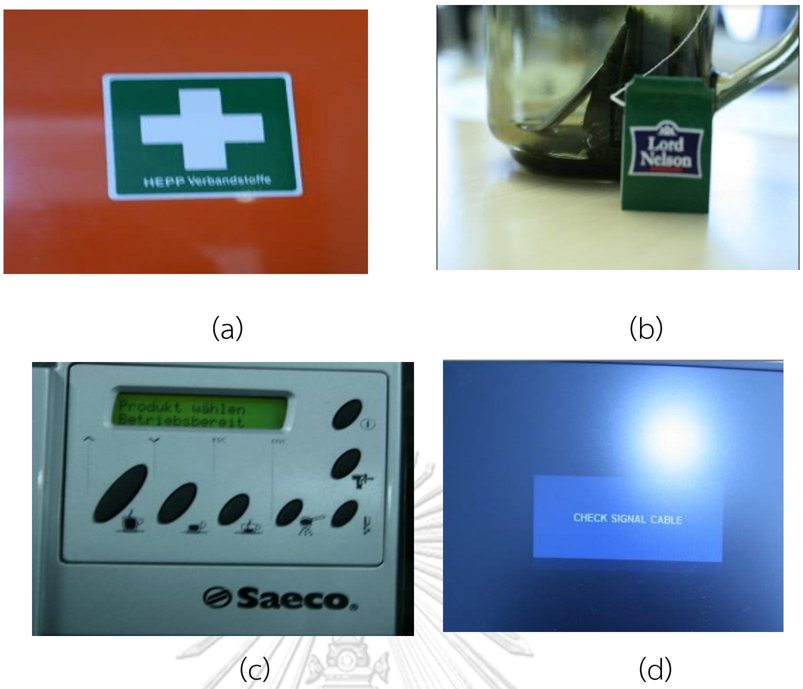


Figure 1.2 (a) Blurred image, (b) blurred image,

(c) low illumination image, (d) high illumination image

From the above-mentioned problems, There are many research about text localization and classification but the problem about text localization and classification still challenging . USTB_TexStar [1] Is the method that invented by Xu-Cheng Yin. The first step is extracting characters by using maximally stable extremal region (MSER) to make non-characters decrease. The next step is to group character by calculating distance between 2 objects. The smallest distance will be chosen to group 2 objects into the same group. After that this method classifies text or non-text by using width, height, ratio of width and height, etc. The last step is to classify text and non-text by using Adaboost. Adaboost is method that change weak model to strong model. Adaboost will start with average weight and find error of model. After that this method will adjust weight to find the best model. This method can make recall 68.26% from ICDAR 2011 data set. In 2013, there are competitions about local text. USTB_TexStar can be localize 66.45% [2] on ICDAR 2013 data set. Kakade Snehal [3] proposed a novel method to localize and recognize text. The first step is decreased noise by using median filter and improve edge of image by using canny edge. Canny edge will make edge sharper. Next threshold is used to change gray scale to binary image. Then 5 features contain Horizontal Crossing, Perimeter, Area, Euler number, and Bounding boxes used to separate text or non text. After that this research use 3 method to classify text or non text contain SVM, Adaboost and Ostu were used to recognize text or non text. The last, this method will separate text bounding box to character bounding box by using horizontal projection and vertical projection. The result after used SVM, Adaboost, Ostu are 78.80%, 75.04%, 64.40%. However this research did not used all image from ICDAR 2013 and SVT 2010 data set. In this research used data set around 100 images. Boris Epshtein et al. [4] proposed text detection by using the width from bound of character to other side. Direction to calculate width come from gradient of pixel. Next this method group pixel by using stroke width. After that used features size of component, aspect ratio to eliminate non text component. This method yields a recall of 60%. Disadvantage of this method is that it can not handle difficult orientation. Yu Zhou [5] adapted Epshtein work. This technique will find width of character edge. Then this technique used histogram of edge to find performance width. Performance width will used to generate superpixel and find the edge of superpixel. Huizhong Chen [6] using MSER to find the interest region and enhance edge by using canny edge. Then this technique delete too small or big objects. Next this technique find width of edge character was found by an adaptive method of Epshtein. After that we group characters by using width of edge and height. This method can find recall 60%. Alessandro Bisacco et al [7] proposed method to extract text from smartphone but focus on only extracting text in horizontal line. This technique adapt machine learning to improve classifier. Hyung Il Koo et al. [8] proposed text detection by using machine learning. This technique starts from finding interest region using MSER and then uses adaboost to learn component connect or not connect. After that this method groups components into the same group. The last, they classify text or non text by using multilayer perceptron.

In this work, we propose an alternative method to localize text by using adaptive thresholding, aspect ratio and Convolutional Neural Network. This thesis is organized as follows background knowledge, proposed method, results and discussion, and conclusion. Background Knowledge contain 9 parts RGB color, gray scale, average filter, binary image, ostu thresholding, adaptive threshold, connected component labeling, erosion and convolutional neural network. Proposed method contain 3 parts adjusting image, text candidate position and classification, and character candidate position and classification.

1.1 Objectives

To design and propose a method to localize and extract text from the background with complicated texture and noise in digital images

1.2 Scopes and assumptions

1. Data set contains RGB images with extension jpg, bmp, and tif.
2. The proposed method can handle text on the background with complicated texture and noise that have different shades of color from text.
3. Overlapping of texts is not included in the scope of the proposed method.
4. Text message should be readable by a human.

CHAPTER II

Background Knowledge

In order to perform text localization and extraction from background with texture and noise, various image processings and other related techniques are used as guidelines.

2.1 Background knowledge

2.1.1 RGB color

The RGB color space is a color space which consists of three primary colors: red, green, and blue. These colors are mixed together to form various colors. When these three primary colors are mixed at the right proportion, they form the secondary colors: yellow, cyan, and magenta. Yellow color contains red color and green colors. Cyan color contains blue and green colors. Magenta color contains red and blue colors. White color contains red, green, and blue colors. In a full-color image, the range of each color starts from 0 to 1. Black color has the values of red, green, and blue as (0,0,0); whereas, white color has the values of red, green, and blue as (1,1,1). Figure 2.1 shows the RGB color space in three dimensions.

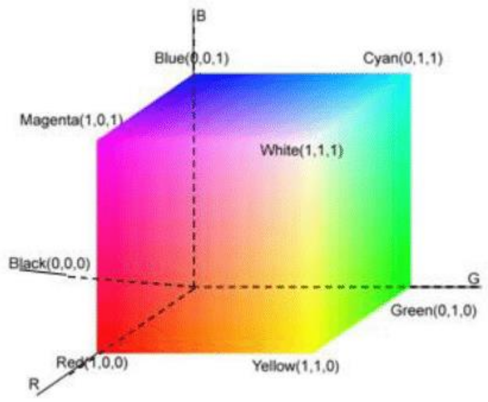


Figure 2.1 RGB color space

Source : https://www.baslerweb.com/en/sales-support/knowledge-base/frequently-asked-questions/what-is-the-rgb-color-space/15179/

RGB color space has many advantages for electronic devices such as television, computer, and others because these devices use color to represent and display an image.

2.1.2 Gray scale

In the field of image processing, grayscale images are typically displayed from pure black color to pure white color. Grayscale images are different from black and white images. Black and white images consist of only two colors, black and white; whereas, grayscale images have many gray levels between black and white. The levels of gray color depends on the number of bits that are used to store gray value of each pixel. For example, if we have an 8-bit grayscale image then we can use 256 level of gray colors to display an image. Figure 2.2 shows the shading of gray levels from black to white

Figure 2.2 Gray scale

Source: http ://www.columbia.edu/itc/visualarts/r4110/f2000/week06/06_03_Color_palettes.pdf

We can usually use mathematical methods to convert an RGB color image to a grayscale one. One of the methods is to compute the level of gray scale from the average value of red, green, and blue colors as expressed in Eq. (2.1) as

$$Gr = \frac{R+G+B}{3} \tag{2.1}$$

where Gr, R, G and B are The levels of gray-scale level, red color, green color, and blue color.

Moreover, another method to compute the level of gray scale from RGB color is expressed in Eq. (2.2). This method calculated by using parameters of digital television .Figure 2.3 shows a color image that is converted to a grayscale image using Eq. (2.2).

$$Gr=0.2989R+0.5870G+0.1140B \tag{2.2}$$

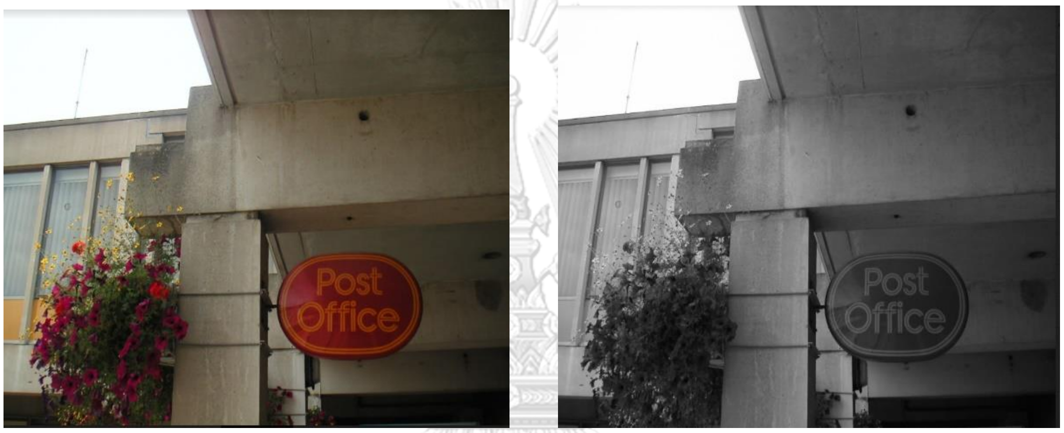


Figure 2.3 RGB and gray scale images from ICDAR 2013

2.1.3 Average filter

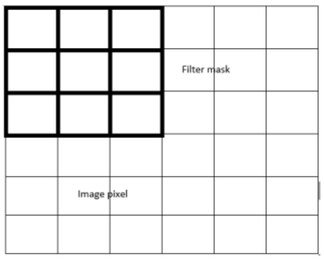
The average filter is a filter of linear class which is used to compute the average intensity of an image from the pixels that are superimposed by a filter as the new intensity value to be used for the pixel at the center of a filter. This filter can be used to smooth an image, delete noise, and blur an image. A filter with smaller size can make an image to be less blurred and less noise can be removed; whereas, a filter with larger size can make an image to be more blurred and more noise can be removed. An image can be enhanced by multiplying the original image with the average filter as expressed in Eqs. (2.3)-(2.5).

$$R(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} F(s,t) I(x+s, y+t) \tag{2.3}$$

$$a = (m-1)/2 \tag{2.4}$$

$$b = (n-1)/2 \tag{2.5}$$

where $R(x,y)$ is the enhanced image, $I(x,y)$ is an input image, $F(x,y)$ is an average filter that the value of average filter equal $\frac{1}{size\ of\ average\ filter}$, m is the height of a filter and n is the width of a filter. Normally size of filter is odd number because It is symmetric and easy to calculate. Figure 2.4 shows an average filter of size 3x3



(a)

I(x-1,y-1)	I(x-1,y)	I(x-1,y+1)
I(x,y-1)	I(x,y)	I(x,y+1)
I(x+1,y-1)	I(x+1,y)	I(x+1,y+1)

(b)

F(-1,-1)	F(-1,0)	F(-1,1)
F(0,-1)	F(0,0)	F(0,1)
F(1,-1)	F(1,0)	F(1,1)

(c)

Figure 2.4 (a) An image that is superimposed by a 3x3 average filter, (b) pixels of an image under an average filter centered at (x,y), (c) an average filter

2.1.4 Binary image

A binary image is an image with only two possible intensity values for each pixel. Numerically, the two values are 0 for black and 1 for white. Binary images can be generated from a grayscale by thresholding the intensity values. In order to obtain a binary image, any pixel whose intensity is greater than or equal to the threshold value is set to 1, otherwise 0. In most cases, 1 represents foreground color and 0 represents background color.

2.1.5 Ostu thresholding

Ostu thresholding is a method that is used to convert a grayscale image to a binary image. This method will find threshold by minimizes the within-class variance or maximizes the between-class variance. In this paper, the between-class variance is calculated by Eqs. (2.6)-(2.10)

$$\sigma^2_{Be} = W_b W_f (\mu_b - \mu_f) \tag{2.6}$$

$$W_b = \sum_{i=0}^{t-1} p_i \tag{2.7}$$

$$W_f = \sum_{i=t}^{L-1} p_i \tag{2.8}$$

$$\mu_b = \frac{\sum_{i=0}^{t-1} i p_i}{N_b} \tag{2.9}$$

$$\mu_f = \frac{\sum_{i=t}^{L-1} i p_i}{N_f} \tag{2.10}$$

where σ^2_{Be} is the between-class variance, W_b is the weight of the background, W_f is the weight of the foreground, μ_b is the mean of the background, μ_f is the mean of the foreground, i is gray scale level, t is a threshold value that is used to separate foreground and background, p_i is the probability of pixel that has gray scale value i, L is the number of gray scale levels, N_b is the number of background pixels, N_f is number of foreground pixels.

2.1.6 Adaptive thresholding

Thresholding can be used to extract the objects from an image by setting all pixels with intensity values above a threshold value to the value considered as the foreground. At the same time, all the remaining pixels will be set to the background value. However, when it comes to the situation when an object obtains different levels of illumination such as high illumination and low illumination in one image, thresholding might not be able to extract an object correctly because some parts of an object may have high intensities; whereas, other parts may have low intensities. In this case, an adaptive thresholding is more suitable. An adaptive thresholding considers intensities of a local region in an image by dividing the original image into smaller regions and use a threshold value that is suitable for each region, to reduce the influence of illuminations.

The threshold value can be set as the mean or the median of intensities within each region. The adaptive threshold can give a good result even in an environment with uneven illuminations. Figure 2.5 shows the result of using (non-adaptive) thresholding while Figure. 2.6 shows the result of using adaptive thresholding. It can be seen that adaptive thresholding yields a better result.

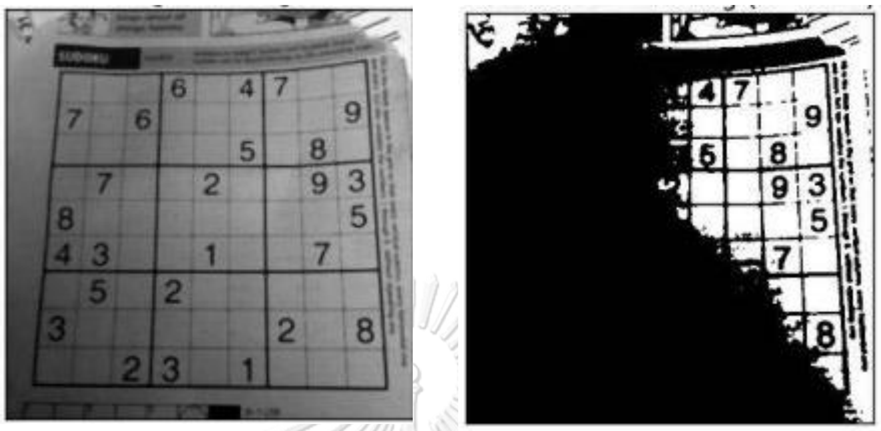


Figure 2.5 Global thresholding

Source: https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html

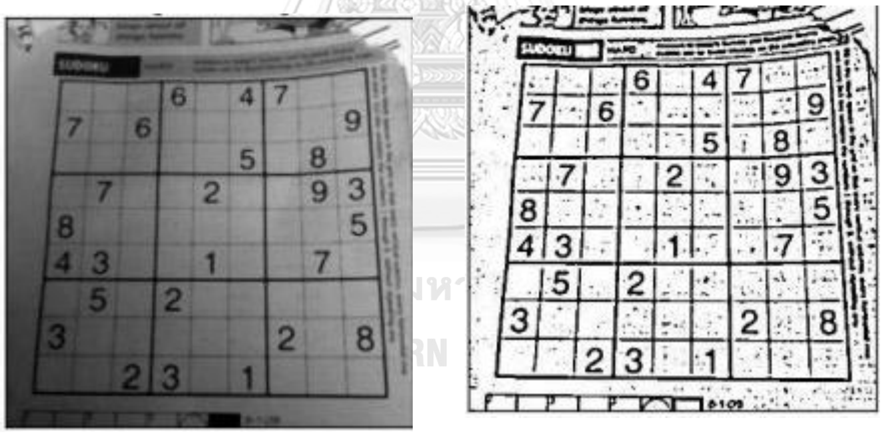


Figure 2.6 Adaptive thresholding using local mean

Source: https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html

2.1.7 Connected component labeling

Connected component labeling is used to find an object in an image by considering connected pixels. There are two types of connected components 4-neighbor connected component and 8-neighbor connected components. The 4-neighbor connected component uses four neighbor pixels (lower, upper, left, and right) around the center pixel. The 8-neighbor connected component uses eight neighbor pixels from all eight directions. Figure 2.7 shows the 4-neighbor and 8-neighbor connected components.

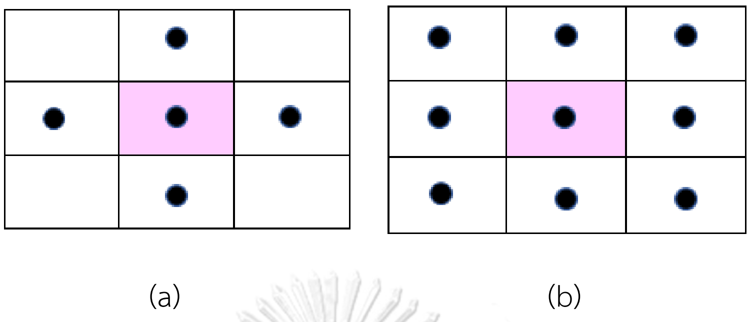


Figure 2.7 (a) The 4-neighbor, (b) the 8-neighbor connected components

Let I represent a binary image with the values $I[x,y]$ at the pixel $[x,y]$ is either 0 or 1. A pixel $[x,y]$ is connected to a pixel $[x',y']$ if there is a sequence that makes pixel $[x,y]=[x_0,y_0]=[x_1,y_1]=\ldots=[x_n,y_n]=[x',y']$ and $[x_i,y_i]$ is a neighbor of $[x_{i-1},y_{i-1}]$ for $i=,1,\ldots,n$. A connected component means a set of pixels that have the same value and every pair of pixel connected. A connected component labeling is used to label each connected component. Figure 2.8 shows the results from using 4-neighbor connected comopnent and 8-neighbor connected components to label objects in an image. It can be seen that the object “2” and the object “__” are counted as two components when using 4-neighbor and counted as one component when using 8-neighbor.

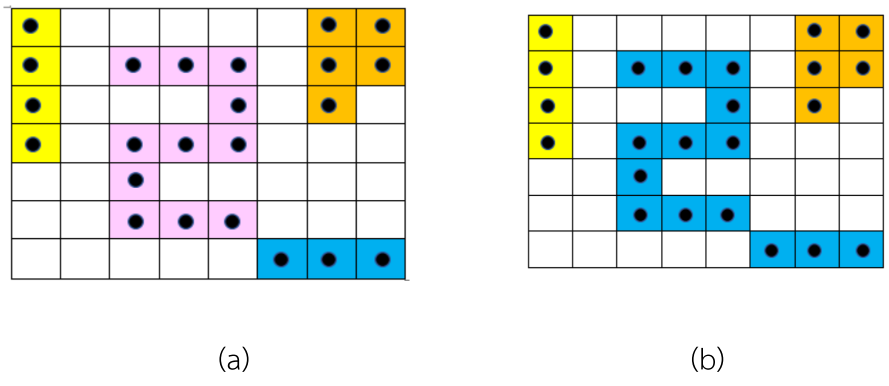


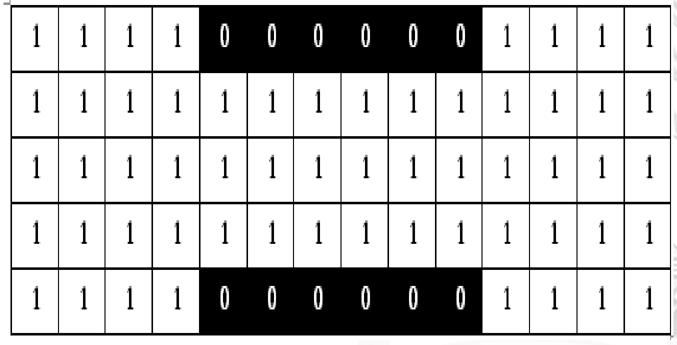
Figure 2.8 (a) 4-neighbor, (b) 8-neighbor connected components.

2.1.8 Erosion

Erosion is a method that makes an object thinner than the original object. The structuring element is used to control the thickness of an object. Erosion is performed by placing the structuring element over a binary image. If all elements in structuring element with values 1 are overlapped with all pixels in the overlapped region of the image values 1, the values of these pixels remain unchanged; otherwise, they are set to 0. Erosion is calculated by using Eq. (2.11).

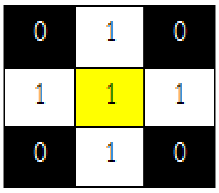
$$A \ominus B = \{z|(B_z) \subseteq A\} \tag{2.11}$$

where A is a binary image, B is a structuring element, z is an element of a structuring element, and B_z is a structuring element. Figure 2.10 shows the result from eroding the original binary image by a 3x3 structuring element in Figure 2.9.



1	1	1	1	0	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	1	1	1	1

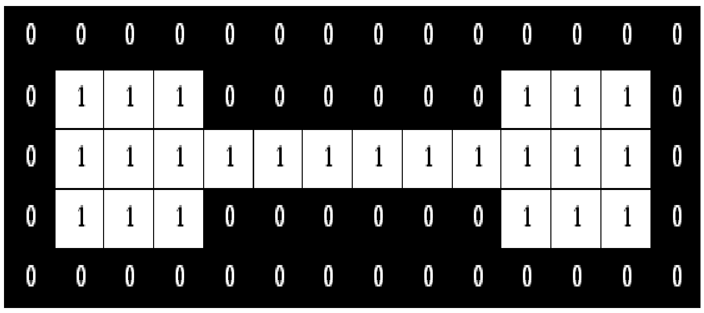
(a)



0	1	0
1	1	1
0	1	0

(b)

Figure 2.9 (a) An input image, (b) a structuring element



0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.10 Output from erosion

2.1.9 Convolutional neural network

Convolutional neural network (CNN) is a neural network that has many structure layer. CNN is used to extract complex feature from data. CNN is often used for rough data such as an image with a lot of noise. CNN structure designed for image data contains two stages: the convolutional and the pooling stages. Figure 2.11 shows the structure of CNN.

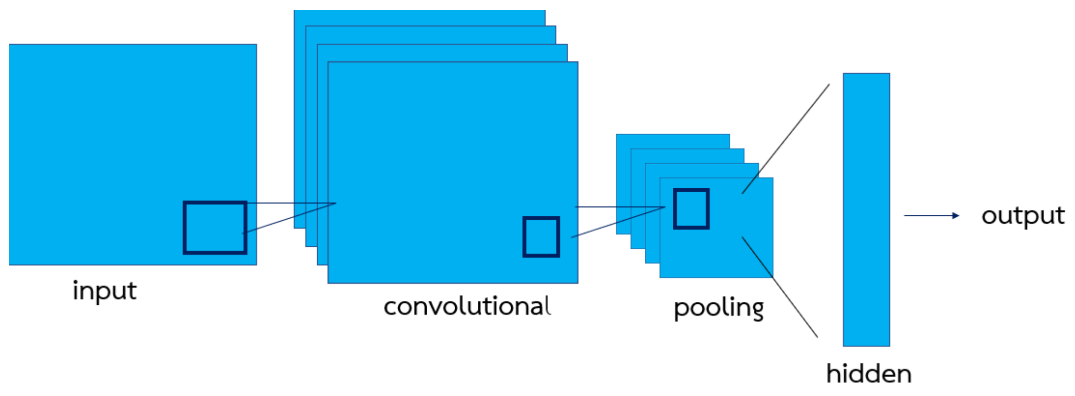


Figure 2.11 Structure of CNN

The convolutional stage makes a sliding window to find the features of an image such as color, edge, and shape. Convolutional stage contains three parts: filter, stride, and padding. The filter is used to find a component of an image. One filter can find only one component at a time and, when training, there are usually many filters to combine. Figure 2.12 shows an example of a filter and the result from filtering.

2	2	4	5	3
6	4	1	5	2
2	1	4	8	3
2	1	1	3	6
1	3	4	5	1

(a)

1	0	0
0	1	0
1	0	0

(b)

0	0	0	0	0
0	8	4	13	0
0	9	9	10	0
0	4	5	11	0
0	0	0	0	0

(c)

Figure 2.12 (a) Input image, (b) sliding window (filter), (c) result

The stride is used to define the number of steps a filter has to move each time on an image. Figure 2.13 shows filtering that uses stride equals to 1.

2	2	4	5	3
6	4	1	5	2
2	1	4	8	3
2	1	1	3	3
1	3	4	5	1

Figure 2.13 Filtering that uses stride = 1

Padding is used to assign the values to the boundary of an image after filtering. We can assign 0 or another value to manage with the boundary of an image. Figure 2.14 shows padding of an image when the value is set to 0.

0	0	0	0	0
0	8	4	13	0
0	9	9	10	0
0	4	5	11	0
0	0	0	0	0

Figure 2.14 Padding of an image

The pooling stage is used for resizing an image having the same detail as the image before resizing. Commonly, there are two types of pooling that are widely used: max pooling and mean pooling. Max pooling is used to find the maximum value in the area that a filter overlaps. Figure 2.15 shows the result after using max pooling with window of size 2x2 and stride equals to 2.

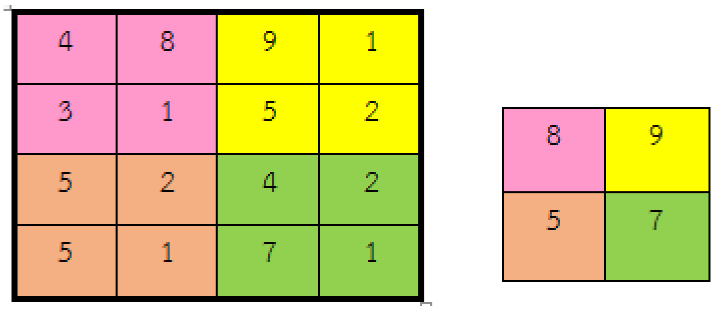


Figure 2.15 An image after using max pooling with 2x2 window and stride = 2

Mean pooling is used to find the mean value in an area that a filter overlaps. Figure 2.16 shows an image after using mean pooling with window of size 2x2 and stride equals to 2.

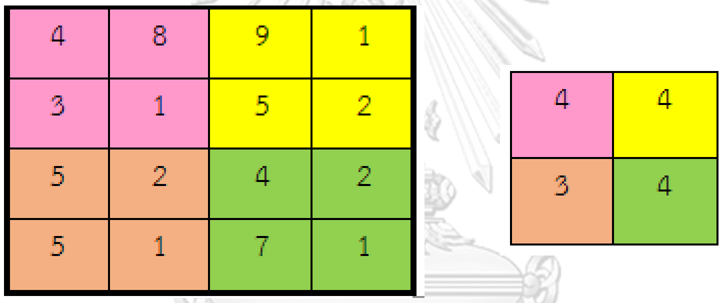


Figure 2.16 An image after using mean pooling with 2x2 window and stride = 2

CHAPTER III

Proposed Method

In this research, we propose an alternative method to find position of text by using adaptive thresholding, aspect ratio and convolutional neural network. The proposed method contains 3 steps. The first step is to adjust an image. The second step is to find the position of text candidate and classify text candidate. The third step is to find the position of character candidate and classify character candidate. Figure 3.1 shows the proposed method.

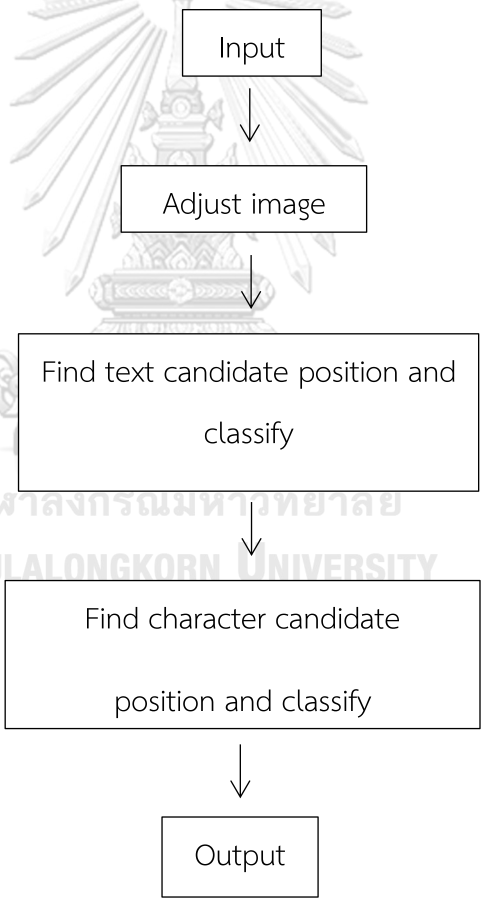
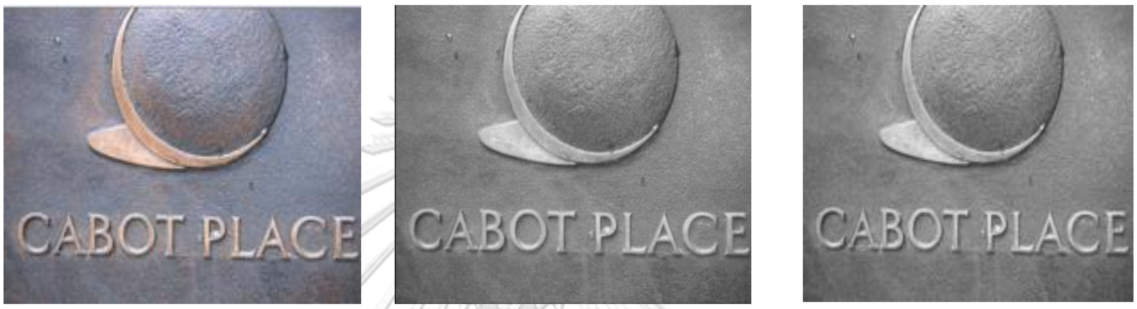

Input
Adjust image
Find text candidate position and classify
Find character candidate position and classify
Output

Figure 3.1 The process of the proposed method

3.1 Adjusting image

Adjusting image contains 4 steps. The first step is to change an RGB image (Figure 3.2 (a)) to a grayscale image (Figure 3.2 (b)). In the second step, average filter is used to reduce noise and particular of image, and remove noise from background. In this research, the average filter of size 3 x 3, 5 x 5, 7x7 and 11 x11 is used but the best result is average filter is size 7x7. Figure 3.2 (c) shows the result from using average filter to remove noise from background.



(a) (b) (c)

Figure 3.2 (a) RGB image, (b) grayscale image, (c) result after adjusting by average filter

The third step, a gray scale image is converted to a binary image using adaptive thresholding. Adaptive thresholding is appropriate for an image that its global brightness is not uniform. By using adaptive thresholding, the threshold value of each window is equal to the average value of the intensities within the window of size m by n. The adaptive thresholding is expressed in Eqs. (3.1), Eq. (3.2) - (3.3) show the setting of the height and the width of a window in this work, while Eq. (3.4) shows the calculation of the threshold value *T(x,y)*.

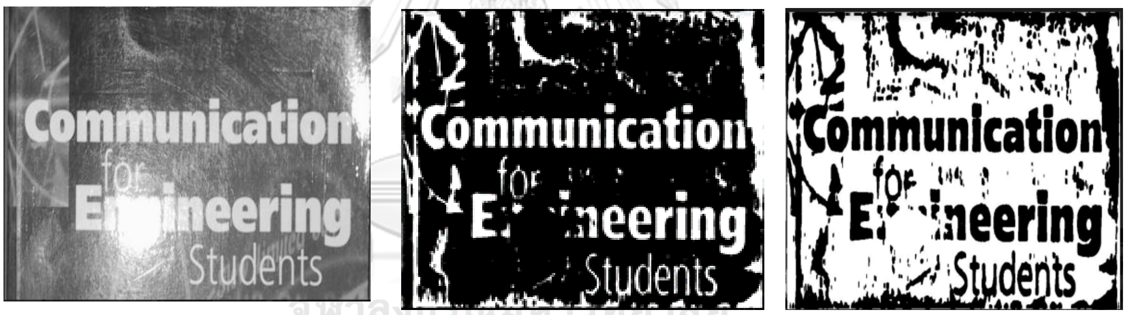
$$g(x,y)=\begin{cases} 0, \text{ if } f(x,y)<T(x,y)-T(x,y)*s \\ 1, \text{ if } f(x,y) \geq T(x,y)-T(x,y)*s \end{cases} \tag{3.1}$$

$$m = 2 * floor(h(f)/16) + 1 \tag{3.2}$$

$$n = 2 * floor(w(f)/16) + 1 \tag{3.3}$$

$$T(x,y)=\frac{1}{mn}\sum_{x,y\in\eta} f(x,y) \tag{3.4}$$

Hence $g(x,y)$ is a binary image, $f(x,y)$ is a gray-scale image, $T(x,y)$ is a threshold value for each window that is calculated by the average value of intensities, s is sensitivity value that affects the chance of a pixel to be white (foreground) or black (background). When the sensitivity is high, the chance of having more foreground pixels will be more than low sensitivity, likewise the low sensitivity will have pixel as background more than high sensitivity, $h(f)$ and $w(f)$ are height and width of an image, η is a set of pixels within each window. After we change from gray scale (Figure 3.3 (a) to binary image (figure 3.3 (b)). We still do not know the color of character is white or black that we must change from black to white and white to black used for finding bounding boxes of text in the next step. Figure 3.3 (c) shows negative of binary image.



(a) (b) (c)

Figure 3.3 (a) Gray scale image, (b) binary image, (c) negative of binary image

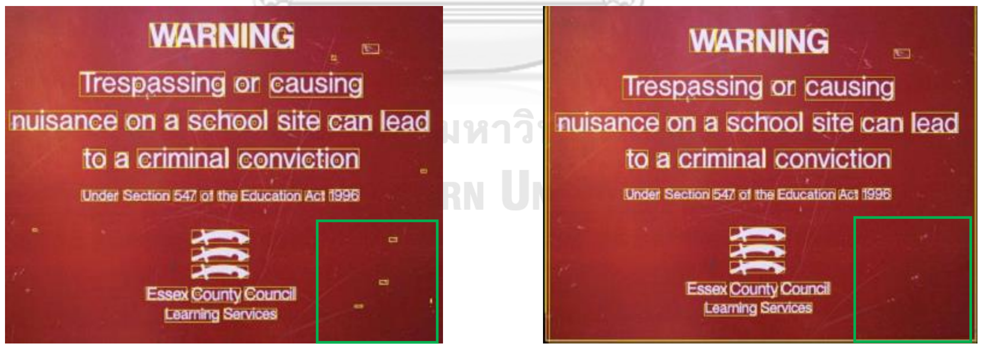
After we change to a binary image, step 4 of adjusting an image is to use erosion to separate connected characters. The reason that we must separate contiguous character because it will easy to localize and classify character. In this research, we use structuring element of size 5x2 because characters are contiguous along horizontal line. Figure 3.4 shows a structuring element of size 5x2.

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Figure 3.4 A structuring element of size 5x2

3.2 Text candidate position and classification

After we obtain a binary image, an 8-neighbor connected component is used to connect pixels in the lower, upper, left, right and diagonal directions. The reason that we used 4-neighbor connected component because many shape of character and text connected in diagonal direction. If a pixel is white or 1 and connected in any of the direction mentioned above, we draw a bounding box around this connected component in the original color image. We classify non-text from bounding box whose area is less than 250 pixels is deleted because this bounding box is too small and has a high chance of being noise or non-text. The reason that we used 250 pixels because we compare the result between 500 pixels, 350 pixels, 250 pixels and 100 pixels. The best value is 250 pixels. Figure 3.5 (a) shows the result after using 8-neighbor connected component to find bounding boxes. Figure 3.5 (b) shows the result after deleting bounding boxes whose area are less than 250 pixels. The green bounding box in Figure 3.5 (a) contains many bounding boxes caused by noise but after deleting noise, these bounding boxes are removed from an image as shown in Figure 3.5(b).



(a) (b)

Figure 3.6 (a) The result after using 8 neighbor connected component to find bounding boxes, (b) the result after deleting bounding boxes that have area smaller than 250 pixels.

3.3 Character candidate position and classification

The last step is to separate text into characters. From the previous step, bounding boxes that have the width that is longer than twice the height have to be separated because we assume that these bounding boxes contain more than one character. These text bounding boxes are extracted from the original image and saved on separate files. Each bounding box image is converted to a binary image by using Ostu thresholding. The reason that used ostu thresholding because almost of text bounding box contain only 2 color text and blackground. The 8-neighbor connected component is used to find bounding boxes in binary image and find the same position of bounding boxes in RGB image. We classify non-text bounding boxes with the area less than 100 or the height less than 8 pixels are deleted. The reason that use area 100 pixel because it is too small and assume be noise.

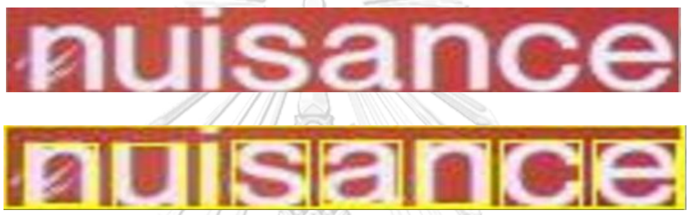


Figure 3.7 Bounding boxes of character candidates.

After removing noise, there are still some bounding boxes of non-character remaining. AlexNet [9] is the Convolutional Neural Network that is used to classify a bounding box whether it is a character or non-character. The reason that we used AlexNet because the result when use to classify object is good. The input data of AlexNet in this work is RGB images with size 227x227. The output that we expect is character bounding box. Training set of AlexNet is RGB image. We crop character and non-character from training set. After that we save into 2 files character and non-character. Figure 3.7 shows the design of AlexNet.

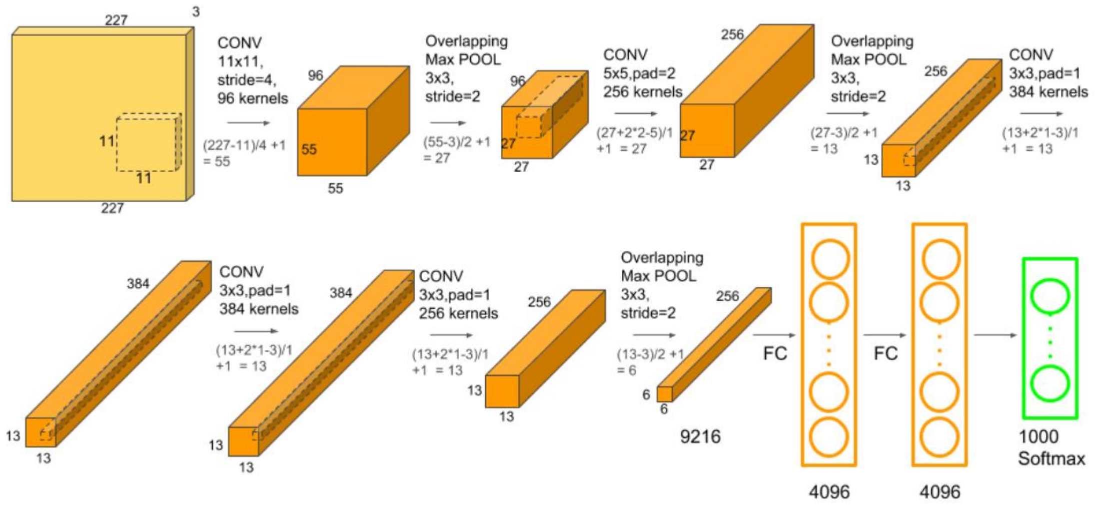


Figure 3.8 The design of AlexNet

Source: https://neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/

CHAPTER IV
Results and Discussion

The data set used in this research is the ICDAR 2013 with 229 images used for training and 233 images for testing. Individual characters and non-characters are cropped by manual from each training image to generate 4446 images. Training set 4446 images contain 3876 from non-character and 570 from character. After that we will resize image to size 227x227. The training characters consist of numbers from 0 to 9, capital letters from A to Z and lower-case letters from a to z. Figure 4.1 shows sample images from ICDAR 2013 training set. Figure 4.2 shows sample images from ICDAR 2013 test set. Figure 4.3 shows samples of character training images and Figure 4.4 shows samples of non-character training images. The first graph of figure 4.5 shows the accuracy of training data and the second graph shows the loss of training data.

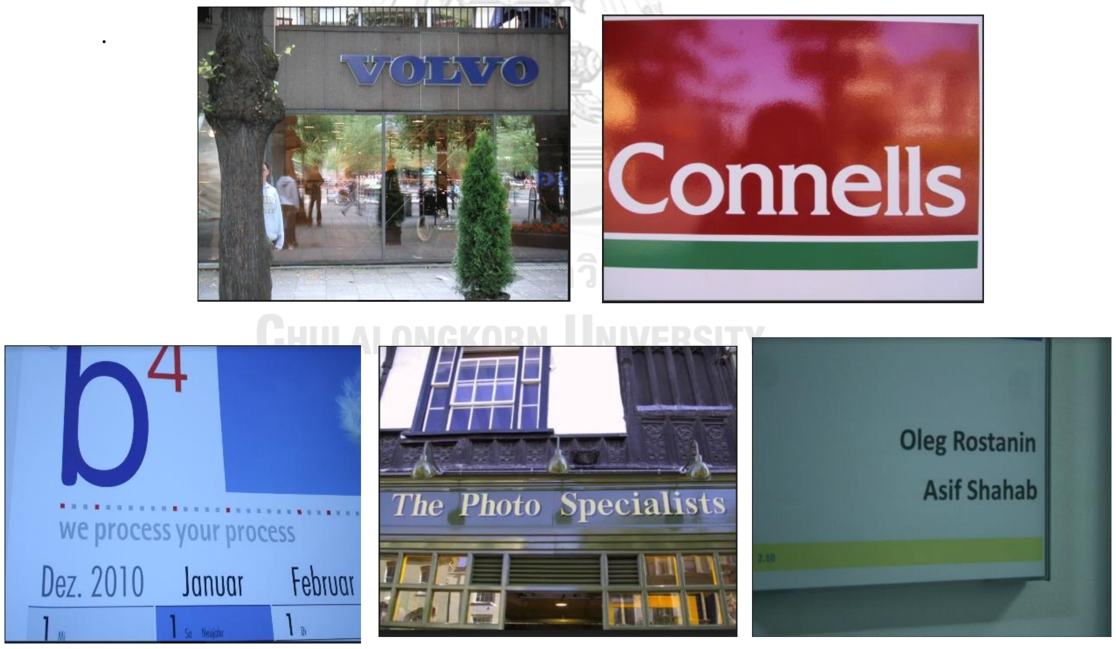


Figure 4.1 Sample images from ICDAR 2013 training set

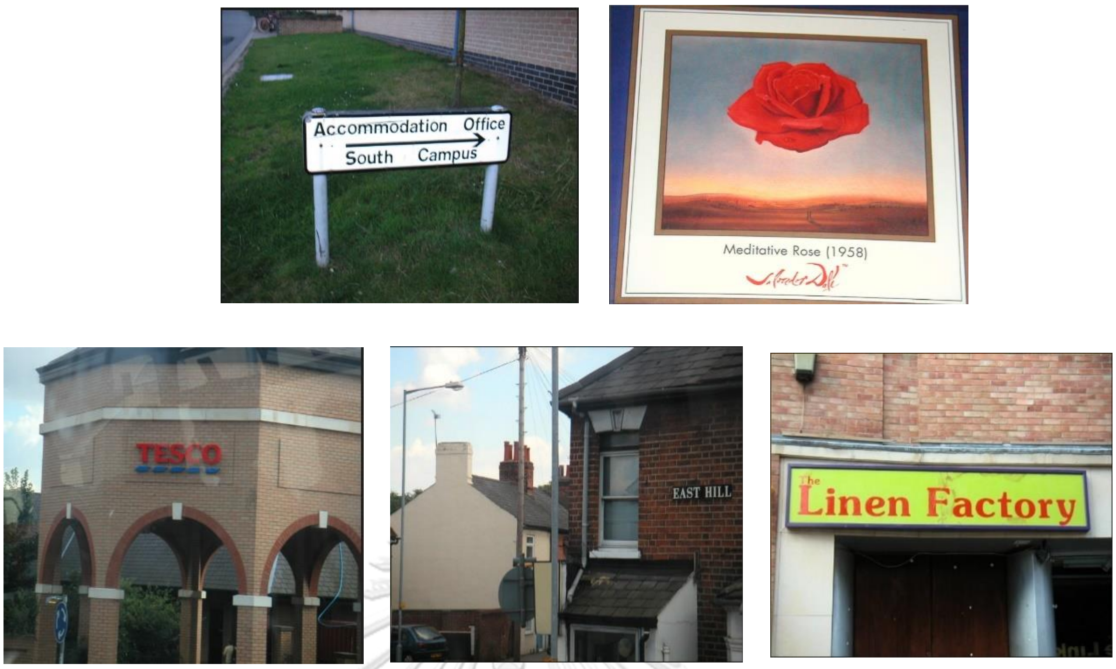


Figure 4.2 Sample images from ICDAR 2013 test set

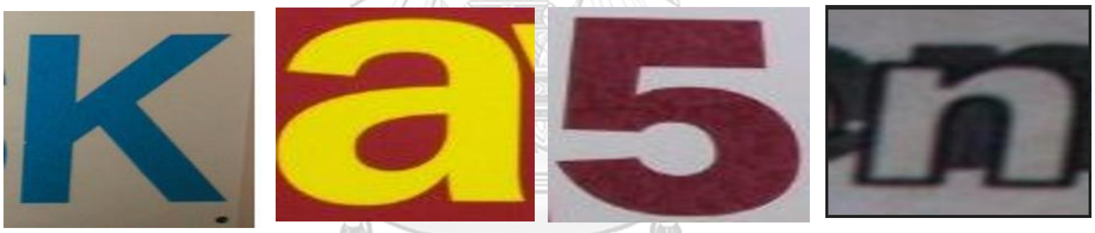


Figure 4.3 Character train image from ICDAR 2013 train set

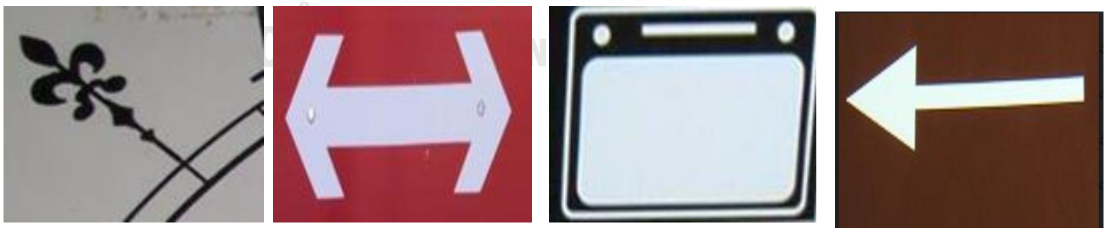


Figure 4.4 Non-character training images from ICDAR 2013 train set

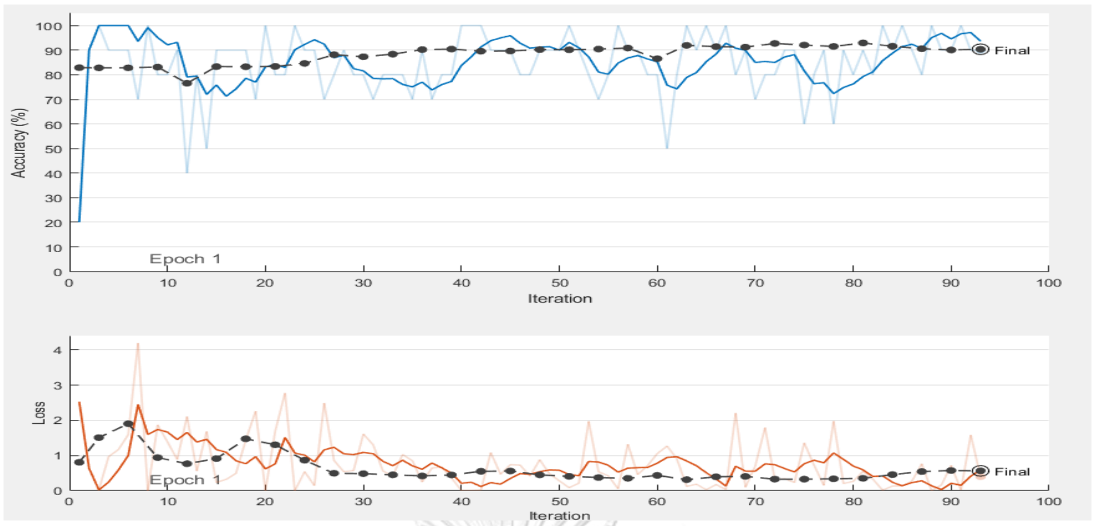


Figure 4.5 The accuracy of training data and the loss of training data

From Figure 4.5 the first graph is accuracy of training set. Accuracy of training set is around 90%. Blue line in first graph is line of accuracy and black dot line is validation graph. Validation graph is use to tell model good or not good. After that validation will adjust parameter. The second graph is loss. Loss value used to tell error in model. Red line in second graph is line of accuracy and black dot line is validation graph. Iteration mean the round that has update parameter. In order to measure the performance of the proposed method, the statistical methods called recall and precision are used as in Eq. (4.1) and (4.2) respectively

$$recall = \frac{TP}{TP+FN} \tag{4.1}$$

$$precision = \frac{TP}{TP+FP} \tag{4.2}$$

where $recall$ is the accuracy interested in reality, $precision$ is the accuracy interested in prediction, TP is true positive when a bounding box contains a character, FN is false negative when no bounding box cannot be drawn for a character, and FP is false positive when a bounding box contains non-text character.

Table 1 show experimental of ICDAR 2013 before using AlexNet and after using AlexNet. The second column shows True positive. The third column shows False negative. The forth column shows False positive.

Table 1 show experimental of ICDAR 2013

Result	True positive	False positive	False negative	recall	precision
Before use AlexNet	4115	10376	1319	75.72%	28.39%
After use AlexNet	3905	6763	1529	71.87%	36.59%

From the evaluation results, The proposed method before use AlexNet has 75.72% recall and 28.39% precision . The proposed method before use AlexNet has 71.87% recall and 36.59% precision. From this table AlexNet can classfy character correct around 95% and non-character correct only 35%.

Figure 4.6 shows the results of text that can be totally localized and extracted. The method can localize all characters when the color of text is different from background.

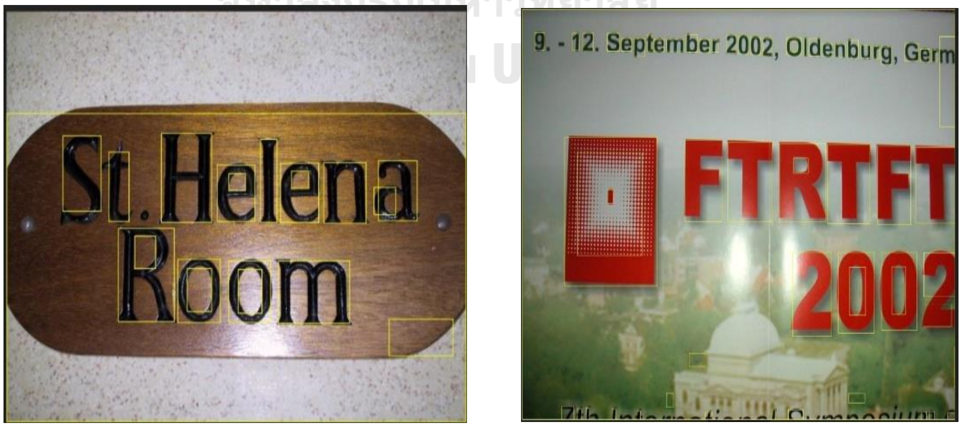


Figure 4.6. The results of text that can be localized and extracted 100%.

.

Figure 4.7 shows the results of text localization with different text sizes in an image. The method can effectively localize text with different sizes in an image.

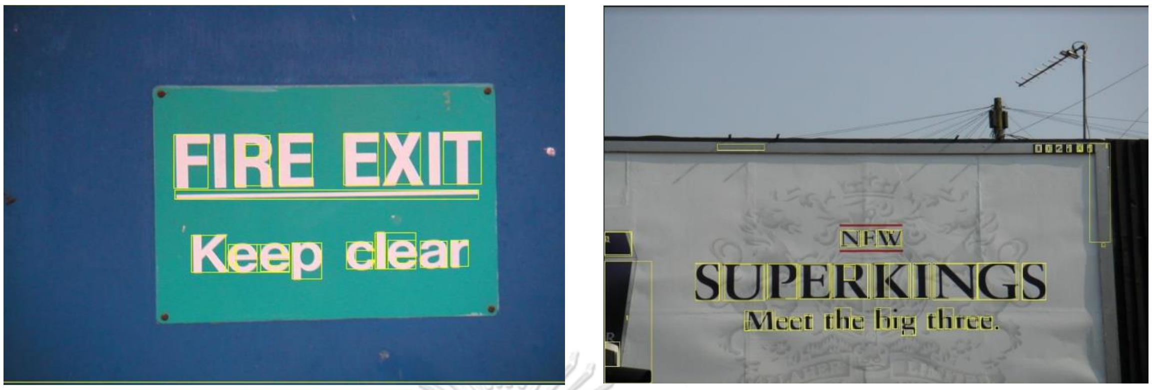


Figure 4.7 The results of text localization with different text sizes

Figure 4.8 shows the result of text localization even when the characters are occluded by the bases of spot light. This method can localize I, G, and E correctly. The reason that can localize because when change to binary image character I, G and E has color the same with spot light and blending in one object,

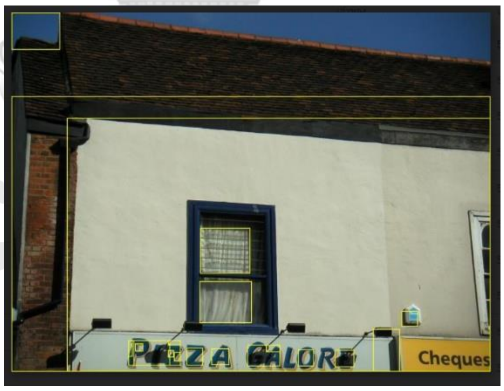
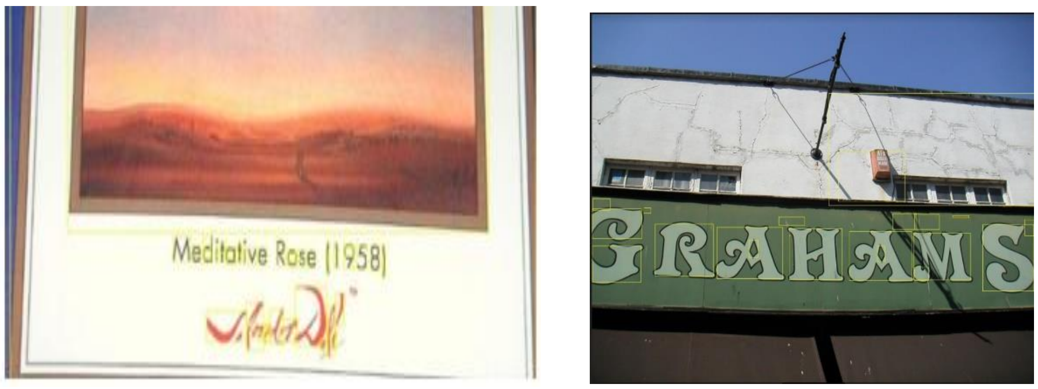


Figure 4.8 The result of occluded text localization

Figure 4.9 shows the results of text localization with special font. Red characters in Fig 4.9(a) and white characters in Figure 4.9(b) are characters with special fonts which are not included in the training data set but the method can localize all characters except character G in Figure 4.9(b) because it is at the edge of an image and the character is splited into two components.



(a) red special characters (b) white special characters

Figure 4.9 The results of text localization with special characters
(a) red special characters (b) white special characters

Figure 4.10 shows the results of text localization with blurred characters. In Figure 4.10, the method can localize some characters but not all of them. The problem is because some characters are very blurred and they cannot be classified.

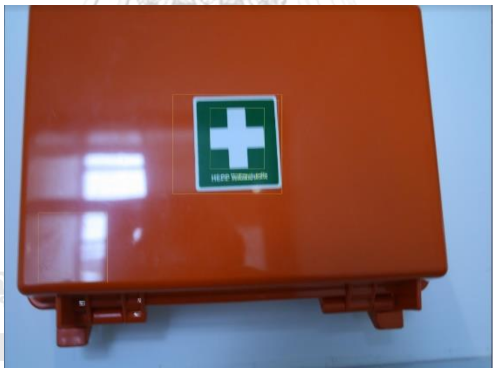


Figure 4.10 The result of text localization with blurred text

Figure 4.11 shows the result of text localization with low illunmination. The proposed method cannot find all characters in a binary image. Many characters disappeared from an image and some of them are connected to the border of a monitor.

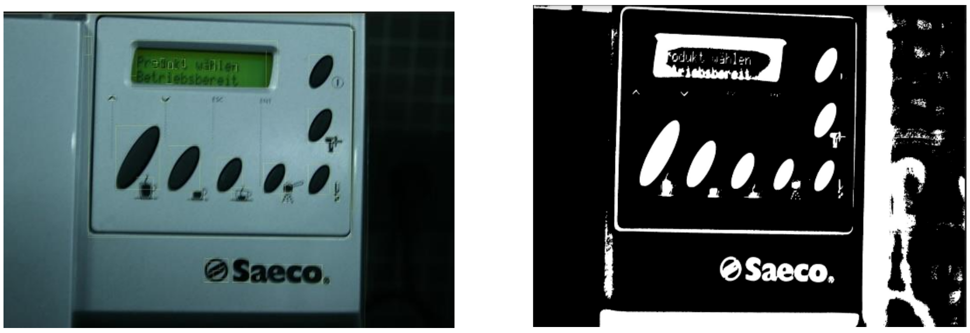


Figure 4.11 The result of low illunmination

Figure 4.12 shows the result of text localization with high illunmination. The method can find almost all of the characters except number 5 because its illumination is very high due to spot light .

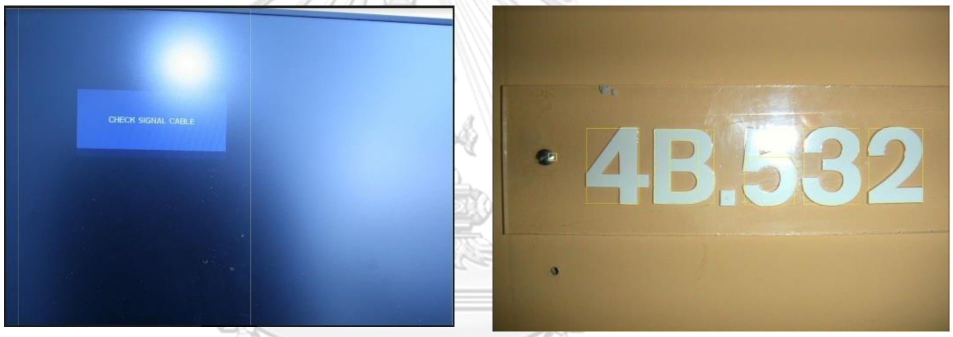
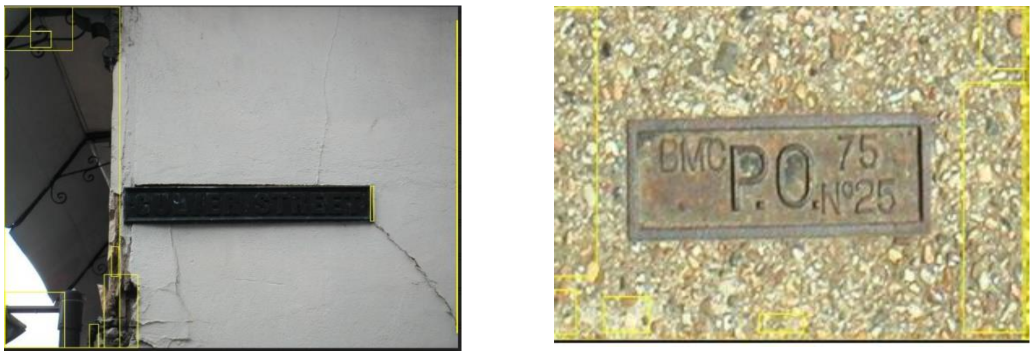
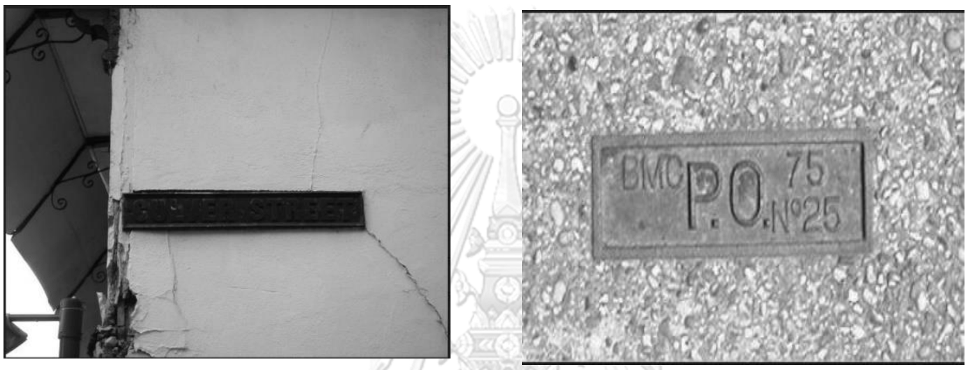


Figure 4.12 The result of high illunmination

The results of text that cannot be localized and extracted are shown as Figure 4.13. The reason that the proposed method cannot localized characters is because the text’s color is similar to the background which causes the gray scale of text and background to be the same.



(a) RGB image



(b) Gray scale image

Figure 4.13 The result of text that cannot localize and extract character (a) RGB image, (b) gray scale image

CHAPTER V
Conclusions

From the results, the recall of the proposed method is around 71.87% which means that most of the bounding boxes of character can be found correctly but there are still some bounding boxes that are not correct. The reason that the proposed method cannot find the bounding boxes is when the color of background and character are the same. Figure 5.1 shows the results of images whose position of text can be localized correctly. On the other hand, the precision of the proposed method is about 36.59% because of two main reasons. The first reason is because the number of non-character training data and the number of character training data are not balanced. In this work, we used 3876 from non-character and 570 from character that mean it has probability that non-character not enough. The second reason is because some of the non-character objects are similar to *i* and *l* characters. Moreover, the color inside the bounding boxes of *i* and *l* usually have only one color that look like background which cause AlexNet to classify them as non-character object (background). Figure 5.2 shows samples of non-character training data and character training data that look similar.

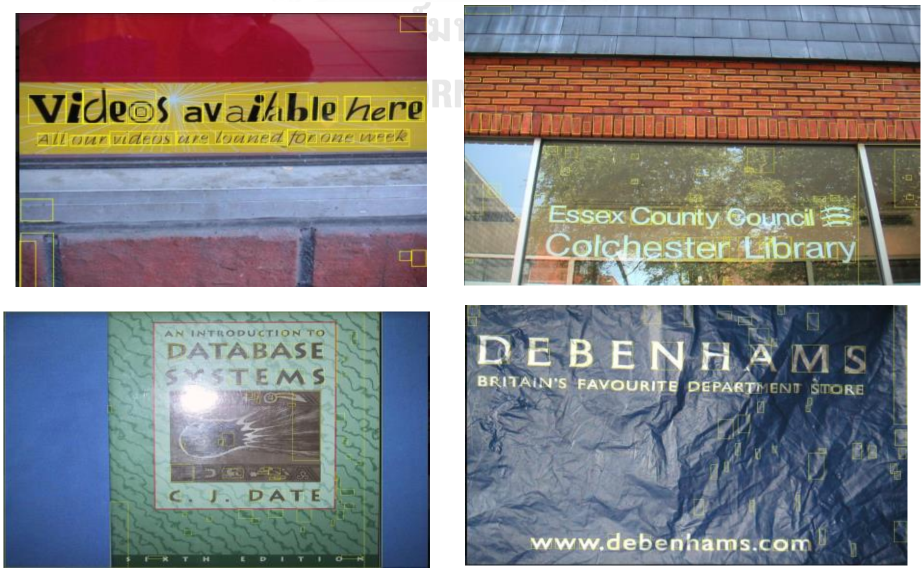
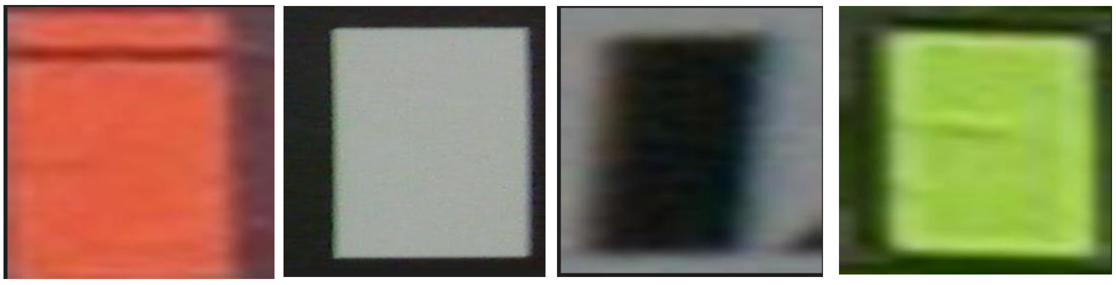
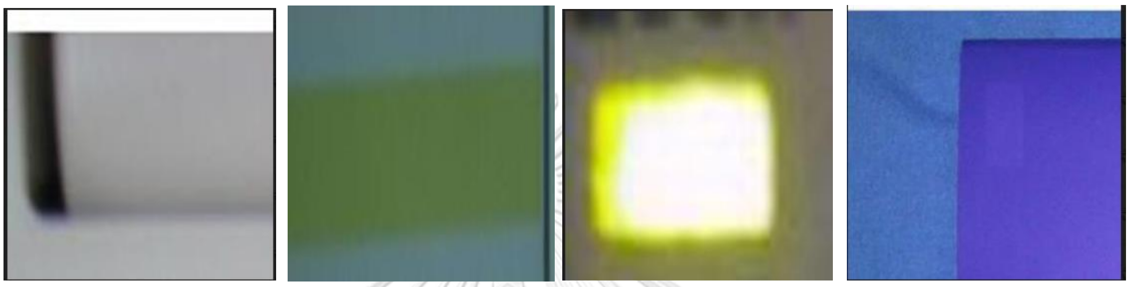


Figure 5.1 The results from the proposed method



(a)



(b)

Figure 5.2 Samples of training data that look similar (a) character, (b) non-character

5.1 Future work

The proposed method can localize and extract text from the background with texture and noise in digital images; however, the method can be improved by modifying the thresholding method, changing the text-to-character technique, and trying other convolutional neural network, such as VGG16, VGG19, GoogleNet, ResNet. After the precision and the recall yield high accuracy, the proposed method can be extended to character classification using one of the deep learning neural network with 62 classes of training data consisting of 26 uppercase English alphabet classes, 26 lowercase alphabet classes and 10 classes of number.

REFERENCES

[1] X.C. Yin, X. Yin, K. Hwang, and H. W. Hao, "Robust Text Detection in Natural Scene Images," IEEE Trans Pattern Analysis and Machine Intelligence (TPAMI), vol.36, pp. 970-983, May 2014.

[2] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, and S. R. Mestre, "ICDAR 2013 Robust Reading Competition," 12 th International Conference on Document Analysis and Recognition, pp. 1484-1493, August 2013.

[3] K.S. Satwashil and V.R. Pawar, "Integrated Natural Scene Text Localization and Recognition," International Conference on Electronics, Communication and Aerospace Technology (ICECA), vol. 1, pp. 371-374, April 2017.

[4] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform," Computer Society on Computer Vision and Pattern Recognition, pp. 2963-2970, August 2010.

[5] S. Liu, Y. Zhou, Y. Zhang, Y. Wang, and W. Lin, "Text Detection in Natural Scene Images," 15th Pacific-Rim Conference on Multimedia, pp. 123-133, December 2014.

[6] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust Text Detection in Natural Images with Edge-Enhamced Maximally Stable Extremal Regions," 18th IEEE International Conference on Image Processing, pp. 2609-2612, September 2011.

[7] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading Text in Uncontrolled Conditions," 2013 IEEE International Conference on Computer Vision, pp.785-792, March 2014.

[8] H. I. Koo and D. H. Kim, "Scene Text Detection via Connected Component Clustering and Nontext Filtering," IEEE Trans Image Process, vol. 22, no. 6, pp. 2296-2305, June 2013.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 25th International Conference on Neural Information Processing Systems, vol. 1, pp.1097-1105, January 2012.

VITA

NAME	Pukjira Pattaranuprawat
DATE OF BIRTH	23 February 1995
PLACE OF BIRTH	Somdetphraphutthaloetla Hospital Amphoe Muang Changwat Samut Songkhram
INSTITUTIONS ATTENDED	B.Sc. King Mongkut's University of Technology Thonburi
HOME ADDRESS	856/5 Tambon Maeklong Amphoe Muang Changwat Samut Songkhram 75000