HANDWRITTEN RECOGNITION SYSTEM FOR PALI CARDS OF BUDDHADASA INDAPANNO

WANWISA CHEVAKULMONGKOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2013

COPYRIGHT OF MAHIDOL UNIVERSITY

Thesis
entitled

HANDWRITTEN RECOGNITION SYSTEM FOR PALI CARDS OF BUDDHADASA INDAPANNO

……..……..…..……
Miss Wanwisa Chevakulmongkol
Candidate

……..……..…..……
Lect. Tanasanee Phienthrakul, Ph.D.
Major advisor

……..……..…..……
Lect. Supaporn Kiattisin, Ph.D.
Co-advisor

……..……..…..……
Asst.Prof. Adisorn Leelasantitham, Ph.D.
Co-advisor

……..……..…..……
Prof. Banchong Mahaisavariya, M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……..……..…..……
Lect. Supaporn Kiattisin, Ph.D.
Program Director
Master of Science Program in Technology of Information System Management
Faculty of Engineering
Mahidol University

Thesis
entitled

HANDWRITTEN RECOGNITION SYSTEM FOR PALI CARDS OF BUDDHADASA INDAPANNO

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)
on
May 7, 2013

……
Miss Wanwisa Chevakulmongkol
Candidate

……
Lect.Waranyu Wongseree, Ph.D.
Chair

……
Asst.Prof. Adisorn Leelasantitham, Ph.D.
Member

……
Lect. Tanasanee Phienthrakul, Ph.D.
Member

……
Lect. Kairoek Choeychuen, Ph.D.
Member

……
Lect. Supaporn Kiattisin, Ph.D.
Member

……
Prof. Banchong Mahaisavariya, M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……
Lect. Worawit Israngkul, M.S.
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

I would like to express my gratitude to my major advisor Lect. Tanasanee Phienthrakul and my co-advisor, Lect. Supaporn Kiattisin, and Asst.Prof.Adisorn Leelasantitham, for excellent guidance and valuable comments in this research.

I also wish to thanks my committee: Lect. Waranyu Wongseree and Lect. Kairoek Choeychuen for their kindness in providing suggestions to my project.

I would like to thanks Asst.Prof. Worasit Choochaiwattana, Asst.Prof. Puttachard Mekthong, and Lect. Suwin Apichartpattanasiri for good advices, which I could adapt to use in my thesis. Moreover, I would like to thank Lect. Pornchai Chanyakorn, who introduced me to Buddhadasa Indapanno Archives.

I would also like to acknowledge Buddhadasa Indapanno Archives for giving the handwritten images of Buddhadasa Indapanno in this research and the staffs of Buddhadasa Indapanno Archives for introducing me to the topic and their help and support the important data for this research.

I would like to special thanks to the lecturers and staffs of Technology Information System Management and Computer Engineering, Faculty of Engineering, Mahidol University for their valuable advice.

Finally, I am grateful to my family for their encouragement, love, and care and to my best friends in TISM 51 Plan A for their help and friendship.

Wanwisa Chevakulmongkol

HANDWRITTEN RECOGNITION SYSTEM FOR PALI CARDS
OF BUDDHADASA INDAPANNO

WANWISA CHEVAKULMONGKOL 5136046 EGTI/M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORY COMMITTEE: TANASANEE PHIENTHRAKUL, Ph.D., SUPAPORN KIATTISIN, Ph.D., ADISORN LEELASANTITHAM, Asst.Prof.,Ph.D.

ABSTRACT

The Dhamma principles of Buddhadasa Indapanno are highly valuable. In order to preserve and propagate these principles, the Buddhadasa Indapanno's handwritten images are kept in digital image file format. However, searching from image files is difficult. Hence, optical character recognition is one method to search these files easily. This research proposes a handwriting recognition system for the files of Buddhadasa Indapanno. One hundred and forty-eight handwritten Pali cards were used in this study. This system consists of four main processes which are pre-processing, character segmentation, feature extraction, and character recognition.

In pre-processing, the contrast of the images was automatically adjusted. Then these images were converted to gray scale images. Median filtering was used to remove noise in binary images. Connected Component Labeling and Projection Profile were applied for character segmentation. The zoning method was used to extract the features of single characters. Finally, a feed-forward backpropagation neural network was used to recognize the characters. There were 34,020 characters used for testing. The accuracy of character recognition was satisfactory. Moreover, this system can be applied to other similar handwritten images.

KEY WORDS: BUDDHADASA INDAPANNO / CHARACTER RECOGNITION / ARTIFICIAL NEURAL NETWORK

119 pages

ระบบรู้จำลายมือเขียนสำหรับบัตรบาลีของท่านพุทธทาส อินทปัญโญ
HANDWRITTEN RECOGNITION SYSTEM FOR PALI CARDS
OF BUDDHADASA INDAPANNO

วันวิสาข์ ชีวกุลมงคล 5136046 EGTI/M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : ธนัสนี เพียรตระกูล, Ph.D., สุภาภรณ์ เกียรติสิน, Ph.D.,
อดิศร ลีลาสันติธรรม, Asst.Prof.,Ph.D.

บทคัดย่อ

หลักธรรมคำสอน ของท่านพุทธทาส อินทปัญโญ ล้วนแต่เป็น ผลงานอันทรงคุณค่า เพื่อเก็บรักษาคำสอนเหล่านี้ให้คงอยู่ เอกสารลายมือของท่านพุทธทาส อินทปัญโญ จึงถูกเก็บอยู่ในรูปแบบของ เอกสารเชิงภาพดิจิตอล แต่ถึงอย่างไรก็ตาม การค้นหาข้อความจากเอกสารเชิงภาพนั้นทำได้ยาก ดังนั้น ระบบรู้จำตัวอักษรลายมือเขียนจึงเป็นวิธีการที่ดีในการเก็บรักษาเอกสารเหล่านี้ ในงานวิจัยนี้จึงได้นำเสนอระบบรู้จำลายมือเขียน ของท่านพุทธทาส อินทปัญโญขึ้น ภาพลายมือที่นำมาใช้ในการศึกษาเบื้องต้นนี้เป็นภาพตัวอักษรลายมือที่อยู่ในหมวดของบัตรบาลี มีจำนวนทั้งสิ้น 148 ภาพ โดยระบบรู้จำตัวอักษรลายมือนี้ ประกอบด้วย 4กระบวนการหลัก คือ กระบวนการประมวลผลภาพเบื้องต้น กระบวนการตัดตัวอักษร กระบวนการสกัดคุณลักษณะของตัวอักษร และกระบวนการรู้จำตัวอักษร

ในกระบวนการประมวลผลภาพเบื้องต้น จะทำการปรับคอนทราสของภาพแบบอัตโนมัติ หลังจากนั้นภาพเหล่านี้จะถูกแปลงไปเป็นภาพระดับเทา และใช้ตัวกรองมัธยฐาน(Median filter) ในการกำจัดสัญญาณรบกวนที่เกิดขึ้นบนภาพขาว-ดำ ส่วนเทคนิคการเชื่อมต่อขององค์ประกอบ (Connected Component Labeling) และ การทำโปรเจคชั่น (Projection Profile) ได้ถูกนำมาประยุกต์ใช้ในกระบวนการตัดแยกตัวอักษร และทำการสกัดคุณลักษณะพิเศษของตัวอักษรโดยใช้วิธีการแบ่งโซน โดยโครงข่ายประสาทแบบไปข้างหน้าและย้อนกลับได้ถูกนำมาใช้ในกระบวนการรู้จำตัวอักษร ซึ่งชุดข้อมูลตัวอักษรที่นำมาใช้ในการทดสอบ มีจำนวนทั้งสิ้น 34,020 ตัวอักษร โดยผลความถูกต้องของระบบในการรู้จำตัวอักษรนี้เป็นที่น่าพอใจ และนอกจากนี้ ระบบนี้ยังสามารถนำไปประยุกต์ใช้กับภาพที่มีลักษณะใกล้เคียงกันได้

119 หน้า

CONTENTS

		Page
ACKNOWLEDGEMENTS		**iii**
ABSTRACT (ENGLISH)		**iv**
ABSTRACT (THAI)		**v**
LIST OF TABLES		**ix**
LIST OF FIGURES		**x**
CHAPTER I	**INTRODUCTION**	**1**
	1.1 Background and Problem Statement	1
	1.2 Objective of the Study	3
	1.3 Scope of the Study	3
CHAPTER II	**LITERATURE REVIEW**	**4**
	2.1 Handwritten Character Recognition	4
	2.2 Digital Image Processing	6
	2.2.1 Image Digitization	6
	2.3 Pre-processing Process	8
	2.3.1 Color Transformations	8
	2.3.2 Thresholding	9
	2.3.3 Median Filter	10
	2.4 Character Segmentation	11
	2.4.1 Connected Component Labelling	12
	2.4.2 Projection Profile	14
	2.5 Feature Extraction Process	15
	2.5.1 Features Extracted from Grayscale Images	16
	2.5.2 Feature Extracted from Binary Image	16
	2.5.3 Feature Extracted from Binary Contour	17
	2.5.4 Feature Extracted from Vector Representation	17
	2.6 Character Recognition Process	18

CONTENTS (cont.)

		Page
	2.6.1 Neural Network Approach	19
	2.6.2 The Back-propagation Algorithm	20
CHAPTER III	**MATERIALS AND METHODS**	**22**
	3.1 Steps of Work	22
	3.1.1 Digital Image Processing Theory and Related Work	23
	3.1.2 Data Collection	23
	3.1.3 Implementation Process	25
	3.1.4 Evaluation	27
	3.2 Material	28
	3.3 Research Schedules	28
CHAPTER IV	**PROPOSED SYSTEM**	**29**
	4.1 Pre–processing Process	30
	4.2 Character Segmentation	40
	4.2.1 Connected Component Labelling	41
	4.2.2 Projection Profile	51
	4.2.3 Line Position Detection	54
	4.3 Feature Extraction	55
	4.4 Character Recognition	57
	4.4.1 Activation Function	59
	4.4.2 Training Function	60
CHARPTER V	**EXPERIMENTAL RESULTS**	**62**
	5.1 The Result of Pre-processing	62
	5.2 The Result of Character Segmentation	80
	5.2.1 Connected Component Labelling Process	81
	5.2.2 The Projection Profile Process	87

CONTENTS (cont.)

		Page
	5.3 The Result of Line Position Detection	90
	5.4 The Result of Zone Analysis	92
	5.5 The Result of Character Recognition	93
	5.5.1 10-Fold Cross Validation Test	93
	5.5.2 Creation of Neural Network Model	94
CHAPTER VI	**CONCLUSION AND RECOMMENDATIONS**	**98**
	6.1 Conclusion	98
	6.2 Limitations	99
	6.3 Recommendation and Future Work	100
REFERENCES		**101**
APPENDICES		**105**
	Appendix A Character Codes	106
	Appendix B Publication	112
BIOGRAPHY		**119**

LIST OF TABLES

Table		Page
2.1	Overview of feature extraction methods for the various representation forms	17
4.1	The definition of image	31
4.2	The values are used in each image group	33
4.3	The default parameter of train the neural network model	60
5.1	The percentage of noise removing	73
5.2	The percentage of single character in each group	83
5.3	The percentage of character segmentation	87
5.4	The comparative of parameter to train the neural network model	90
5.5	The percentage of character recognition	91
5.6	The false of the character classification	93
4.9	The motifs with minimum and maximum values of Inverse difference moment $0^o$and Inverse difference moment 45^o	44
4.10	The motifs with minimum and maximum values of Inverse difference moment $0^o$and Inverse difference moment 45^o	45
4.11	The optimal distance value	49
4.12	The precision of query images	51
4.13	The average precision of 100 images	57
4.14	The category result of 20 unseen images with the least distance value	57
4.15	The top five retrieval results of the unseen example	58

LIST OF FIGURES

Figure		Page
2.1	Optical character recognition classifications	5
2.2	Coordinate convention	7
2.3	Connected component in an image	12
2.3	Connected component in an image	13
2.4	Projections	14
2.5	Neural network architecture	19
2.6	The stochastic gradient descent version of the Backpropagation algorithm for feedforward networks containing two layers of sigmoid units	20
3.1	Step of work	22
3.2	Sample of Buddhadasa Indapanno's handwritten image image	24
3.3	Sample of Buddhadasa Indapanno's handwritten image image	24
3.4	Sample of Buddhadasa Indapanno's handwritten image image	24
4.1	System overview	29
4.2	The flowchart of pre-processing	30
4.3	An example of image in group 1	31
4.4	An example of image in group 2	31
4.5	An example of image in group 3	32
4.6	An example of image in group 4	32
4.7	An example of image in group 5	32
4.8	An example of contrast adjusted image in group 1	33
4.9	An example of contrast adjusted image in group 2	34
4.10	An example of contrast adjusted image in group 3	34
4.11	An example of contrast adjusted image in group 4	34
4.12	An example of contrast adjusted image in group 5	35

LIST OF FIGURES (cont.)

Figure		Page
4.13	An example of binary image in group 1	36
4.14	An example of binary image in group 2	36
4.15	An example of binary image in group 3	36
4.16	An example of binary image in group 4	37
4.17	An example of binary image in group 5	37
4.18	An example of denoised image in group 1	39
4.19	An example of denoised image in group 2	39
4.20	An example of denoised image in group 3	39
4.21	An example of denoised image in group 4	40
4.22	An example of denoised image in group 5	40
4.23	The flowchart of character segmentation process	41
4.24	The samples result of character have single characters and touching character	42
4.25	The samples result of character have single characters and touching character and not character	42
4.26	Separated non-characters	43
4.27	Samples of single character from criterion 1	45
4.28	Samples of touching character from criterion 1	45
4.29	Samples of single character from criterion 2	46
4.30	Samples of touching character from criterion 2	47
4.31	Samples of single character from criterion 3	49
4.32	Samples of touching character from criterion 3	49
4.33	Samples of single character from criterion 4	51
4.34	Samples of touching character from criterion 4	51
4.35	Segmentation in the horizontal direction	53
4.36	Segmentation in the vertical direction	53

LIST OF FIGURES (cont.)

Figure		Page
4.37	The image is segmented in the horizontal and vertical direction	54
4.38	The text lines are separated by vertical projection	55
4.39	The image resize to 80 x 64 pixels	56
4.40	The sample of value after extracted with 4 x 4 pixels per zone	57
4.41	The neural network architecture of Buddhadasa Indapanno's handwritten	58
4.42	Tan Sigmoid curve	59
4.43	Log Sigmoid curve	60
5.1	Median Filter 3x3 in Group 1	63
5.2	Median Filter 3x3 in Group 2	63
5.3	Median Filter 3x3 in Group 3	63
5.4	Median Filter 3x3 in Group 4	64
5.5	Median Filter 3x3 in Group 5	64
5.6	Gaussian Filter 3x3 in Group 1	65
5.7	Gaussian Filter 3x3 in Group 2	65
5.8	Gaussian Filter 3x3 in Group 3	65
5.9	Gaussian Filter 3x3 in Group 4	66
5.10	Gaussian Filter 3x3 in Group 5	66
5.11	Wiener Filter 3x3 in Group 1	67
5.12	Wiener Filter 3x3 in Group 2	67
5.13	Wiener Filter 3x3 in Group 3	67
5.14	Wiener Filter 3x3 in Group 4	68
5.15	Wiener Filter 3x3 in Group 5	68
5.16	Median Filter 5x5 in Group 1	69
5.17	Median Filter 5x5 in Group 2	69
5.18	Median Filter 5x5 in Group 3	69

LIST OF FIGURES (cont.)

Figure		Page
5.19	Median Filter 5x5 in Group 4	70
5.20	Median Filter 5x5 in Group 5	70
5.21	Gaussian Filter 5x5 in Group 1	71
5.22	Gaussian Filter 5x5 in Group 2	71
5.23	Gaussian Filter 5x5 in Group 3	71
5.24	Gaussian Filter 5x5 in Group 4	71
5.25	Gaussian Filter 5x5 in Group 5	71
5.26	Wiener Filter 5x5 in Group 1	73
5.27	Wiener Filter 5x5 in Group 2	73
5.28	Wiener Filter 5x5 in Group 3	73
5.29	Wiener Filter 5x5 in Group 4	74
5.30	Wiener Filter 5x5 in Group 5	74
5.31	Original Binary Image	76
5.32	Result of Median Filter 3x3	76
5.33	Result of Median Filter 5x5	76
5.34	Result of Wiener Filter 5x5	76
5.35	The original images and binary images of 5 groups	78
5.36	A sample of input image in character segmentation	80
5.37	A sample of images after segmented by connected component labeling process	81
5.38	A sample of images after segmented by connected component labeling process	81
5.39	A sample of images after segmented by connected component labeling process	82
5.40	A sample result of image is not character image	82
5.41	A sample of character image after removed not character	83

LIST OF FIGURES (cont.)

Figure		Page
5.42	A sample of character image after removed not character	83
5.43	A sample of character image after removed not character	84
5.44	A sample result of single character image	84
5.45	A sample result of single character image	85
5.46	A sample of touching character images	86
5.47	A sample of touching character images	87
5.48	A sample result of character images after segmented by horizontal segmentation.	87
5.49	A sample of single character images result	88
5.50	A sample of single character images result	88
5.51	A sample of single character images result	89
5.52	The sample of vertical projection	90
5.53	The text lines are separated by vertical projection	91
5.54	The result of line separation from Free OCR to word version 5.6.5 program	91
5.55	The result of line separation from this research	92
5.56	The tendency of accuracy rate	94
5.57	The similarity of consonant	95
6.1	The sample of character segmentation error	100

CHAPTER I
INTRODUCTION

1.1 Background and Problem Statement

Handwritten characters are important to communicate such as many letters, notes and lectures were written by human. The Buddhist's teaching was recorded by handwriting. Some of handwritten documents are valuable documents that should be maintained and preserved. These documents were written on paper or natural materials, which were easy to be destroyed. When the technology had been developed, these documents were converted to the digital image. This approach preserves the documents to existence. The knowledge of image processing and optical character recognition (OCR) were applied to convert the documents into digital file that are easier to publish and search.

The handwritten recognition is classified into two types; off-line and on-line handwriting recognition methods [1]. In on-line recognition, electronic pen is used as input device. The input characters will be analyzed real-time. For off-line recognition, the printing documents or handwritten documents are scanned, and saved into the computer.

This research uses Buddhadasa Indapanno's handwritten image from Buddhadasa Indapanno's archives. Dhamma principles of Buddhadasa Indapanno are high valuable and they are used in all generation. Buddhadasa Indapanno had perfect habitual practice, and people in world praised as the great philosopher who influence to the Buddhism. Furthermore, the United Nations Educational, Scientific and Cultural Organization (UNESCO) declared Buddhadasa Indapanno is the important person in the world, on 20 October 2005.

After the death of Buddhadasa Indapanno, Suan Mokkh Phalaram, Dhamma foundation, and disciples preserve all of manuscripts and equipments. In preliminary of survey, there are more than 20,000 lists of note book about 600,000 pages, 50,000 photographs, 1,900 gigabyte of sound and a lot of historical records.

Suan Mokkh Phalaram entrusts the Buddhadasa Indapanno's archives to collect, maintain, and preserve of Buddhadasa Indapanno's manuscripts. The Buddhadasa Indapanno's archives store the manuscript and digital image document files. The manuscripts are stored in temperature and moisture control room and the document are preserved in the standard equipments. The handwritten documents of Buddhadasa Indapanno are scanned into digital images and keep in the digital database. Therefore, OCR can be applied to convert digital images to characters. After this process, the words in these documents can be searched.

Many researchers studied in character recognition such as Neural Network based handwritten character recognition system without feature extraction [1], Pradeep, J et al, proposed method to handwritten character recognition without feature extraction .The character recognition processes used feed forward back-propagation neural network to recognize of each character. Thai character recognition on tax form [2], Phoorat Pakawathana proposed method to recognize the printed Thai character on labels on tax forms. Thai handwritten character recognition by Genetic Algorithm (THCRGA) [3], Chomtip Pornpanomchai et al, presented the system frameworks to recognize the Thai handwritten character by genetic algorithm.

This research proposed the Buddhadasa Indapanno's handwritten character recognition systems. Connected Component Labeling and Projection Profile technique are used to segment the characters, in character segmentation process. Feed-forward back-propagation neural network is used to train and recognized of each character.

The handwritten documents of Buddhadasa Indapanno are the ancient documents. These documents are yellow-brownish background and there are the difference in darkness and brightness of backgrounds. For each page, there are both single characters and touched characters. The touched characters mean many characters are connected, characters and vowels are connected, vowels and tones are connected, and three levels (characters, vowels, and tones) are connected.

The Buddhadasa Indapanno's handwritten character recognition system proposes in this research that consists of 4 processes, i.e., Pre-processing, Character segmentation, Feature extraction, and Character recognition.

1.2 Objective of the Study

The objective of this research is to design and develop the system of the of Buddhadasa Indapanno's handwritten recognition. Connected component labeling and projection histograms method are used to separate each character in character segmentation processes, and feed-forward back-propagation neural network is applied to handwritten character recognition.

1.3 Scope of the Study

1) The images of Buddhadasa Indapanno's handwriting are used in this research have 148 pages.

2) These images are in the Pali card category.

3) Digital image documents are stored in TIFF file with 300 dots per inch of resolution.

4) The touching characters will be used to training neural in unknown class.

5) The accuracy of the systems will be compared with the original image.

CHAPTER II
LITERATURE REVIEW

2.1 Handwritten Character Recognition

Optical character recognition (OCR) is one of the most successful applications of automatic pattern recognition [4]. Furthermore, OCR is the process of translating images of handwritten, typewritten, or printed text into a format understood by machines for the purpose of editing, indexing, searching, or compression[5], [6].

The most difficult problem in the field of OCR is the recognition of unconstrained cursive handwriting. The present tools for modeling almost infinitely many variations of human handwriting are not yet sufficient. The similarities of distinct character shapes, the overlaps, and interconnection of the neighboring characters further complicate the problem [7].

In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. In the off-line recognition, the writing is usually captured optically by a scanner and the completed writing is available as an image. However, in the on-line system, the two dimensional coordinates of successive points are represented as a function of time. The strokes of character are made by the writer in a real time. The on-line methods have been shown to be superior to their off-line counterpart in recognizing handwritten character due to the temporal information available with the former [1], [8]. Figure 2.1 shows the type of handwritten recognition.

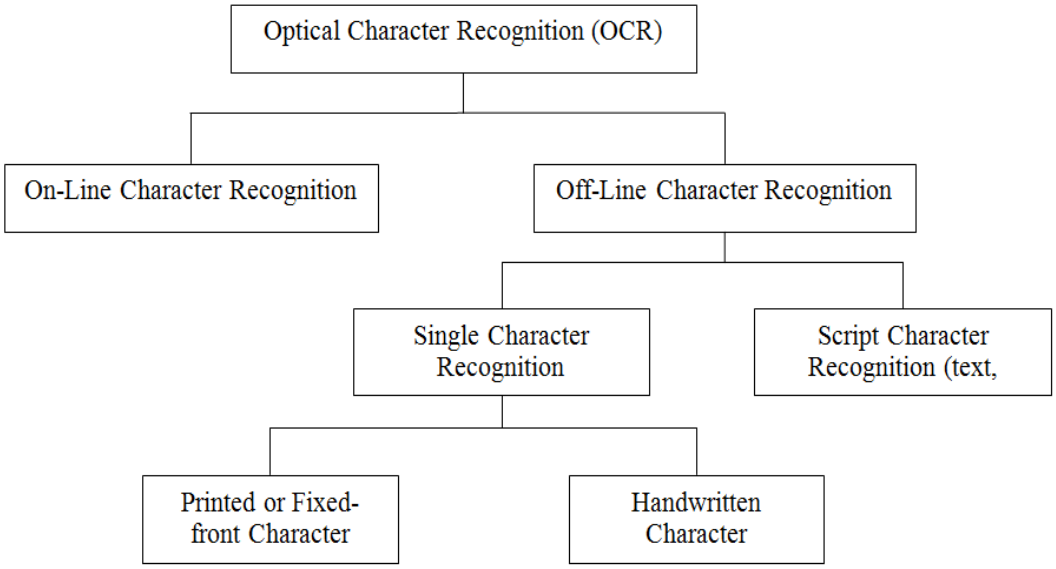


Figure 2.1 Optical character recognition classifications [9]

Many researches presented the off-line handwritten recognition technique to improve the high recognition accuracy level. The technique was used in each research is similar or difference depended on handwritten documents characteristic. For the example a research of J. Pradeep et. Al, [1] presented typical handwritten characters recognition for English alphabets without feature extraction using multilayer Feed Forward neural network. S. Tangwongsan, and C. Sumetphong [10] proposed the technique for restoration of Thai Historical Documents with broken characters. They used Kod Mai Tra Sam Duang document in their study. Nafiz Arica et.al, [7] presented off-line cursive handwriting recognition using Hidden Markov Model (HMM). The gray-level word image as input data and document image is not preprocessed for noise reduction and normalization.

2.2 Digital Image Processing

An image may be defined as a two-dimensional, $f(x,y)$, where x and y are spatial(plane) coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, there are called image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, and pixels. Pixel is the term most widely used to denote the elements of a digital image [11].

2.2.1 Image Digitization

An image to be processed by computer must be represented using an appropriate discrete data structure, for example, a matrix. An image captured by a sensor is expressed as a continuous function $f(x,y)$ of two coordinates in the plane. Image digitization means that the function $f(x,y)$ is sampled in to a matrix with M rows and N columns [12]. The values of the coordinates (x,y) become discrete quantities. For notational clarity and convenience are used integer values for these discrete coordinates. Thus, the values of the coordinates at the origin are (x,y) = (0,0). The next coordinate values along the first row of the image are represented as (x,y) = (0,1). There are important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure 2.2 shows the coordinate convention.

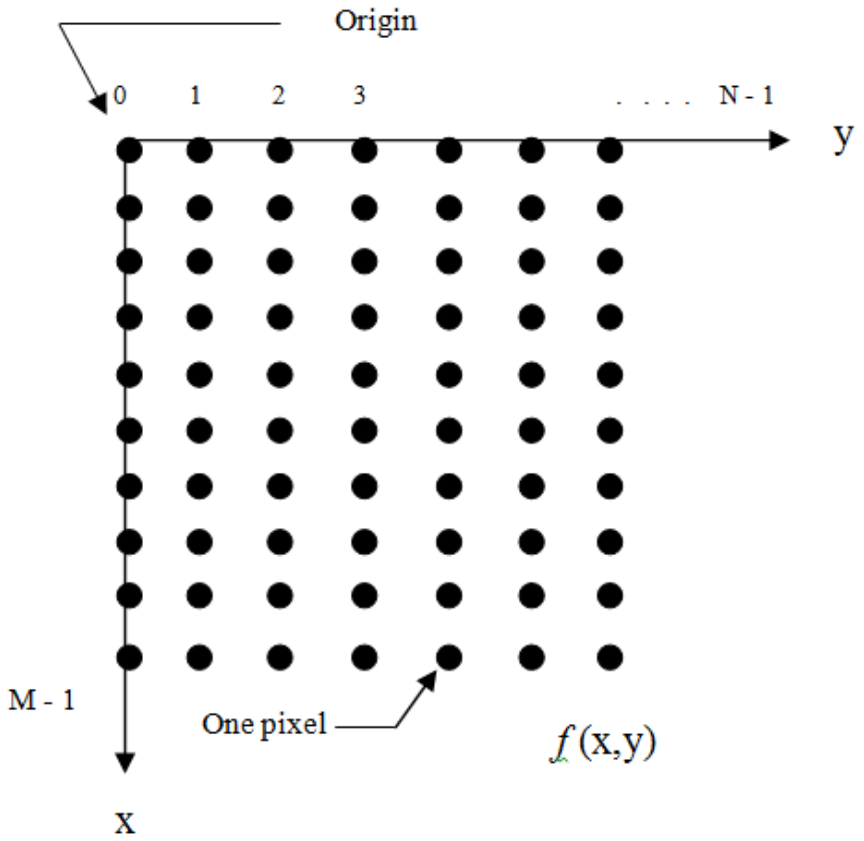


Figure 2.2 Coordinate convention

The notation introduced in the preceding paragraph allows us to write the complete M x N digital image in the following compact matrix form:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(0,1) & \dots & f(0,N-1) \\ \vdots & \vdots & & \vdots \\ \cdot & \cdot & & \cdot \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, and pixel. [11]

The most of handwritten character recognition system consists of four main stages which are:

1. Pre-processing
2. Segmentation
3. Feature extraction
4. Recognition process, but in some systems may have post processing after recognition process.

2.3 Pre-processing Process

Pre–processing is the name used for operations on images at the lowest level of abstraction-both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightness) [13].

2.3.1 Color Transformations

General, spatial domain techniques are more efficient computationally and require less processing resources to implement [11]. This technique based on processing the color components of a color image or intensity component of a monochrome image within the context of a single color model [12]. The color transformation using the expression

$$g(x,y) = T[f(x,y)] \tag{2.2}$$

when $f(x,y)$ is a color input image, $g(x,y)$ is the transformed or processed color output image, and T is an operator on f over a spatial neighborhood of (x,y) [11]. The basic intensity transformations is restrict attention to color transformation of the form

$$s_i = T_i(r_1, r_2, \ldots, r_n), \quad i = 1, 2, \ldots, n \tag{2.3}$$

where, for notation simplicity, r_i and s_i are variables denoting the color components of $f(x,y)$ and $g(x,y)$ at any point (x,y), n is the number of color components, and $\{T_1, T_2, \ldots, T_n\}$ is a set of transformation or color mapping functions that operate on r_i to produce s_i. Note that n transformations, T_i,combine to implement the single transformation function, T, in equation 2.1. The color space chosen to describe the pixels of f and g determines the value of n. In the RGB color space is selected, for example, $n = 3$ andr_1, r_2, and r_3 denote the red, green, and blue components of the input image, respectively [11].

2.3.2 Thresholding

Gray level thresholding is the simplest segmentation process. Many objects or image regions are characterized by constant reflectivity or light absorption of their surfaces; a brightness constant or threshold can be determined to segment objects and background. Thresholding is computationally inexpensive and fast – it is the oldest segmentation method and is still widely used in simple apple application; thresholding can easily be done in real time using specialized hardware [13].

The basics of intensity thresholding [11]

One obvious way to extract the objects from the background is to select a threshold, T, that separates these modes. Then, any point (x,y) in the image at which $f(x,y) > T$ is called an *object point*; otherwise, the point is called a background point. In other words, the segmented image, $g(x,y)$, is given by

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \qquad (2.4)$$

when T is a constant applicable over an entire image, the process given in equation 2.3 is referred to as global thresholding. When the value of T changes over an image the term variable thresholding is used. The term local or regional thresholding is used sometimes to denote variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y).

2.3.3 Median Filter

Median filtering is a nonlinear filtering technique that is known for preserving sharp changes in signal and for being particularly effective in removing impulsive noise. One effective use of median filters has been the reduction of high-frequency and impulsive noise in digital images without the extensive blurring and edge destruction associated with linear filters. Because the median filter is nonlinear, spectral analysis gives little insight into the filtering process. [14]

Median filter is the best filter in Order Statistic filters that are nonlinear spatial filters whose response is based on ordering (ranking). The pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The median filter, which replaces the value of a pixel by the median of the intensity values in the neighborhood of that pixel:

$$\hat{f}(x,y) = \underset{(s,t) \in S_{xy}}{median} \{ g(s,t) \} \tag{2.5}$$

The value of the pixel (x,y) is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise – reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of *impulse noise*, also called *salt and pepper noise* because of its appearance as white and black dots superimposed on an image. Thus, the principal function of median filters is to force points with distinct intensity levels to be more like their neighbors. In fact, isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $m^2/2$ (one – half the filter area), are eliminated by an $m \times m$ median filter. In this case “eliminated” means forced to the median intensity of the neighbors. Larger clusters are affected considerably less. [11]

In historical image, the pre-processing process is important to enhance the image suitable for segmentation process. The most of pre-processing process consist of noise removing, binarization. However, the steps of pre–processing processes that are depended on input document image. The pre-processing process is important to

enhance the image suitable for segmentation process. The most research using Otsu's method to convert gray scale image into binary image as [1], [2], [10], [15], [16], [17], and [18] and some research is applied the Otsu's method suitable to work such as [18] , there are using Otsu's method to choose an initial threshold value (t_o) and threshold value between t_o and 255 (the maximum gray scale value) is selected that minimizes the number of connected components in the binary image, the image resulted from thresholding at that value. Moreover, the research in [1] is using dilation and filing the holes to produce the pre-processing image suitable for segmentation after convert the image into binary image by global thresholding technique. And the methods to reduce the noise are occurred in the image. In [17] using Gaussian filtering technique to remove noise in gray scale image, and some research using median filtering for smoothing and noise elimination [15],[18], and with salt and pepper [2]. The method to reduce the noise is occurred in the image, there are using Gaussian filtering to remove noise in grayscale image. And some research using median filtering for smoothing and noise elimination [15] with salt and pepper [2]. However, the histogram stretching operation will be using normalize the background intensity of the resulting foreground image to white [18].

2.4 Character Segmentation

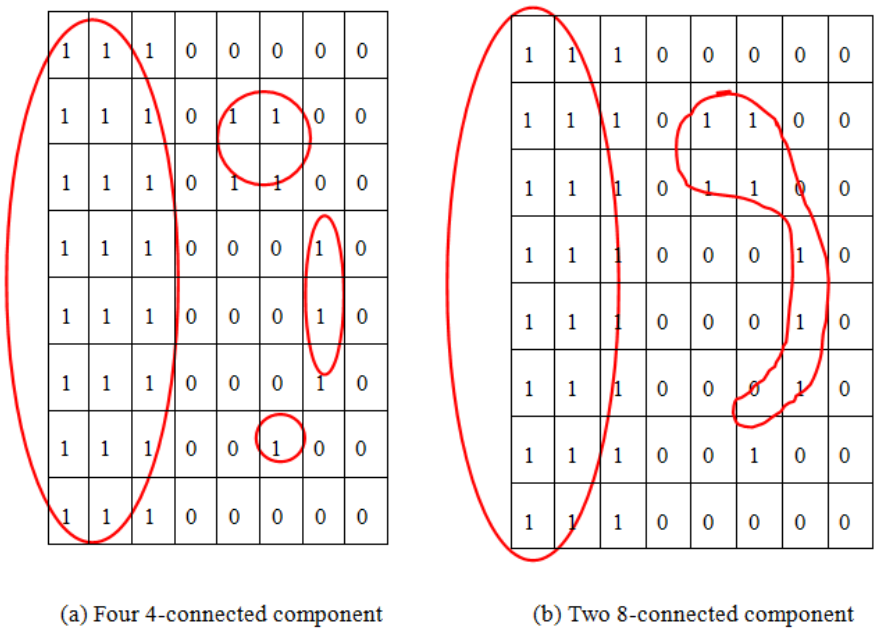
Character segmentation is a technique to separation of the image into lines, line into words, and word into single characters such as [19], [20], and [21]. The segmentation process is one of the important steps in the handwritten recognition system [20], [22], and [23], because there are directly affects to accuracy rate of recognition.

In ancient and historical document, the physical structures are harder to extract and low quality due to aging or faint typing. Most of handwritten page are consisted narrow spaced line with overlapping and touching components [24]. In this research is used connected component labeling and projection profile technique to segment the handwritten of Buddhadasa Indapanno document images.

2.4.1 Connected Component Labeling

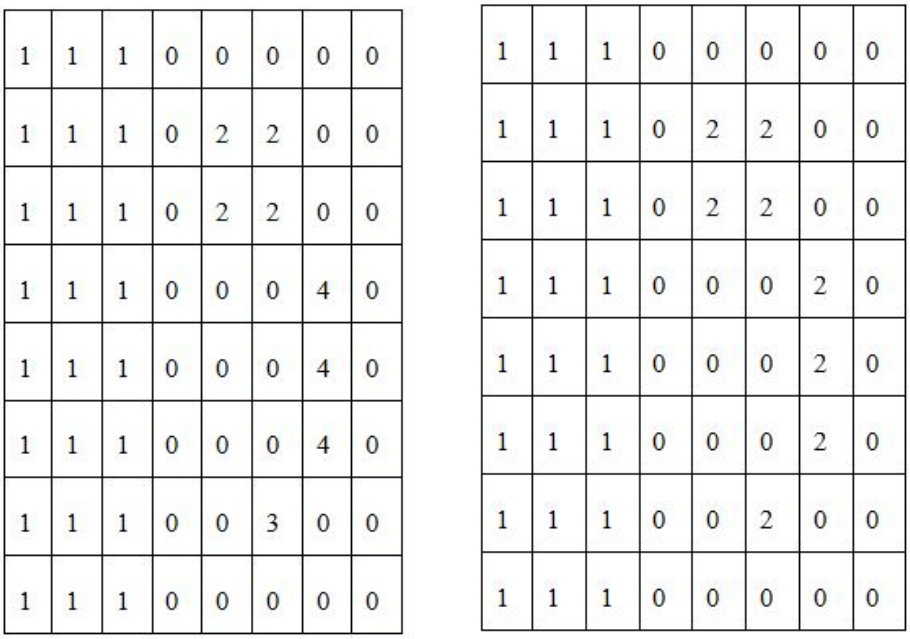
Connected component labeling is a basic operation in machine vision. The purpose of labeling is to treat the individual connected components of a digitized image as separate objects and obtain a multi valued picture in which the points corresponding to each component have a unique non – zero label [25].

The term connected component was just defined in terms of a path, and the definition of a path in turn depends on adjacency. This implies that the nature of a connected component depends on which form of adjacency we choose, with 4-and 8-adjacency being the most common. In Figure 2.1 show the effect that adjacency can have on determining the number of connected components in an image. Figure 2.3(a) show a small binary image with 4-connected components. Figure 2.3(b) show that choosing 8-adjacency reduces the number of connected components to two. Figure 2.3(c) shows the label matrix corresponding to Figure 2.3 (a). Figure 2.3(d) shows the label matrix corresponding to Figure 2.3(a) [12].



(a) Four 4-connected component (b) Two 8-connected component

Figure 2.3 Connected component in an image [12]



1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	3	0	0
1	1	1	0	0	0	0	0

(c) Label matrix obtained using 4-connectivity

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	2	0	0
1	1	1	0	0	0	0	0

(d) Label matrix obtained using 8-connectivity

Figure 2.3 Connected component in an image [12] (cont.)

The pixels in each different connected component are assigned a unique integer, from 1 to the total number of connected components. In other words, the pixels labeled 1 belong to the first connected component; the pixels labeled 2 belong to the second connected component; and so on. Background pixels are labeled 0 [12].

The connected component labeling can use to find the word boundary or segment the image into single character, if the characters not touching there is a good result. The image in [1] is segmented into single character by connected component labeling method. Each character is assigned the number with labeling process. And [26] proposed the new method to segment the touching character in Chinese handwriting with contour analysis, there are using the connected component labeling before segment the touching character by contour analysis.

2.4.2 Projection Profile

Horizontal and vertical region projections $p_h(i)$ and $p_v(j)$ are defined as

$$p_h(i) = \sum_j f(i,j) \quad , \quad p_v(j) = \sum_i f(i,j) \tag{2.6}$$

Region description by projections is usually connected to binary image processing. Projections can serve as a basis for definition of related region descriptors; for example, the width (height) of a region with no holes is defined as the maximum value of the horizontal (vertical) projection of a binary image of the region. These definitions are shown in Figure 2.4. [13]

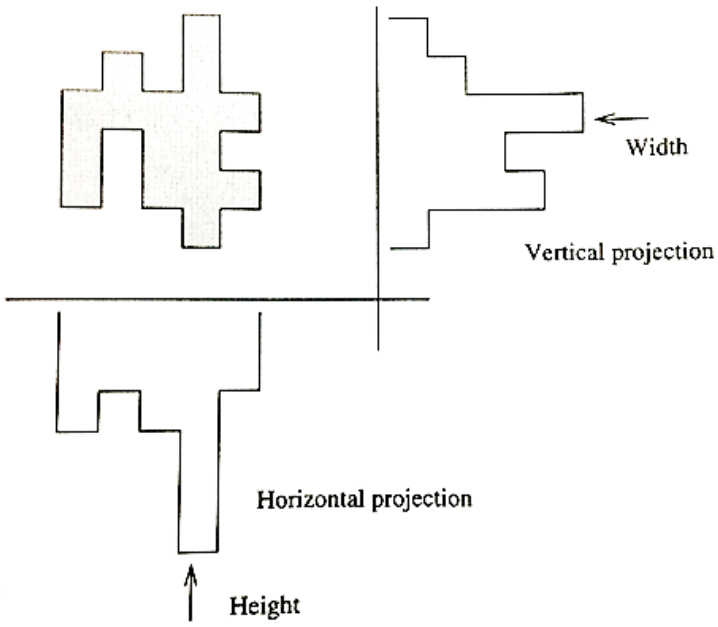


Figure 2.4 Projections [13]

Projection profiles are commonly used for printed document segmentation. This technique can be adapted to handwritten documents with little overlap. [24]

2.4.2.1 Horizontal Projection

The horizontal projection profile is a popular technique to line segmentation [17, 27]. If the page is skew corrected and the lines well separated, the histogram will present well separated peaks and valleys. However, in handwritten documents this method can be problematic because of the hill and dale writing as well as the ascenders and descenders of the characters that very often begin in a line and end up in another. Thus, the valleys of the histogram are not very clear. Although problematic, this method can give a good estimation of the number of the lines is included in a document page [27]. The technique computes horizontal projection histogram, the count of black pixels for each column of the raster image [17]. For example, in [2] segmented the character image into line by horizontal projection profiles to find the valley as a reference point for line segmentation.

2.4.2.2 Vertical Projection

The vertical projection also called the “vertical histogram” of a print line that can serve for detection of white space between successive characters. Moreover, that can indicate locations of vertical strokes in machine print, or regions of multiple lines in handprint [19]. The vertical projection profile is performed the word segmentation in each text line [27], that is obtained by summing pixel values along the horizontal axis for each y value [24].

2.5 Feature Extraction Process [4]

Feature extraction is an important method in achieving good recognition performance. Different feature extraction methods are designed for different representations of the characters, such as solid binary characters, character contours, skeletons (thinned characters) or gray-level sub-images of each individual character. In this work is focuses on feature extracted from binary image with zoning method.

2.5.1 Features Extracted from Grayscale Images

A major challenge of using gray-scale image-based methods is to locate candidate character location. One can use a locally adaptive binarization method to obtain a good binary raster image and use connected components of the expected character size to locate the candidate characters. This approach cannot be used if the character size is expected to vary.

The desired result of the localization or segmentation step is a sub-image containing one character and, expect for background pixels, no other objects. However, when print objects appear very close to each other in the input image, this goal cannot always be achieved. Often, other characters or print objects may accidentally occur inside the sub-image, possibly distorting the extracted features. This is one of the reasons why every character recognition system has a reject option.

2.5.2 Feature Extracted from Binary Image

A binary raster image is obtained by a global or locally adaptive binarization of the gray-scale input image. Choosing a better locally adaptive binarization may be alleviate the problem in character connected components, broken characters into two or more connected components but the result of this method may still not perfectly in isolate character.

The binary raster representation of a character is a simplification of the gray-scale representation. The image function Z(x,y) now takes on two values (say,0 and 1) instead of the ,say 256 gray-level values. This means that all the methods developed for the gray-scale representation are applicable to the solid binary raster representation as well.

A solid binary character may be converted to other representations, such as the outer contour of the character, the contour profiles, or the character skeleton and features may be extracted from each of these representations as well. For the purpose of designing OCR systems, the goal of these conversions is to preserve the relevant information about the character shape and discard some of the unnecessary information.

2.5.3 Features Extracted from the Binary Contour

The closed outer contour curve of a character is a closed piecewise linear curve that passes through the centers of all the pixels which are four-connected to the outside background and no other pixels. Following the curve, the pixels are visited in, say counter-clockwise order and the curve may visit an edge pixel twice at locations where the object is one-pixel wide. Each line segment is a straight line between the pixel centers of two eight-connected neighbors.

2.5.4 Feature Extracted from Vector Representation

Character skeletons are obtained by thinning the binary raster representation of the characters. The task of choosing the right one often involves a compromise; one wants one-pixel wide eight-connected skeletons without spurious branches or displaced junctions, some kind of robustness to rotation and noise and at the same time a fast and easy-to-implement algorithm. The feature extraction methods are show in Table 1.

Table 2.1 Overview of feature extraction methods for the various representation forms (gray level, binary, vector)

Grayscale sub-image	Binary		Vectors (skeleton)
	Solid symbol	Outer contour	
Template matching	Template matching		Template matching
Deformable templates	-		Deformable templates
Unitary transforms	Unitary transforms		Graph description
	Projection profile	Contour profile	Discrete features
Zoning	Zoning	Zoning	Zoning
Geometric moments	Geometric moments	Spline curve	
Zernike moments	Zernike moments	Fourier descriptors	Fourier descriptors

2.6 Character Recognition Process

Pattern recognition is a process of classification of objects into a number of categories or classes. Depending on the application, these objects can be images or signal waveforms or any type of measurements that need to be classified [28]. Due to the fact that pattern recognition is a basic attribute of human beings and other living thin *gs*, it has been taken for granted for long time. We are now expected to discover the mechanism of their recognition, simulate it, and put it into action with the modern technology to benefit the human beings [29].

Character recognition is another important area of pattern recognition, with major implications in automation and information handling. The pattern recognition software then takes over to recognize the characters, that is, to classify each character in the correct “letter, number, punctuation” class. Storing the recognized document has a twofold advantage over storing its scanned image. First, further electronic processing, if needed, is easy via a word processor, and second, it is much more efficient to store ASCII characters than a document image [28].

In the general approach of pattern recognition are four which are: Template matching, Statistical techniques, Structural techniques, and Neural Networks [27]. In this work are focuses in neural networks due to in the off-line systems, the neural networks have been successfully used to yield comparably high recognition accuracy level [30]. For the example research were use the neural network to recognize the character as, in [10] proposed Optical character recognition technique to solve the broken characters are occurred after convert image into binary image in Thai historical document “Kod Mai Tra Sam Duang”, the neural network was used to recognize the character. The average of accuracy rate is 93.07%. In [1] using neural network to recognize the character without feature extraction. The accuracy rate of character recognition is 90.19%. And in [2] using neural network to recognize Thai character with Tax Form which the accuracy rate in documents obtained from a line printer is 98.88% and 93.77 % with characters from labels on tax form. But some research using genetic algorithm to recognize Thai character in [3], there is accuracy rate is around 88%.

2.6.1 Neural Network Approach

Artificial neural networks (ANNs) provide a general, practical method for learning real-valued, discrete-valued, and vector-valued function from examples. Algorithms such as Back propagation use gradient descent to tune network parameters to best fit a training set of input – output pairs [31].

An ANNs can also be defined as a data-processing system consisting of a large number of simple, highly interconnected processing elements in an architecture inspired by the structure of the cerebral cortex of the brain. These processing elements are usually organized into a sequence of layers with full or random connections between the various layers. This arrangement is show in figure 2.5, where the input layer is a buffer that presents data to the network. This input layer is not a neural computing layer because the nodes have no input weights and no activation functions. The top layer is the output, which presents the output response for a given input. The other layer is called intermediate or hidden layer because it usually has no connections with the outside world. Typically, the input, hidden, and output layers are designated the *i*th, *j*th, and *k*th layers, respectively.

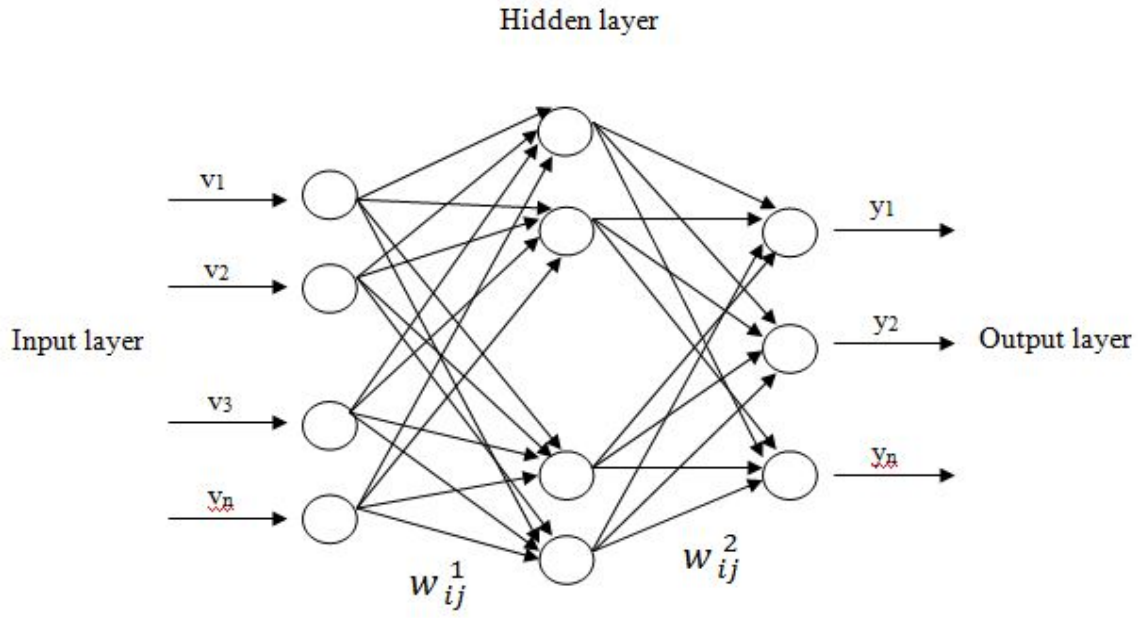


Figure 2.5 Neural network architecture [32]

Artificial neural networks do not consist of one network, but a diverse family of networks. The network topology, the individual neuron characteristics, the learning or training strategy, and the training data determine the overall function or functionality achieved. A typical neural network is fully connected, which means that each neuron in any given layer is connected with all the neurons in the next layer. [32]

2.6.2 The Back-propagation Algorithm [31]

The Back-propagation algorithm learns the weights for a multilayer network, given a network with a fixed set of units and interconnections. It employs gradient descent to attempt to minimize the squared error between the network output values and the target values for these outputs. The Back-propagation algorithm learns is shown in Figure 2.6.

Backpropagation (*training_examples*, η, n_{in}, n_{out}, n_{hidden})

Each training example is a pair of the form $(\vec{x}, \vec{t})$, *where* $\vec{x}$ *is the vector of network input values, and* $\vec{t}$ *is the vector of target network output values.* η *is the learning rate (e.g., .05).* n_{in} *is the number of network inputs,* n_{hidden} *the number of units in the hidden layer, and* n_{out} *the number of output units. The input from unit i into unit j is denoted* x_{ji}, *and the weight from unit I to unit j is denoted* w_{ji}.

- **Create a feed-forward network with** n_{in} **inputs,** n_{hidden} **hidden units, and** n_{out} **output units.**
- **Initialize all network weights to small random numbers (e.g., between -.05 and .05).**
- **Until the termination condition is met, Do**
  - **For each** $(\vec{x}, \vec{t})$ **in** *training_examples*, **Do**

    Propagate the input forward through the network:

    1. **Input the instance** $\vec{x}$ **to the network and compute the output** o_u **of every unit** u **in the network.**

    Propagate the errors backward through the network:

    2. **For each network output unit** k, **calculate its error term** δ_k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

    3. **For each hidden unit** h, **calculate its error term** δ_h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$$

    4. **Update each network weight** w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

where

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

Figure 2.6 The stochastic gradient descent version of the Backpropagation algorithm for feedforward networks containing two layers of sigmoid units

The Backpropagation algorithm is presented in Figure 2.6. The algorithm as described here applies to layered feedforward networks containing two layers of sigmoid units, with units at each layer connected to all units from the preceding layer.

CHAPTER III
MATERIALS AND METHODS

3.1 Steps of Work

This section describes the step and methodology of Buddhadasa Indapanno's handwritten character recognition. The theory and methodology of digital image processing is described in section 3.1.1. The data of Buddhadasa Indapanno's handwritten images used in this research is described in section 3.1.2. The step of implementation the system is describes in section 3.1.3. In the final, section 3.1.4 describes the evaluation of the system.

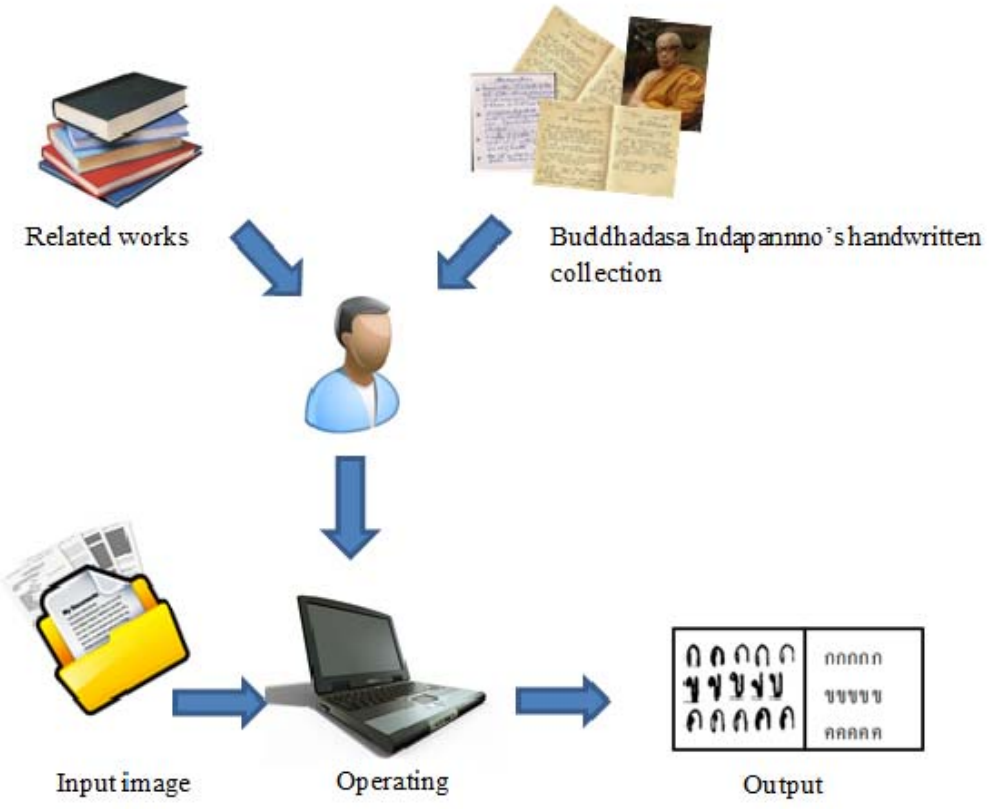


Figure 3.1 Step of work

Figure 3.1 shows the overall steps to develop the Buddhadasa Indapanno's handwritten character recognition system. The theories researches and related work on digital image processing are studied. The handwritten images are selected and collected to use in this research. After that, the knowledge of image processing and the handwritten images were selected are used to designed the Buddhadasa Indapanno's handwritten character recognition system. The accuracy of this system is compared between the output and original image.

3.1.1 Digital Image Processing Theory and Related Work

In this step, the techniques relevant to the Buddhadasa Indapanno handwritten character recognition system will be studied. Pre-processing process, character segmentation process, feature extraction process, character recognition process, and post-processing process will be applied in this research.

In pre-processing processes, the characteristic of the digital images the techniques to process two-dimensional digital image, and the techniques to enhance the quality of image such as adjusting the contrast, image binarization image, and noise removing will be studied. For character segmentation processes, the related research and techniques for dividing the handwritten character image will be searched.

Then, feature extraction process with zoning analysis will be reviewed for the methods and theories for character recognition are studied. Fed-forward back-propagation neural network is reviewed. In the post-process, the techniques to improve the performance of character recognition with Thai dictionary will be studied and applied.

3.1.2 Data Collection

The Buddhadasa Indapanno's handwritten images from Buddhadasa Indapanno Archives are used in this research. There are 148 images that are scanned with 300 dots per inch of resolution and stored in TIFF format. The samples of handwritten images are show in Figure 3.2, 3.3, and 3.4.

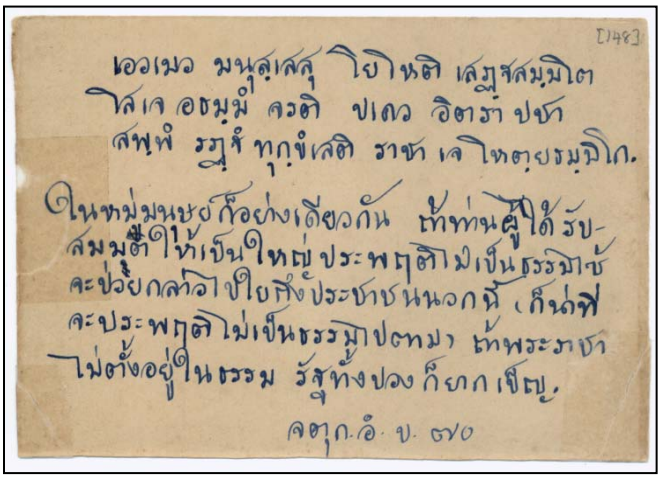


Figure 3.2 Sample of Buddhadasa Indapanno's handwritten image

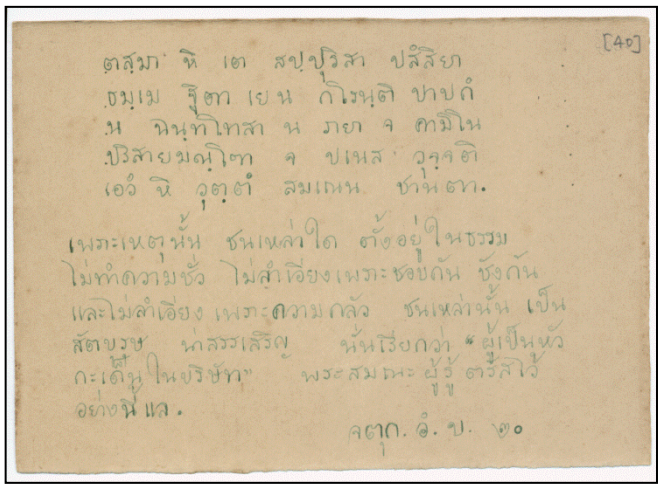


Figure 3.3 Sample of Buddhadasa Indapanno's handwritten image

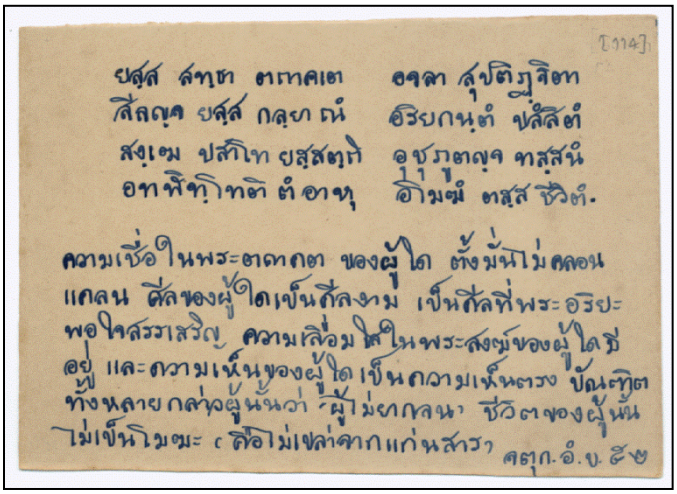


Figure 3.4 Sample of Buddhadasa Indapanno's handwritten image

In Figure 3.2, many single and touching characters are in the image and there are smudge on the background. Figure 3.3, the mostly of line in character are thin there are difficult to read. Figure 3.4, the mostly of character in the image are thick and have touching character in the horizontal and vertical. The Buddhadasa Indapanno's handwritten images are show in Figure 3.2, 3.3, and 3.4 above, each image are difference in the thick or thin of character , color of ink on character and moreover, some image has smudge on the background . From describe previous, there is a difficult problem to clear noise the image after convert the grayscale image into binary image, segmented the character into single character, and recognized the character correctly on every Buddhadasa Indapanno's handwritten images.

3.1.3 Implementation Process

Buddhadasa Indapanno's handwritten character recognition system is including 4 main steps such as Pre-processing, character segmentation, feature extraction, and character recognition.

3.1.3.1 Pre-processing

Pre-processing processes is the quality improvement, removing noise of an input image, and convert image into binary image. In this stage is including 5 sub-processes which are:

- Crop Image: Image is added to the system. In the first step, user has to crop the image in only text area. Image after cropped will be calculated the mean of image’s pixel. The average values will be used to classify the image groups to adjust the contrast in different value. The similar of image will be classified in the same group.

- Adjustment Contrast: Each of handwritten images has a difference dark or light. Therefore, to improve the brightness of image to appropriate, the image was adjustment the contrast with contrast stretching method.

- Binarization: The image after improve the quality of image with adjustment the contrast, these images was converted to the binary image by threshold value.

- Noise reduction: Noise can be occurred after converting the image into binary image. The objective of this section is removing noises in the image using median filtering method with window mask 3x3.

3.1.3.2 Character Segmentation

The character segmentation processes has two sub- processes to divide the handwritten character image into individual character, there are include of connected component labeling and projection profile. The output of this process is single characters that are data input to feature extraction processes.

- Connected Component Labeling

This method can be dividing the word images out of the background. The single characters in the image are divided. This research defined the connectivity of the neighborhood is 4-connected. The touching character cannot be segmented in this method.

- Projection Profile

This method was separated the touching character into single character images. The image input in this process will be thinning before projection. The projection profile in this research has three sub-process which are horizontal projection, vertical projection, and horizontal and vertical projection.

3.1.3.3 Feature Extraction

The character images in the character segmentation process (single character group and single character and touching character together group) are input data of this step. The important features of each character are extracted with zoning method. Before feature extraction, size of each character image will be adjusted into 80 x 64 pixel and the character images were divided by 4 x 4 pixels per zone.

3.1.3.4 Character Recognition

In this step, the important features of each character are used as input data to recognize that character with feed-forward back-propagation neural network. All characters are classified to 87 classes as the consonant are 42 classes, vowel are 17 classes, tone are 3 classes, number are 11 classes, symbol are 13 classes, and unknown is 1 class.

3.1.4 Evaluation

In this section, the accuracy of this system will be evaluated by comparison between the result and original image, the step of this section is described as follow:

1. Learning Accuracy Evaluation: All characters will be trained by 10 fold cross validation to find suitable method. Then these data will be used for creating a model. This model will be evaluate in order to get the learn accuracy.

2. Usage Testing: The accuracy of characters recognition after post - process will be measured and the results were compared between output and original images.

3. Conclusion and documentation: This step is analyzing the result of the system to conclusion and makes the documentation.

3.2 Material

Hardware:

Personal computer which had a minimum requirement as follows:

- CPU : Pentium 4, 2.4GHz
- 4 GB of RAM
- 250 GB of Hard disk

Software:

- Operating Systems : Microsoft Windows 7 Enterprise (64 bit)
- Application development Tools : MATLAB R2010a

3.3 Research Schedules

Schedule	2012												2013				
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5
Data collection	━	━															
Study research and related work		━	━	━													
System implementation				━	━	━	━	━									
System test								━	━	━							
System evaluation										━	━	━	━				
Conclusion & Documentation												━	━	━	━	━	━

CHAPTER IV
PROPOSED SYSTEM

The overall of proposed system is shown in Figure 4.1. There are 4 main steps which are Pre-processing process, Segmentation process, Feature extraction process, and Recognition process.

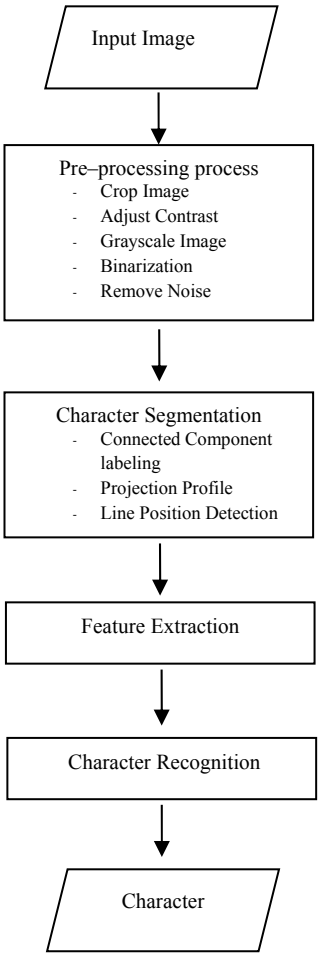


Figure 4.1 System overview

4.1 Pre–processing Process

Image Pre-processing is the process that improves the quality of image, remove noise and convert image into binary image. In this research, includes 5 sub-pre-processing steps. The flow of these processes is shown in Figure 4.2.

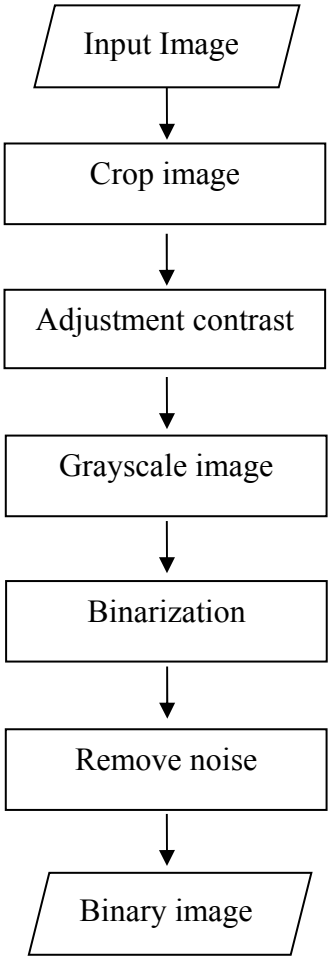


Figure 4.2 The flowchart of pre-processing

- Crop Image: Image is added to the system. In the first step, the image is cropped in only text area. When an image is added to the system, user must define the text area. Then, that image will be cropped. After that, the mean of pixels in cropped image will be calculated. The average values will be used to classify the group of image in order to adjust the contrast with the different parameter. The groups of image are classified in 5 groups. The definition of image in 5 groups is described in Table 4.1

Table 4.1 The definition of image

Group of image	Definition	Mean of pixel
Group 1	smeared on the background	< 160
Group 2	thick and small character	≥160 and <169
Group 3	normal characters	≥169 and <176
Group 4	thick and fade characters	≥176 and <180
Group 5	thin and fade characters	≥180

The sample of Buddhadasa Indapanno's handwritten in 5 groups are shown in Figure 4.3 – 4.7

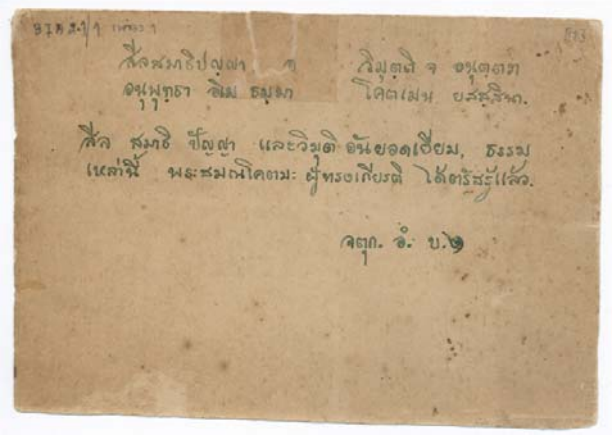


Figure 4.3 An example of image in group 1

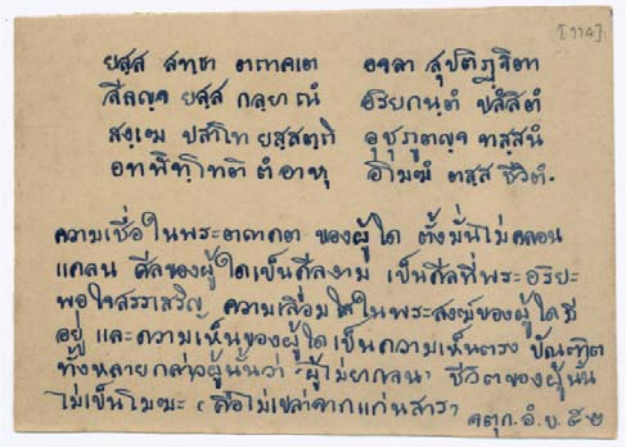


Figure 4.4 An example of image in group 2

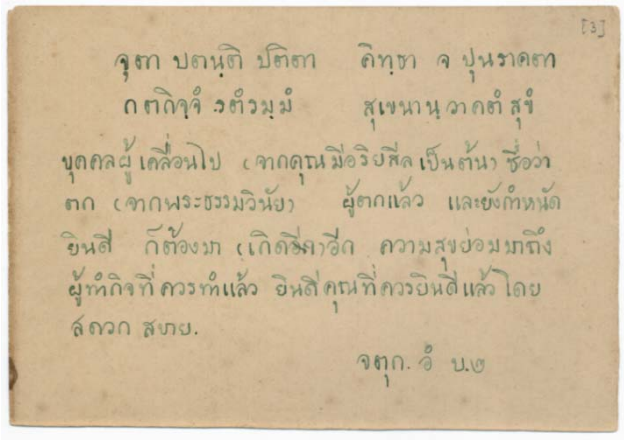


Figure 4.5 An example of image in group 3

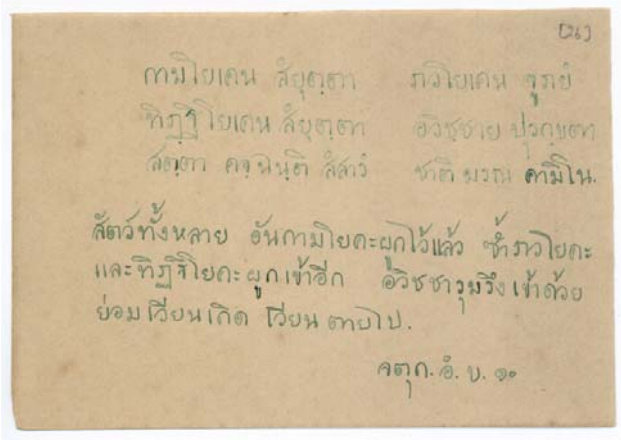


Figure 4.6 An example of image in group 4

Figure 4.7 An example of image in group 5

- Adjustment Contrast: Each of handwritten images has a difference dark or light. Therefore, to improve the brightness of image to appropriate, the image was adjustment the contrast with contrast stretching method. The syntax was show below:

$$g = imadjust\,(f, [\,low_in \quad high_in\,], [\,low_out \quad high_out\,])\,, \qquad (4.1)$$

where the intensity values in f will be map to new intensity values in g. Which the values between low_in and $high_in$ are map to values between low_out and $high_out$, that is values below low_in and above $high_in$ are clipped. The values used for adjustment the contrast in 5 groups are shown in Table 4.2.

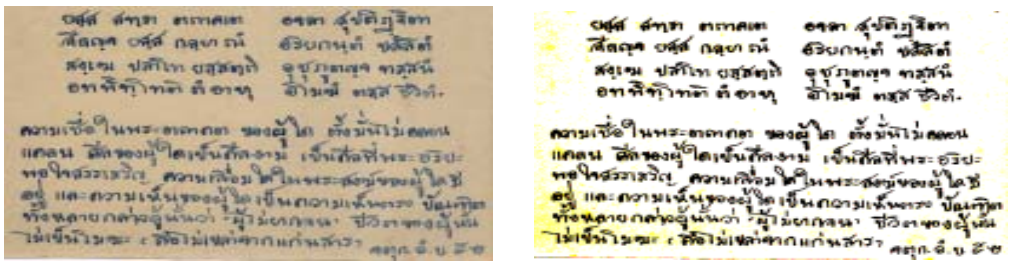
Table 4.2 The values are used in each image group

Group of image	$[\,low_in \quad high_in\,]$	$[\,low_out \quad high_out\,]$
Group 1	[0.58 0.60]	[0 1]
Group 2	[0.35 0.38]	[0 1]
Group 3	[0.65 0.68]	[0 1]
Group 4	[0.72 0.73]	[0 1]
Group 5	[0.83 0.85]	[0 1]

The sample of image in 5 groups after adjusted the contrast are shown in Figure 4.8 – 4.12

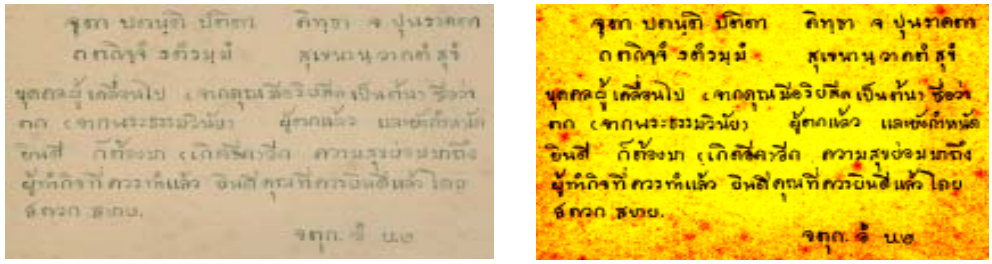
(a) Before adjust the contrast (b) After adjust the contrast

Figure 4.8 An example of contrast adjusted image in group 1



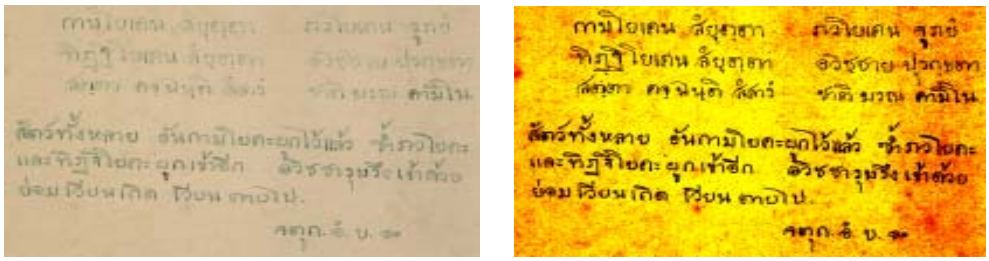
(a) Before adjust the contrast (b) After adjust the contrast

Figure 4.9 An example of contrast adjusted image in group 2



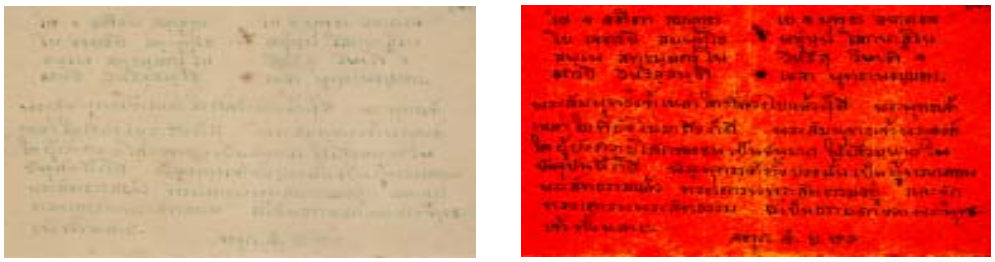
(a) Before adjust the contrast (b) After adjust the contrast

Figure 4.10 An example of contrast adjusted image in group 3



(a) Before adjust the contrast (b) After adjust the contrast

Figure 4.11 An example of contrast adjusted image in group 4



(a) Before adjust the contrast (b) After adjust the contrast

Figure 4.12 An example of contrast adjusted image in group 5

- Grayscale image: The color image after improve the quality of image with adjustment the contrast, these images was converted to grayscale image by equation 4.2. [33]

$$0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (4.2)$$

- Binarization: The grayscale image was converted to binary image by equation 4.3 and Figure 4.13 – 4.17 is show the sample of image after convert grayscale image into binary image.

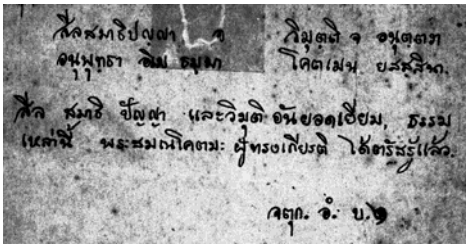
$$g(x,y) = \begin{cases} 1 ; f(x,y) \neq 0 \\ 0 ; f(x,y) = 0 \end{cases} , \quad (4.3)$$

where

$g(x,y)$ is the pixel in any point (x,y) in new image

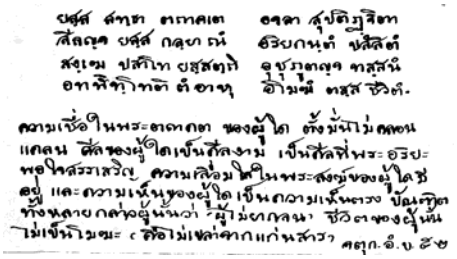
$f(x,y)$ is the pixel in any point (x,y) in original image

The pixel labeled 1 is white pixels that correspond to the background, whereas pixels labeled 0 is black pixels that correspond to the objects.



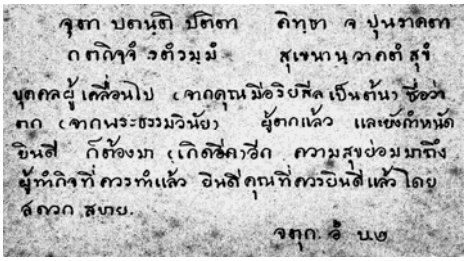
(a) Gray scale Image (b) Binary Image

Figure 4.13 An example of binary image in group 1



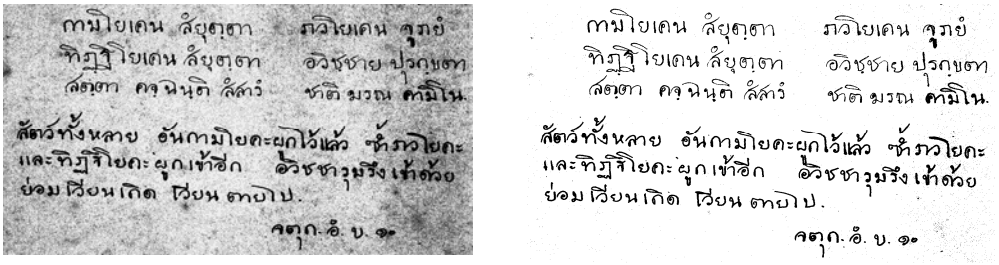
(a) Gray scale Image (b) Binary Image

Figure 4.14 An example of binary image in group 2



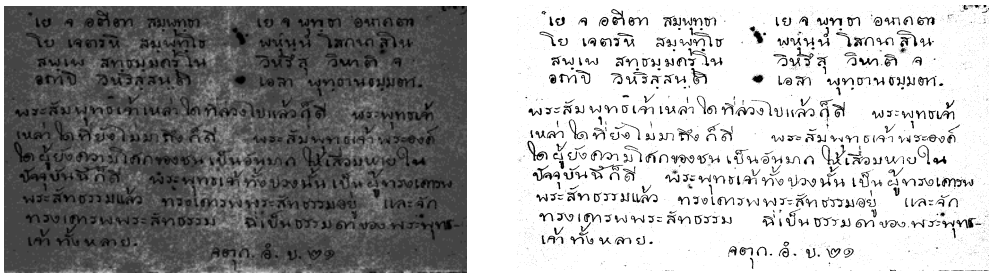
(a) Grayscale Image (b) Binary Image

Figure 4.15 An example of binary image in group 3



(a) Grayscale Image (b) Binary Image

Figure 4.16 An example of binary image in group 4



(a) Grayscale Image (b) Binary Image

Figure 4.17 An example of binary image in group 5

- Noise reduction: The objective of this section is removing noises are occurred on binary image. This research is compared 3 techniques which are Median Filter, Gaussian Filter, and Wiener Filter to find the suitable technique for reduce noise are occurred on Buddhadasa Indapanno's handwritten image after convert grayscale image into binary image. The windows sizes are used to reduce noise in 3 techniques are windows size 3×3 and 5×5.

Median filter is non-linear spatial filters. This technique is popular to reduce noise with salt-and-pepper noise bester. The sliding window is applied to a sequence from left to right and top to bottom. This research is used window 3×3 and 5×5 to test. The pixel value in the window will be sorted from minimum to maximum. The pixel value in the center after sort is a median value. The median value is replaced in the center of window. The advantages of this technique are, the isolate noises are reduced however the sharpness of image are not reduce.

Gaussian filter is low-pass filters. This technique is appropriate to reduce the Gaussian noise. The standard deviation (σ) and window size of Gaussian filter are effect to the smoothness and blur of image. This research set σ to 0.5 and Gaussian filter window are 3×3 and 5×5. The window size is increased the smoothness and blur of image are increased too. The calculation of Gaussian filter is shown in equation 4.5.

$$k = 2\pi\sigma^2 \tag{4.4}$$

$$G_{(n1,n2)} = \frac{1}{k} e^{\frac{-(n1^2+n2^2)}{\sigma^2}} \tag{4.5}$$

where k is a summation of Gaussian filters in an image. [33]

Wiener filter is linear spatial filters. This technique based on a statistical approach with minimum mean square error which, often applied in the frequency domain. The noise variance is effected to the smoothness of image. However, the Wiener filter technique is better than Median filter to reduce Gaussian noise. The calculation of Wiener filter is shown in equation 4.8.

$$\mu = \frac{1}{M \times N} \sum_{n1,n2 \in N} a_{(n1,n2)} \tag{4.6}$$

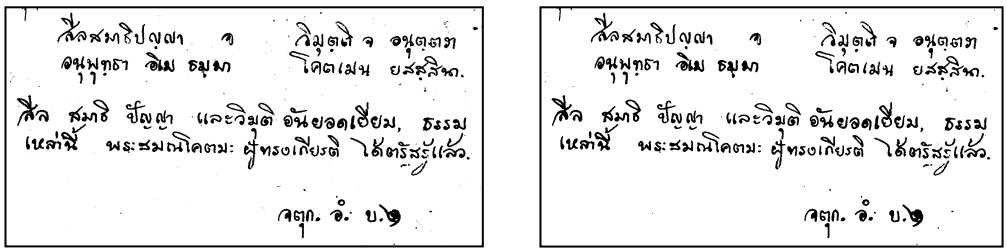
$$\sigma^2 = \frac{1}{M \times N} \sum_{n1,n2 \in N} a^2_{(n1,n2)} - \mu^2 \tag{4.7}$$

$$b_{(n1,n2)} = \mu + \frac{(\sigma^2 - v^2)(a_{(n1,n2)})}{\sigma^2} \tag{4.8}$$

when $b_{(n1,n2)}$ is a result of an interested pixel, and $a_{(n1,n2)}$ is an interested pixel on a noised image, μ is local mean, σ^2 is variance around an interested pixel, and v^2 is the noise variance. An average of all the local estimated variances can be used instead of noise variance in case of it is not available. [33]

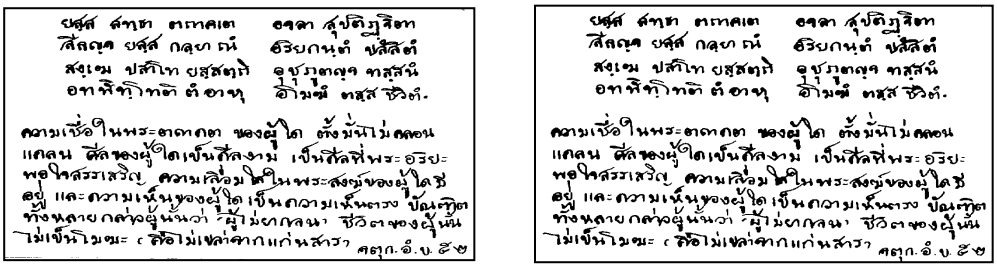
Due to the noises are occurred in the binary image is salt-and-pepper noise, this research will be using Median filter to remove noise.

The samples of images are removed noise with window 3x3 of Median Filter. The sample images are shown in Figure 4.18 – 4.22. The results of image filtering are explained in Chapter V.



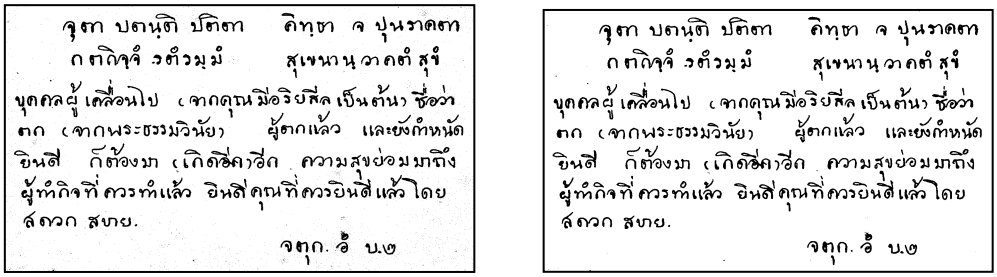
(a) Gray scale Image (b) Binary Image

Figure 4.18 An example of denoised image in group 1



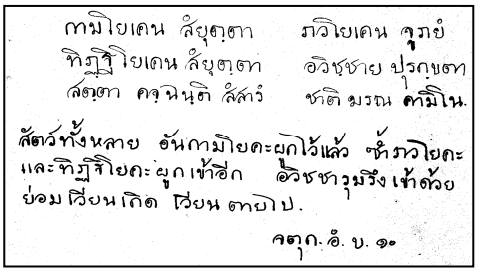
(a) Gray scale Image (b) Binary Image

Figure 4.19 An example of denoised image in group 2

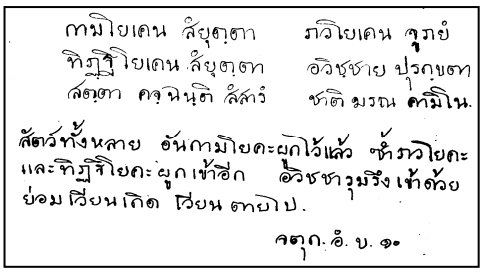


(a) Gray scale Image (b) Binary Image

Figure 4.20 An example of denoised image in group 3

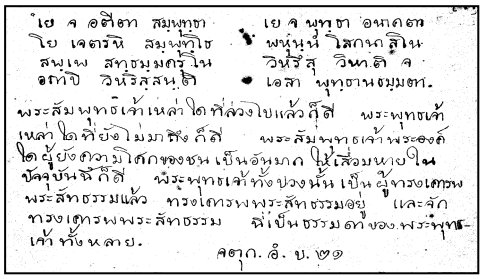


(a) Gray scale Image

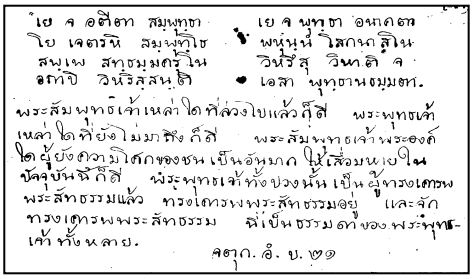


(b) Binary Image

Figure 4.21 An example of denoised image in group 4



(a) Gray scale Image



(b) Binary Image

Figure 4.22 An example of denoised image in group 5

4.2 Character Segmentation

The character segmentation a process is applied the Connected Component Labeling and Projection Profile to divide the handwritten character image into individual character. The outputs of this process are single characters. These characters are data inputs of feature extraction processes. The flowchart of overall steps of this process is shown in Figure 4.23.

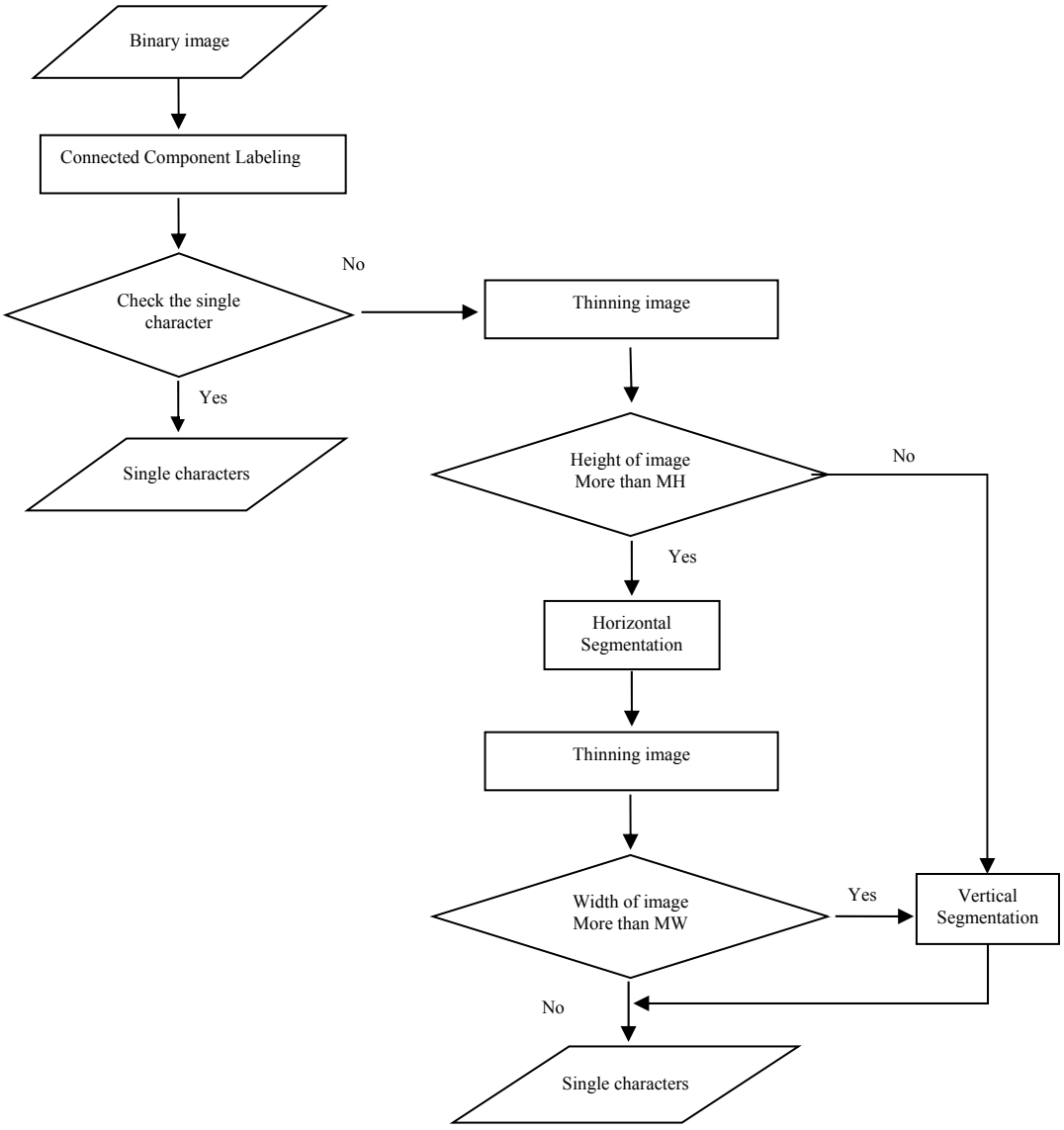


Figure 4.23 The flowchart of character segmentation process

4.2.1 Connected Component Labeling

This method can be used for separating the characters out of the background. This research defines the connectivity of the neighborhood is 4-connected. The single characters in the image are divided, but the touching character cannot be segmented in this method. In MATLAB, the syntax of Connected Component Labeling is shown in equation 4.9. [12]

$$[L, num] = bwlabel(f, conn) \quad (4.9)$$

where , f is input binary image, $conn$ is specifies the desired connectivity 4 or 8 this research uses 4-connected ,L is the output, there are called a label matrix, and num is the optional, give the total number of connected component found

when the image is segmented with Connected Component Labeling, there are single characters, and touching characters. However, some images have single characters, touching characters and non-characters (spot and border line of image). The sample results of this process are shown in Figure 4.24 and 4.25.

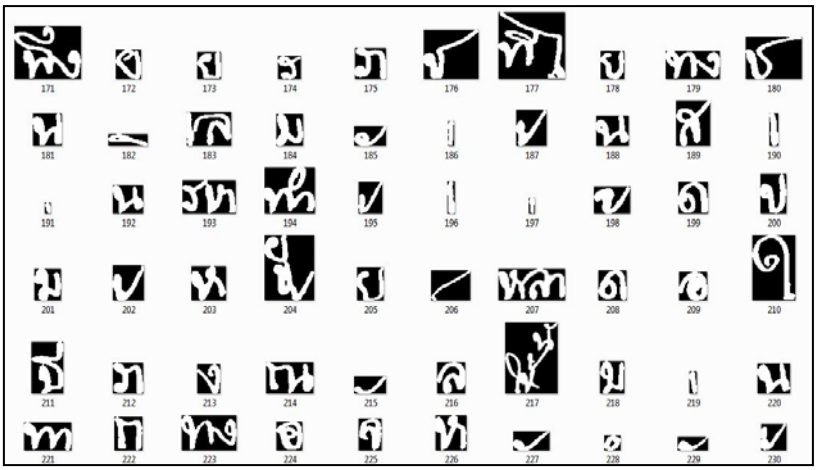


Figure 4.24 The samples result of character have single characters and touching character

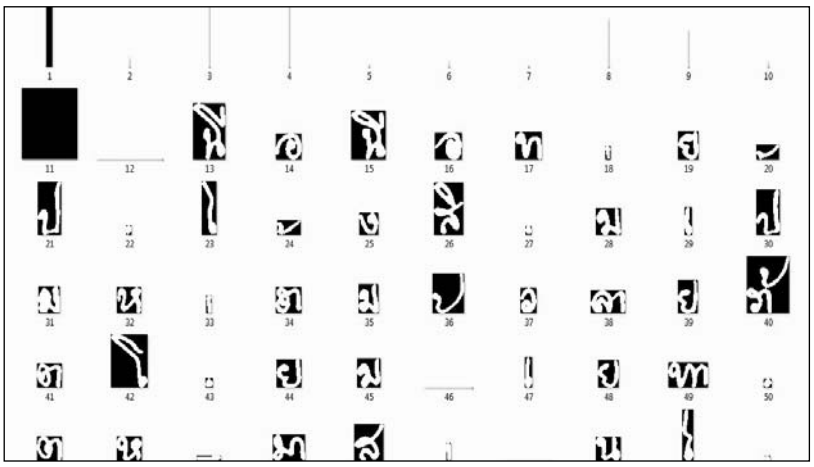


Figure 4.25 The samples result of character have single characters and touching character and not character

Figure 4.24 and 4.25 show the sample of segmented character images with the Connected Component Labeling. The results in Figure 4.24 are single characters and touching characters that are not non-character. The results in Figure 4.25 show single characters, touching characters, and non-characters. The characters 1 – 10 in Figure 4.25 are the non-characters that occurred after segment the character in this process.

After segmented with Connected Component Labeling, there are 2 sub-processes to separate non-character out of the character group and separate single characters out of touching characters.

Separation of non-character: The criterion to separate non-character out of character group are used the height (h) and width (w) of character and proportion of white (pw) and black (pb) pixel. The criterion to separate the non-characters is shown in Algorithm 4.1 and the sample of separated non- characters are showed in Figure 4.26.

Algorithm 4.1 The criterion to separate non-characters

if ((h < 10) and (w ≤ 10)) or
((pw < 6) and (pb >93)) or
((pw ≤ 10) and (pw >90)) or
((abs(h-w)) > 130)
then noises are removed

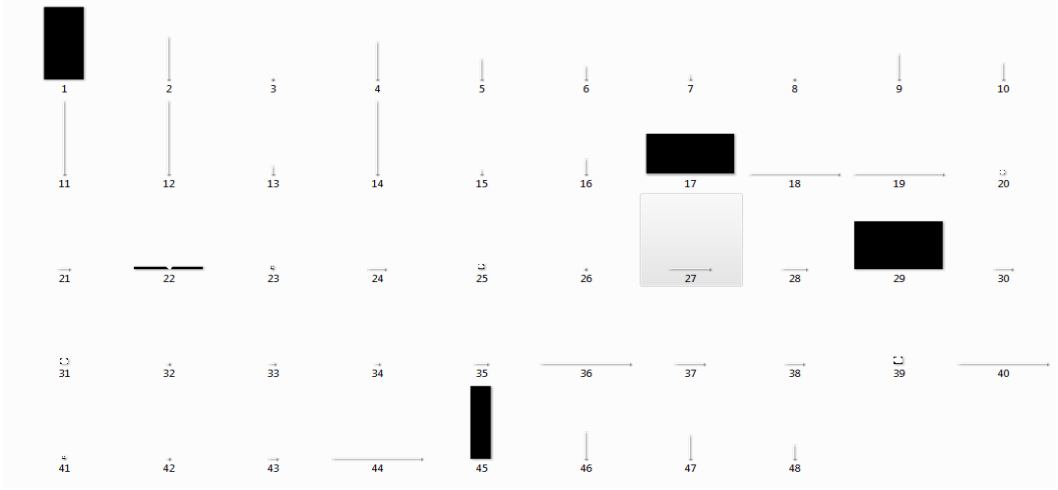


Figure 4.26 Separated non-characters

- Separation of single character: The single characters are separated from touching character. The criterion are used:
    - height (h) of a character
    - width (w) of a character
    - proportion of white (pw) and black (pb) pixel
    - mean of height of characters in each page (MH)
    - mean of width of characters in each page (MW)
    - mean of white pixel in horizontal direction (wh)
    - mean of white pixel in vertical direction (wv)

The criterion to separate the single character from touching character is divided to 4 criterions. These criterions are shown in Algorithm 4.2 – 4.5.

Algorithm 4.2 Criterion 1

```
if (MH < 30)
   if ((h ≤ 30) and (w ≤ 24)) or ((h ≤ 15) and (w ≤ 30))
      single character
   else
      touching character
   end
end
```

The sample results of single character from criterion 1 are shown Figure 4.27, and the samples of touching characters are shown in Figure 4.28.

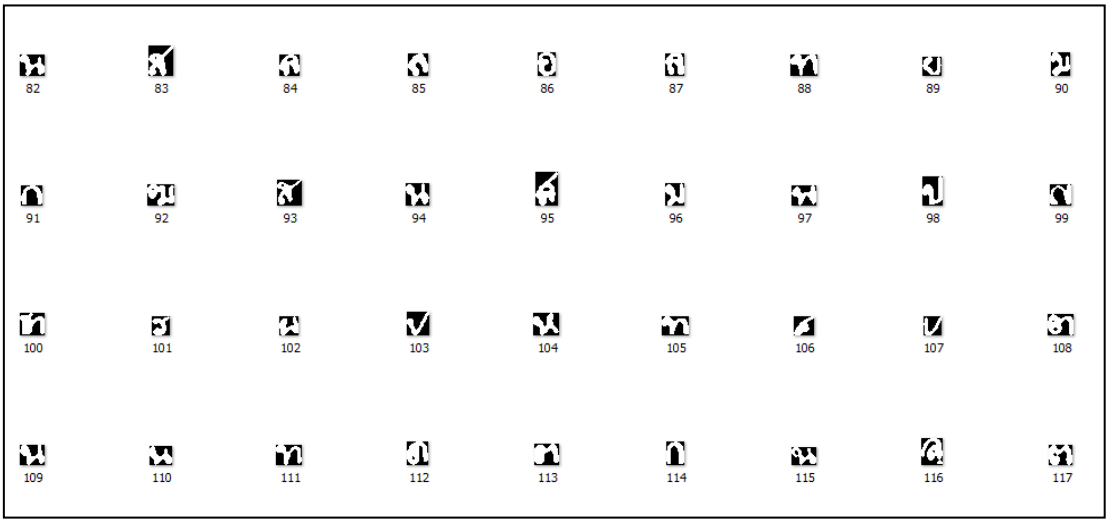


Figure 4.27 Samples of single character from criterion 1

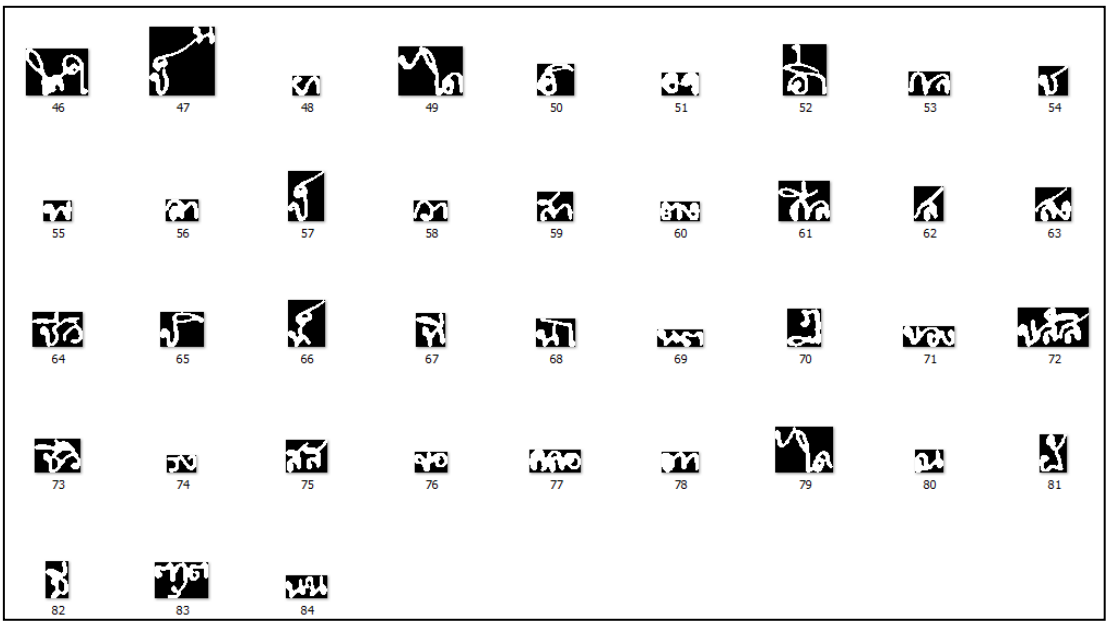


Figure 4.28 Samples of touching character from criterion 1

Algorithm 4.3 Criterion 2

if (MH ≥ 30) and (MH < 40)

    if ((h ≤ 53) and (w ≤ 43)) or ((h ≤ 25) and (w < 25))

        single character

        else

            if ((h < 30) and (w < 45)) or ((h< 30) and (pw < 45))

                single character

```
                else
                    if  ((h ≥ 30) and (h < 40)and (pw< 30) and (pw < 40) and (w ≤ 50))
                    or ((h≥30 and h < 40) and (pw <35) and (w ≤ 60))
                            single character
                        else
                            touching character
                    end
            end
        end
end
```

The sample results of single character from criterion 2 are shown Figure 4.29, and the samples of touching characters are shown in Figure 4.30.

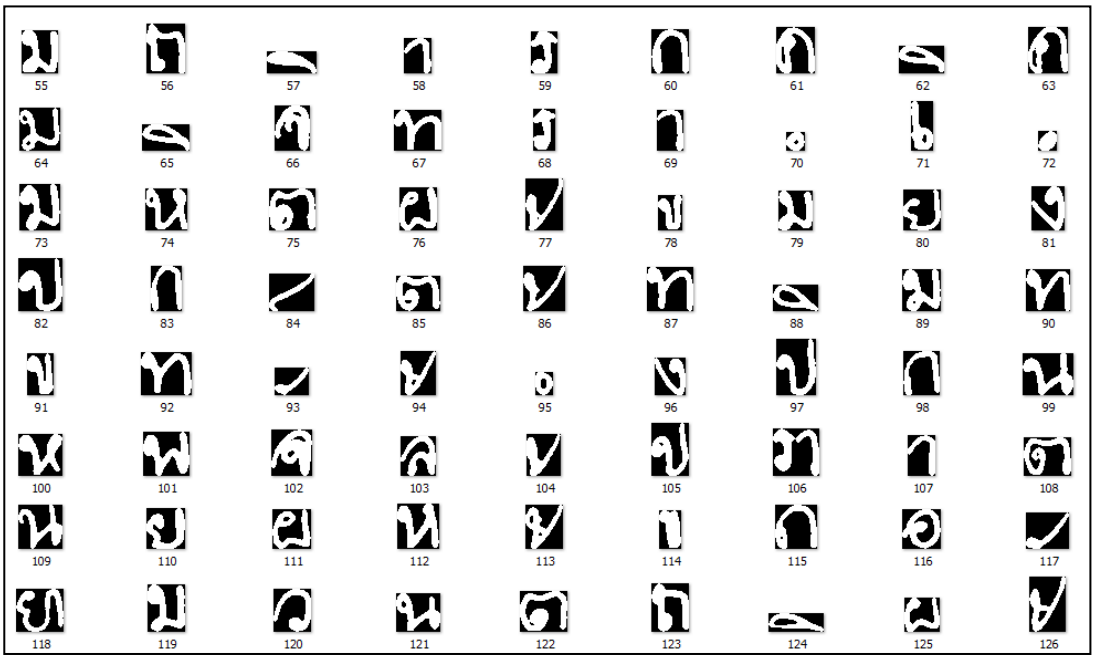


Figure 4.29 Samples of single character from criterion 2

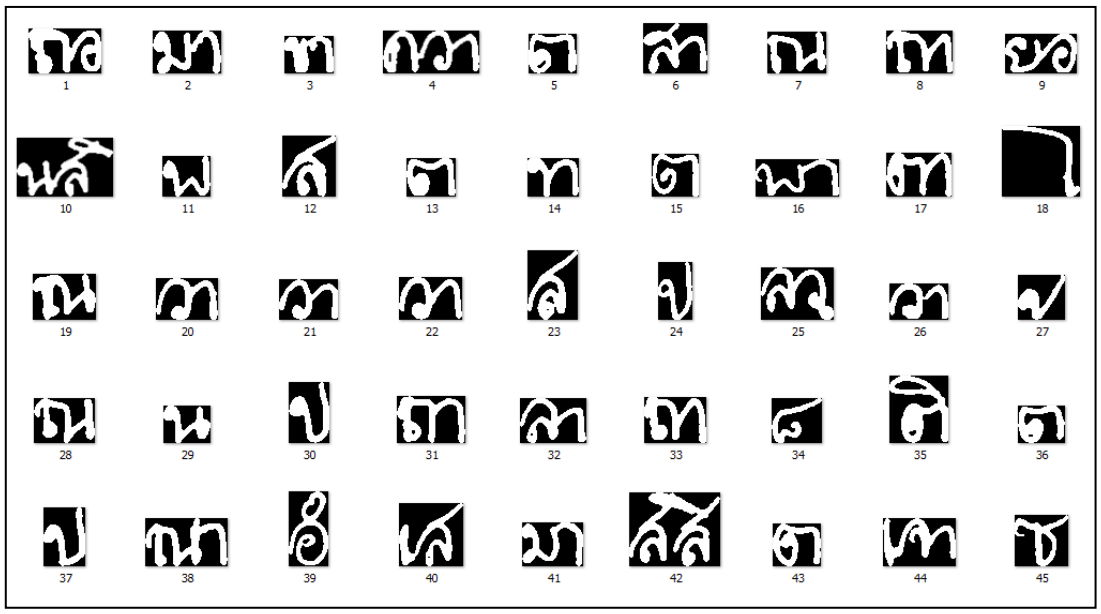


Figure 4.30 Samples of touching character from criterion 2

Algorithm 4.4 Criterion 3

```
if (MH ≥ 40)  and (MH < 50)
    if ((MH ≤ 45) and (MW ≤ 45))
        if (h ≤ 50 and w < 43)
            single character
                else
                    if ((h < 30 and  w <56)) or ((wh ≤ 15) and (wv ≤ 15))
                        single character
                            else
                                if ((w ≤ 50) and (wh ≤ 25) and (wv ≤ 25 ))
                                    single character
                                else
                                    touching character
                                end
                    end
        end
    else if ((MH ≤ 45) and (MW ≥ 45))
            if (h ≤ 55 and w ≤ 48)
```

```
            single character
                else
                    if  ((h < 30 and  w <56)) or ((wh ≤ 15) and (wv ≤ 15))
                        single character
                            else
                                if  ((w ≤ 50) and (wh ≤ 25) and (wv ≤ 25 ))
                                        single character
                                    else
                                        touching character
                                end
                    end
        end
    else if (MH > 45)
        if ((h ≤ 60) and (w < 44))
            single character
                else
                    if ((h < 30 and  w <56)) or ((wh ≤ 15) and (wv ≤ 15))
                        single character
                              else
                                 if  ((w ≤ 50) and (wh ≤ 25) and (wv ≤ 25 ))
                                        single character
                                   else
                                        touching character
                                 end
                    end
        end
    end
end
```

The sample results of single character from criterion 3 are shown Figure 4.31, and the samples of touching characters are shown in Figure 4.32.

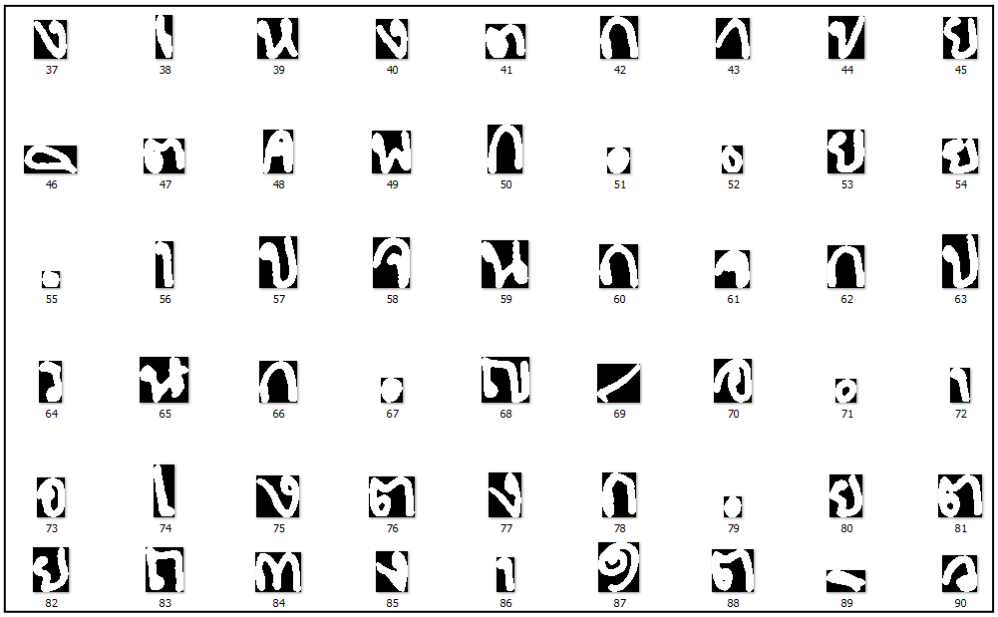


Figure 4.31 Samples of single character from criterion 3

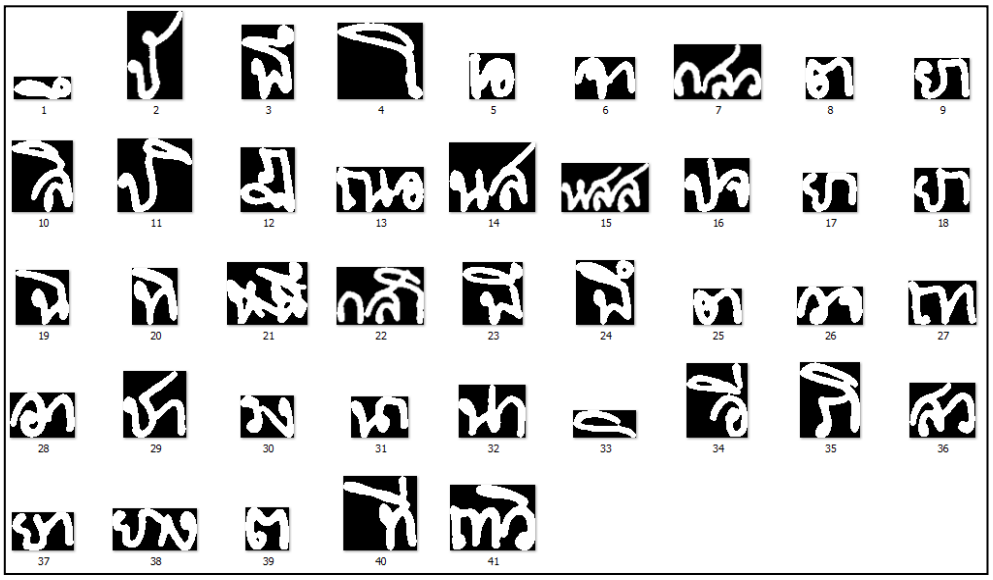


Figure 4.32 Samples of touching character from criterion 3

Algorithm 4.5 Criterion 4

```
if (MH ≥ 50)
    if (MW < 55)
        if (h ≤ 60 and w ≤ 45)
            single character
        else
```

```
            if ((h < 90) and (wh ≤ 25) and (wv ≤ 25)) or ((w < 90) and (wh ≤ 25) and
               (wv ≤ 25))
               single character
             else
               if ((h ≥ 90) and (wh ≤ 20) and (wv ≤ 20)) or ((w ≥ 90) and (wh ≤ 20)
                  and (wv ≤ 20))
                    single character
                else
                    touching character
               end
            end
        end

    else if (MW ≥ 55))
        if (h <80 and w ≤ 57)
           single character
         else
            if ((h < 90) and (wh ≤ 25) and (wv ≤ 25)) or ((w < 90) and (wh ≤ 25) and
               (wv ≤ 25))
                single character
             else
               if ((h ≥ 90) and (wh ≤ 20) and (wv ≤ 20)) or ((w ≥ 90) and (wh ≤ 20)
                  and (wv ≤ 20))
                  single character
               else
                  touching character
               end
            end
        end
    end
end
```

The sample results of single character from criterion 4 are shown Figure 4.33, and the samples of touching characters are shown in Figure 4.34.

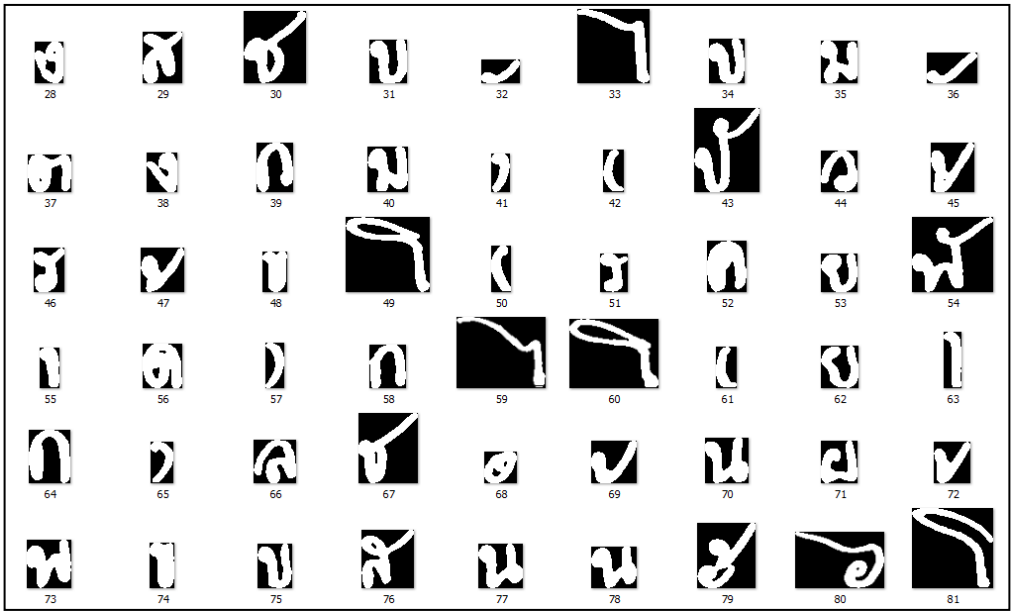


Figure 4.33 Samples of single character from criterion 4

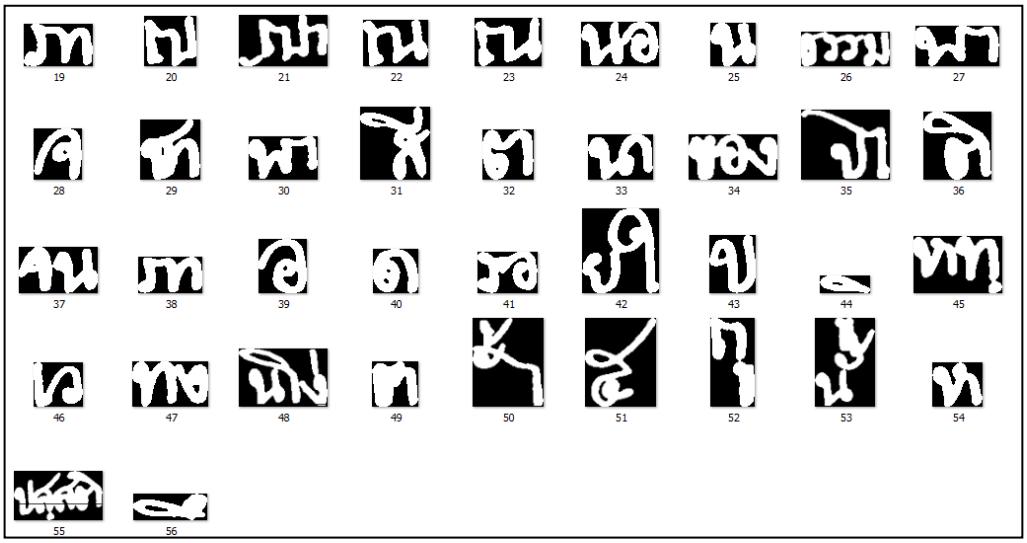


Figure 4.34 Samples of touching character from criterion 4

4.2.2 Projection profile

The touching characters are separated into single character images. The input images in this process will be thinned before projection. The character images after thinning process have the clearly cutting points. This each image will be checked

their width and height to re-segment again. Then, horizontal projection, vertical projection, or both horizontal and vertical projection will be selected for the touching characters. The algorithm to segment the touching characters with projection profile is showed in Algorithm 4.6.

Algorithm 4.6 Projection algorithm

1. Compare between height of character (*h*) and mean of single character's height (MH).

   - *If the height of thin character less than MH, the thin character will be segmented in vertical projection only. The most of results in this stage are single character or characters are touched in horizontal direction.*

   - *If the height of thin characters is more than MH, the thin character will be segmented in horizontal projection. The result of this stage will be checked. (there are both single and touching character)*

2. The results of horizontal projection will be thinned.

3. check the width of thin character with MW (mean of single character's width)

   - *If the width of thin character less than MW. The most of results in this stage are single character.*

   - *If the width of thin characters is more than MW, the character will be segmented in vertical projection. (The result of this stage will be single character)*

The cutting points are considered from summation of thinned image in horizontal direction. Summation of touching point is mostly equaled 1. The positions have summation is 1 are selected to the cutting point. The criterions of cutting point are used:

- height (h) of a character
- width (w) of a character
- position of sum equal 1 (position)

The sample image of vertical projection, horizontal projection, and both horizontal and vertical projection are showed in Figure 4.35, 4.36, and 4.37, respectively.

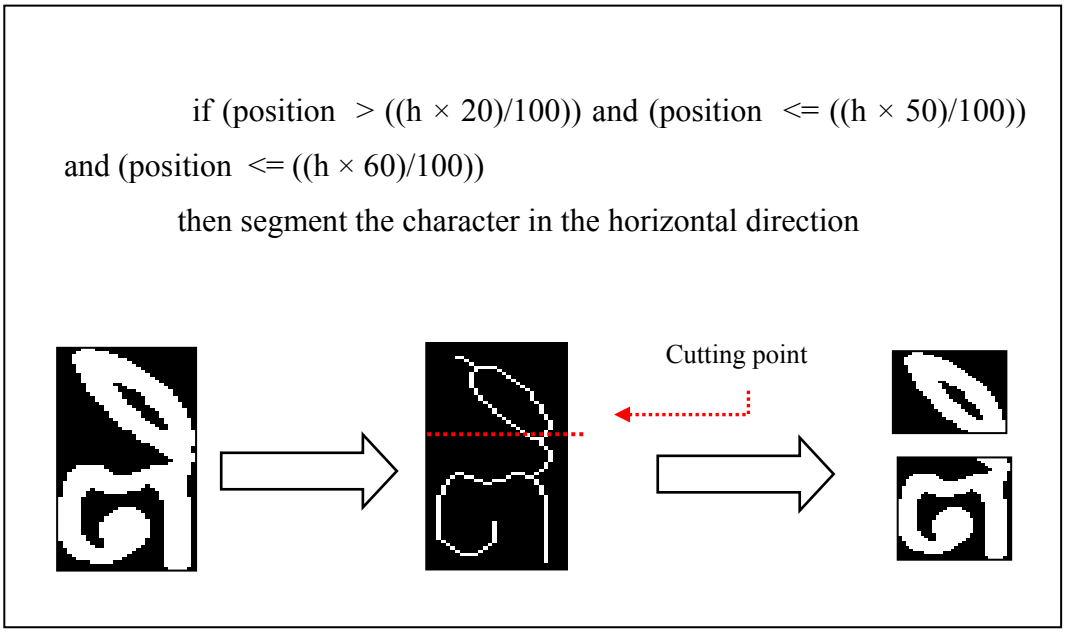


Figure 4.35 Segmentation in the horizontal direction

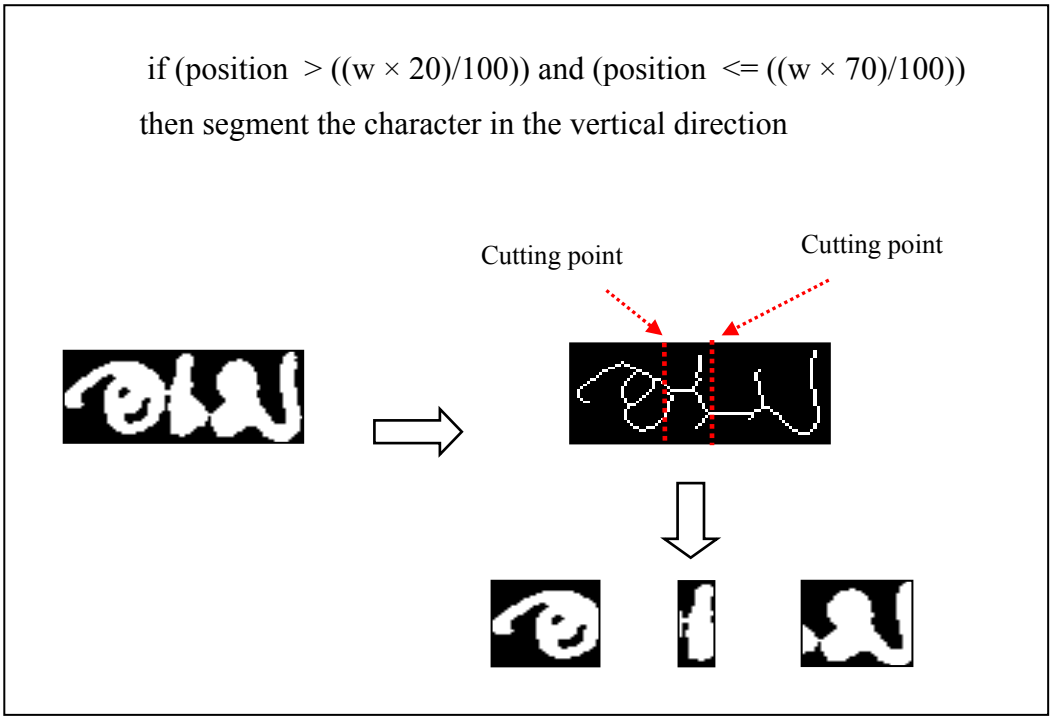


Figure 4.36 Segmentation in the vertical direction

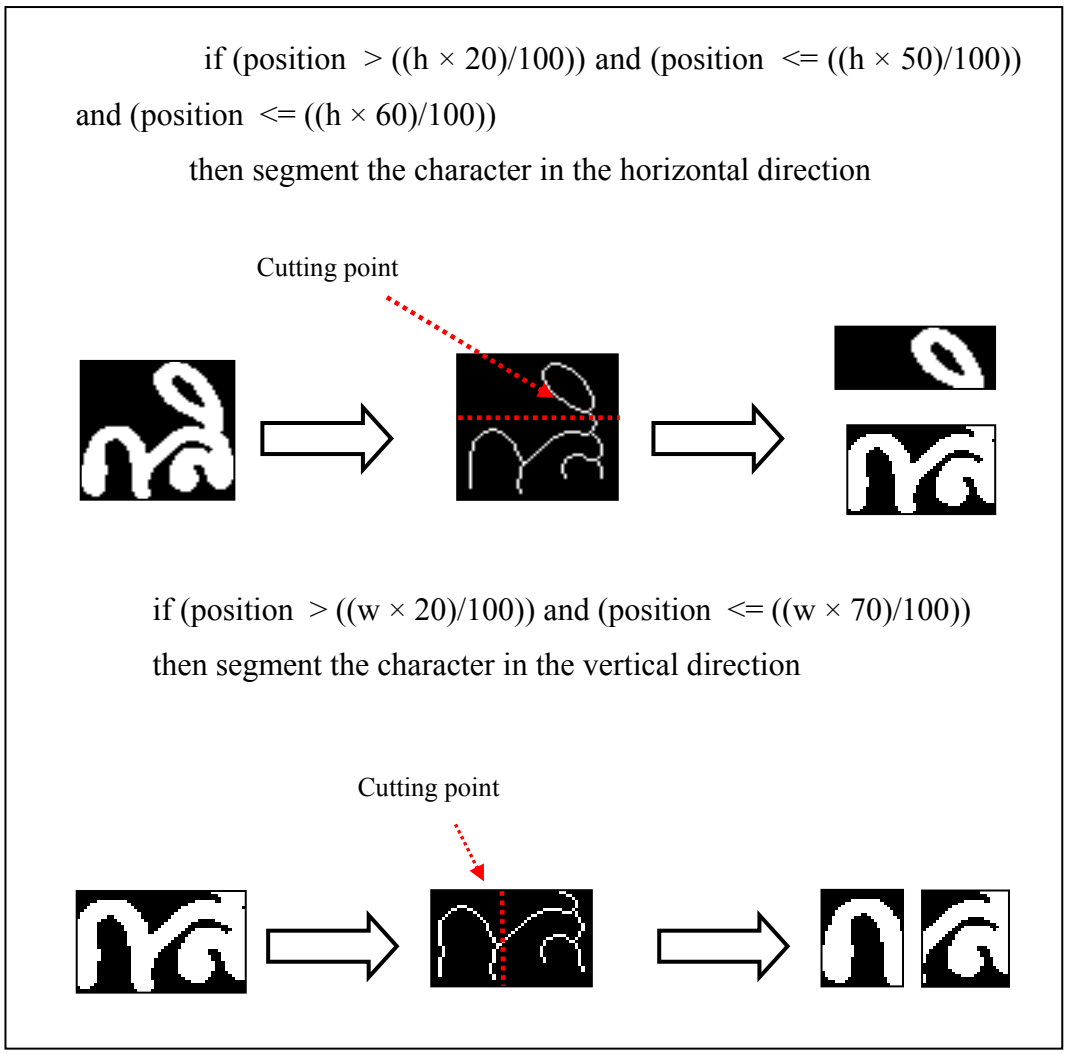


Figure 4.37 The image is segmented in the horizontal and vertical direction

4.2.3 Line Position Detection

Find the text line of image: The image after removed noise will be used to find the text line. The vertical projection is used to find the number of text line in image. The dash line is the point to separate the text lines that are minimal of black pixel. The algorithm to separate the text line is shown in Algorithm 4.7 and sample of image is separated the text line with horizontal direction is shown in Figure 4.38. The position of text line is used to sort the character after character recognition process.

Algorithm 4.7 Line Detection Position algorithm

1. Summation of image's pixel in horizontal direction (SUM)
2. if (SUM == 0)

   then there is the points to separate the text line (TL)

   Check the height (h) of text line

   if (h ≥18)

   then there is a text line
3. Find the sub-line (SL)

   if (SUM > 100) and (h > 15)

   then the text line is separated to sub-line

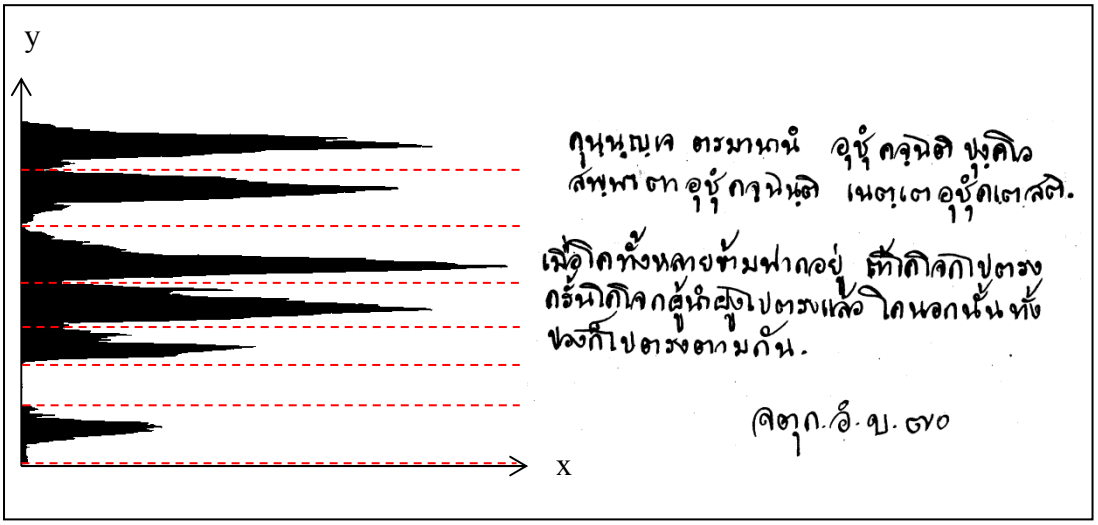
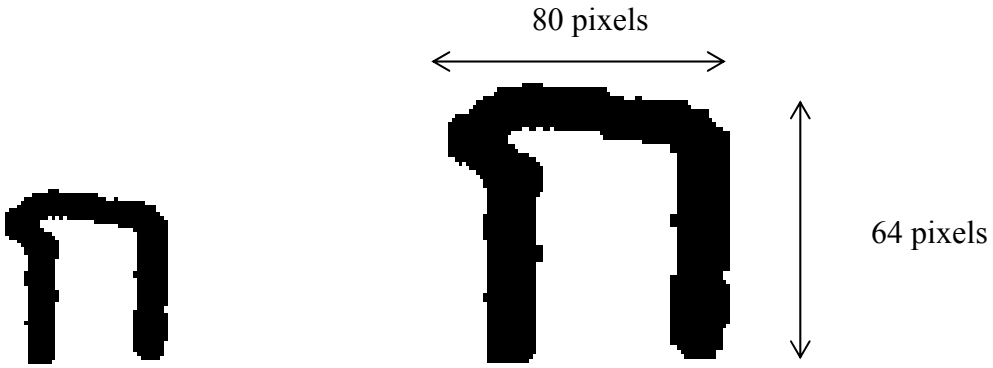


Figure 4.38 The text lines are separated by vertical projection

4.3 Feature Extraction

The character images in the character segmentation process are input data of this step. The important features of each character are extracted with zoning method. Before feature extraction, size of each character image will be adjusted into 80 x 64 pixels, and the zone of character images were divided to 16 (4 x 4) pixels per zone. Each zone of character image is average the density of black pixels. The sample of average value in each zone is shown in Figure 4.39.



Fixture 4.39 The image resize to 80 x 64 pixels

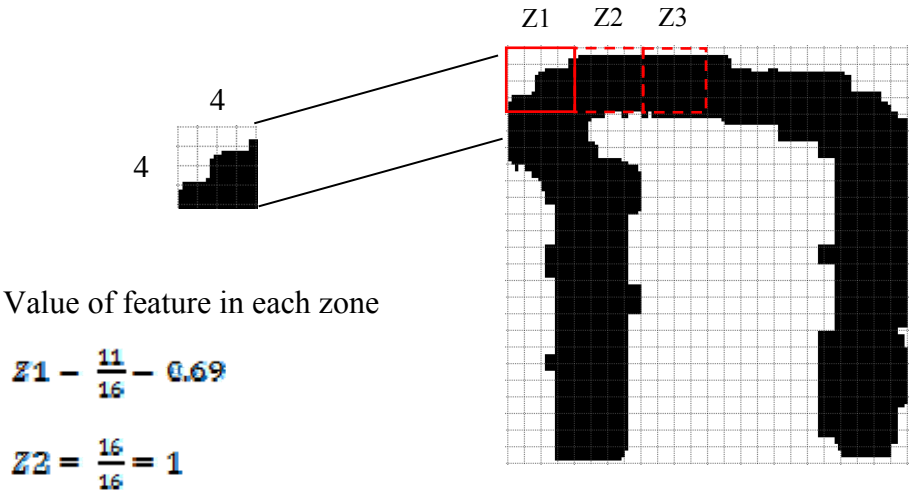
The number of features in each image is input to character recognition process .Therefore, the number of feature in each character image in this research is 320 features. The calculation is shown in equation 4.10.

$$\text{Number of features} = \left(\frac{\text{number of pixels in resized image}}{\text{number of pixels in zone}}\right) \quad (4.10)$$

$$= \left(\frac{80 \times 64}{16}\right)$$

$$= 320$$

The value of each zone is calculated the mean of black pixel from right to left and top to bottom. The samples of feature values in each zone are shown in Figure 4.40.



Value of feature in each zone

$$Z1 = \frac{11}{16} = 0.69$$

$$Z2 = \frac{16}{16} = 1$$

$$Z3 = \frac{16}{16} = 1$$

Result after extract the feature image

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11		Z320
0.69	1	1	0.88	0.75	0.5	1	0.75	0.25	0.25	0.25	...	1

Figure 4.40 The sample of value after extracted with 4 x 4 pixels per zone

The advantage of zoning method that is not complex. This method is using the short time to process. However, the size of zone is effect to the time to training Neural Network model and correctness of character recognition.

4.4 Character Recognition

In this step, the important features of each character are used as input data to recognize that character. This research using feed-forward back-propagation neural network, due to the Artificial Neural Network (ANN) is a computational model extensively to use in pattern recognition. The ANN model can solve all problems in automatic reasoning including a pattern recognition problem. However, the over fitting will be occurred if the node of hidden layers are large. Whereas, the nodes of hidden layers are small, that cannot learn the problem [34], [35].

The totality of characters are used to train the neural model is 28,584 character. The classes of characters are classified to 87 classes as the 42 consonants, 17 vowels, 3 tones, 11 numbers, 13 symbols, and unknown class. The structure of neural network in this research consists of input layer, hidden layer, and output layer. The number of input node is 320 that are equaled to the number of features. Each hidden node will calculate the obtained value using Tan Sigmoid function, and number of nodes in hidden layer is derived from number of features and number of output nodes (320+87). The output layer has 87 nodes, each node will calculate the obtain value using Log Sigmoid function.

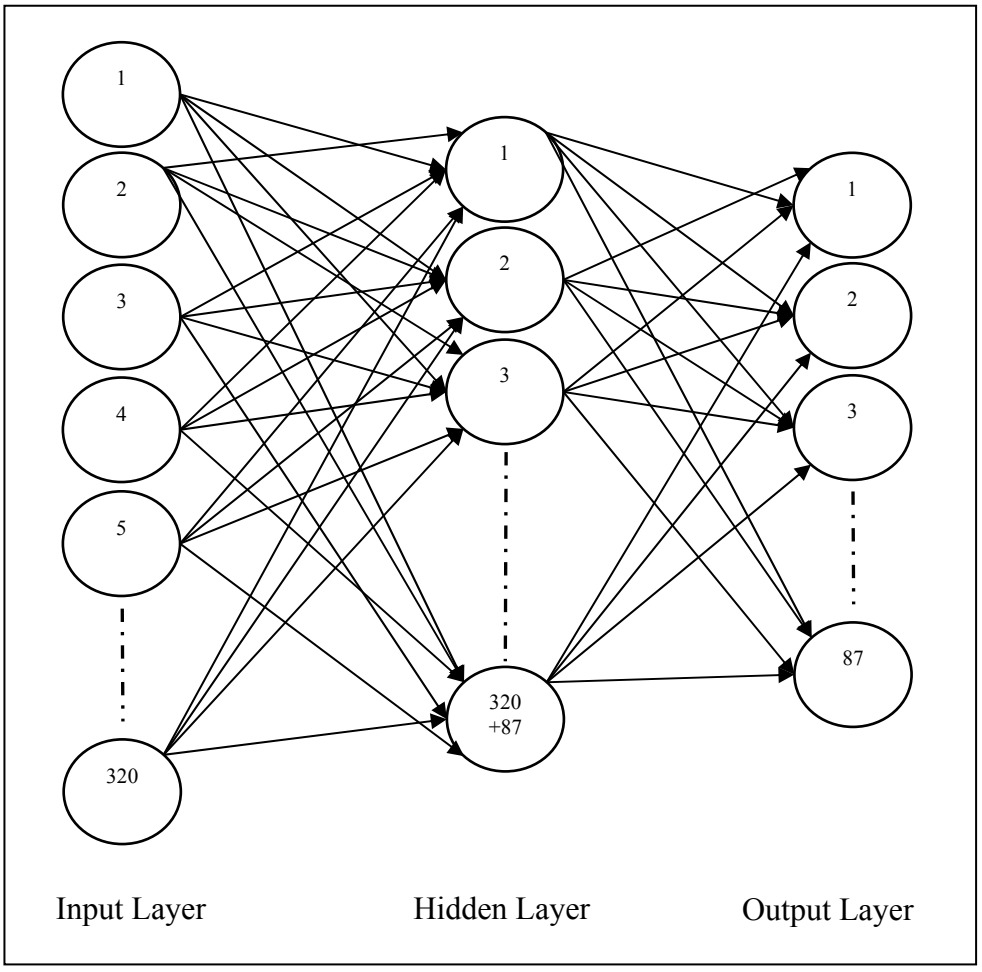


Figure 4.41 The neural network architecture of Buddhadasa Indapanno's handwritten

4.4.1 Activation Function [36]

The activation function is used to compute the hidden and output nodes of the neural network system. The sigmoid activation function is used to supports in this network architecture. Due to, this function is particularly advantageous for use in neural networks trained by back-propagation, because sigmoid is easy to differentiate, and thus can dramatically reduce the computation burden for training. The sigmoid function is applied to applications whose desired output values are between 0 and 1 [37]. The sigmoid function is shown in equation 4.11 and 4.12. The sigmoid curve is shown Figure 4.42 and 4.43.

Tan Sigmoid function

$$S(x) = \tanh(cx) = \frac{e^{cx} - e^{-cx}}{e^{cx} + e^{-cx}}, \quad (4.11)$$

where c > 0 , c = const, is a positive scaling constant.

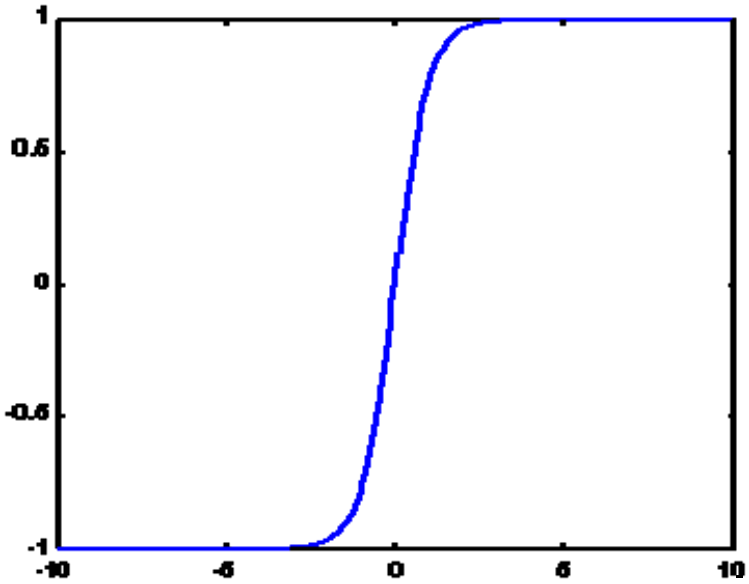


Figure 4.42 Tan Sigmoid curve

Log Sigmoid function

$$S(x) = \frac{1}{1+e^{-cn}}, \quad (4.12)$$

where c > 0 , c = const, is a positive scaling constant.

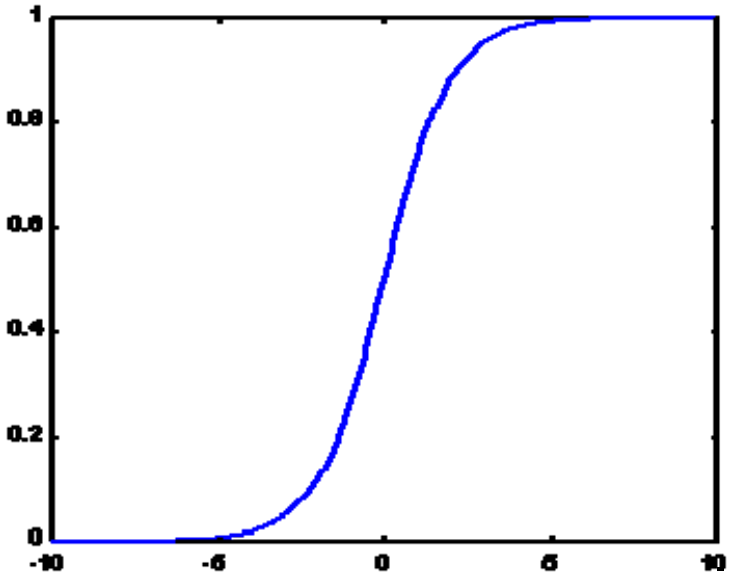


Figure 4.43 Log Sigmoid curve

4.4.2 Training Function

The network training algorithm in this research using Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation (GDX), the function in MATLAB used traingdx. The traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate [33]. The parameter of traingdx function is shown in Table 4.3.

Table 4.3 The default parameter of train the neural network model

Training Parameters	Default Value	Description
net.trainParam.epochs	10	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.lr	0.01	Learning rate
net.trainParam.lr_inc	1.05	Ratio to increase learning rate
net.trainParam.lr_dec	0.7	Ratio to decrease learning rate
net.trainParam.max_fail	5	Maximum validation failures
net.trainParam.max_perf_inc	1.04	Maximum performance increase
net.trainParam.mc	0.9	Momentum constant.
net.trainParam.min_grad	1E-10	Minimum performance gradient
net.trainParam.show	25	Epochs between displays (NaN for no displays)
net.trainParam.time	inf	Maximum time to train in seconds

To find the suitable model of neural network, this research will compared the neural model by compared the maximum number of epochs to train in 1000, 3,000, 5000, and 10000 with 10 fold cross validation. After that these data will be used for creating a model. This model will be evaluate in order to get the learn accuracy.

CHAPTER V
EXPERIMENTAL RESULTS

The results of Buddhadasa Indapanno's handwritten character recognition system are described in this chapter. The system is developed with MATLAB version R2010a. The images are used in this research is the Pali card, there are148 pages. The characteristic of images are yellow-brownish background and the color of characters is green and blue. The resolutions of these images are 300 dpi and stored in TIFF file.

5.1 The Result of Pre-processing

The handwritten images of Buddhadasa Indapanno are classified in 5 groups followed to the mean of pixel image after cropped. The mostly of image in each groups will have the mean of pixel image nearby and the characteristic of image are similar. The characteristic of image in five groups will be described following:

Group 1: The images in this group will have smear on the background and there is lower brightness contrast.

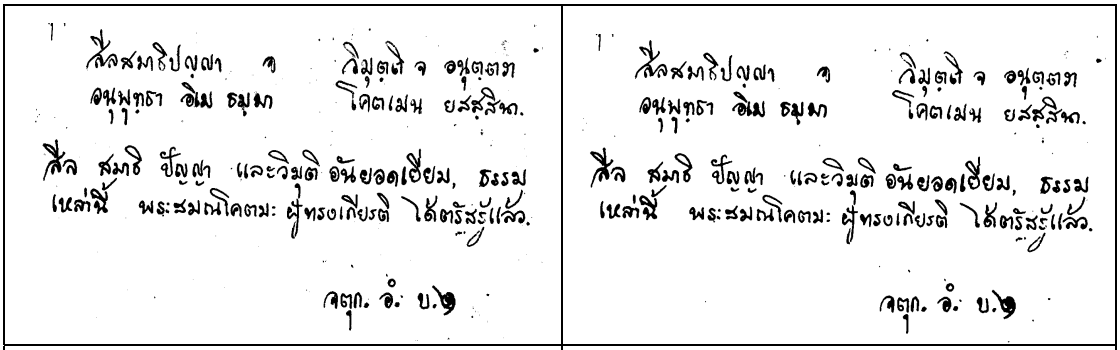
Group 2: The images in this group are consisted the small and thick character but the brightness contrast of image are normally.

Group 3: The mostly of size and thickness character in this image group are suitable.

Group 4: The characters in this image group consisted the characters are thick, thin, and fade together.

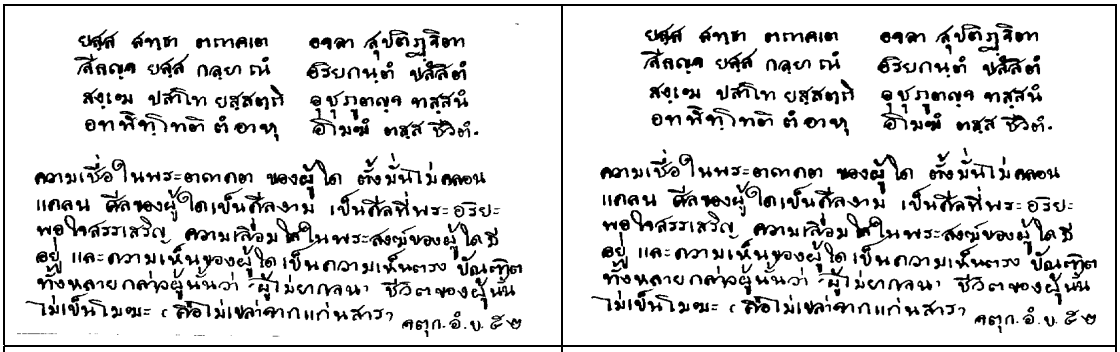
Group 5: The characters in this image group are thin and fade.

In Binarization process, the noises are occurred on the binary image. This research is compared the method to remove noise with Median Filter, Gaussian Filter, and Wiener Filter with window 3x3 and 5x5. The results of each method are shown in Figure 5.1 – 5.30.



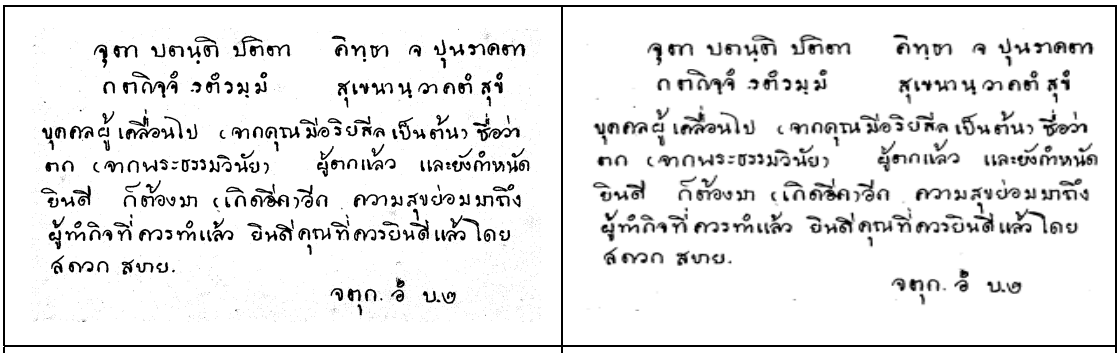
(a) Binary image (b) Sample result

Figure 5.1 Median Filter 3x3 in Group 1



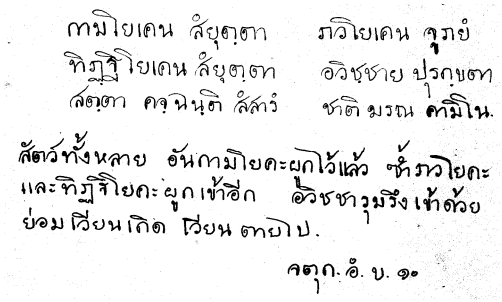
(a) Binary image (b) Sample result

Figure 5.2 Median Filter 3x3 in Group 2

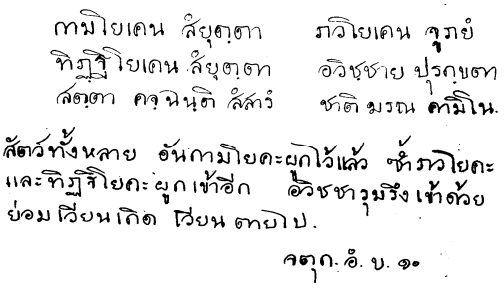


(a) Binary image (b) Sample result

Figure 5.3 Median Filter 3x3 in Group 3

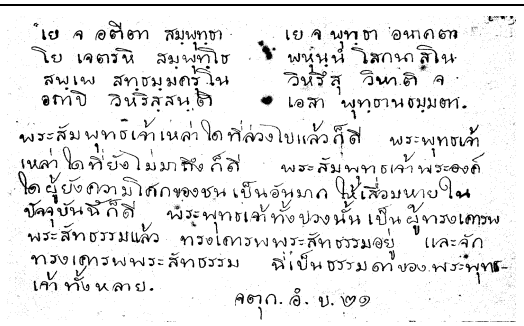


(a) Binary image

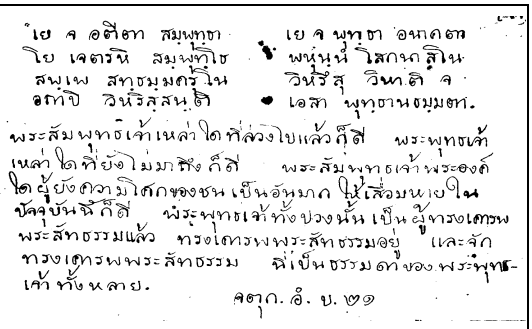


(b) Sample result

Figure 5.4 Median Filter 3x3 in Group 4



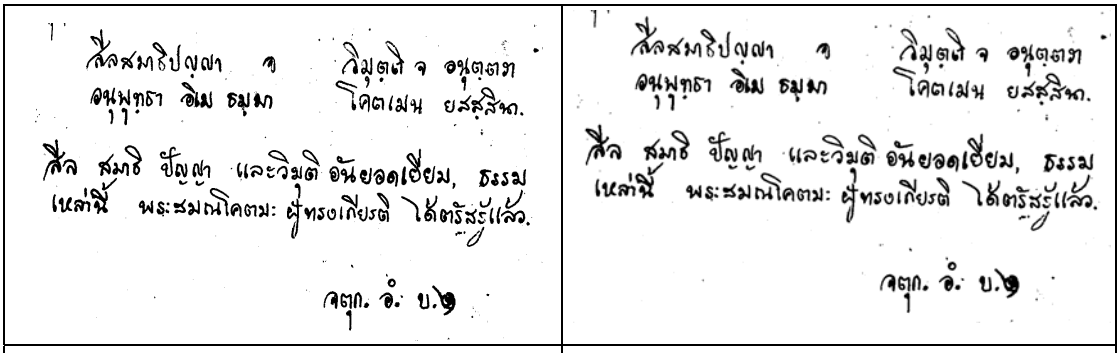
(a) Binary image



(b) Sample result

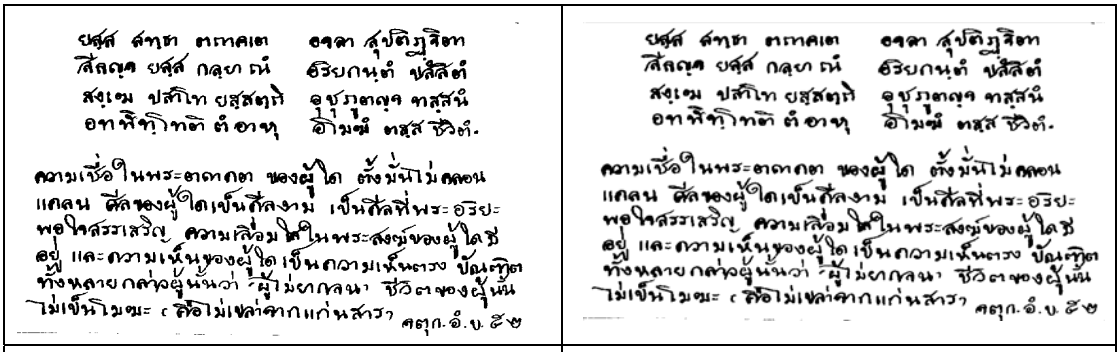
Figure 5.5 Median Filter 3x3 in Group 5

The Median Filter with window size 3x3 can remove the noise in these images. Moreover, the characters are not destroyed. For, the binary images in group 5, the mostly of noises are removed but the noise in the corner are remained.



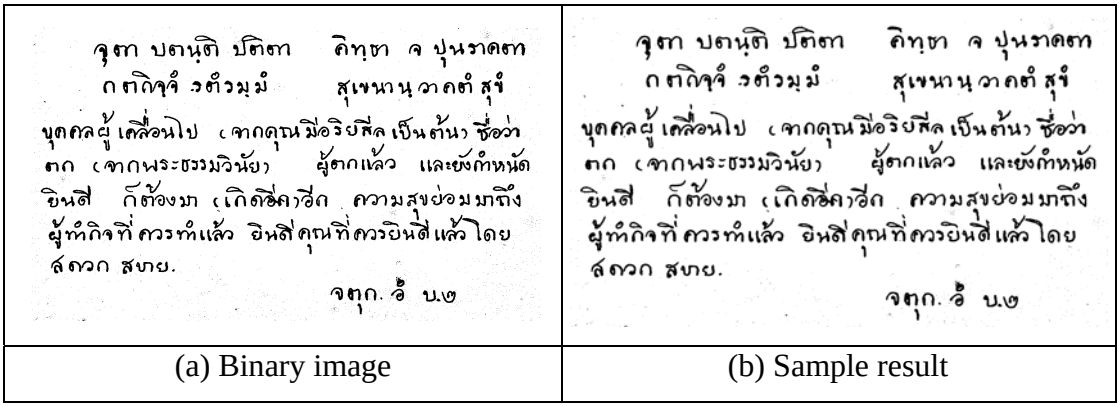
(a) Binary image (b) Sample result

Figure 5.6 Gaussian Filter 3x3 in Group 1



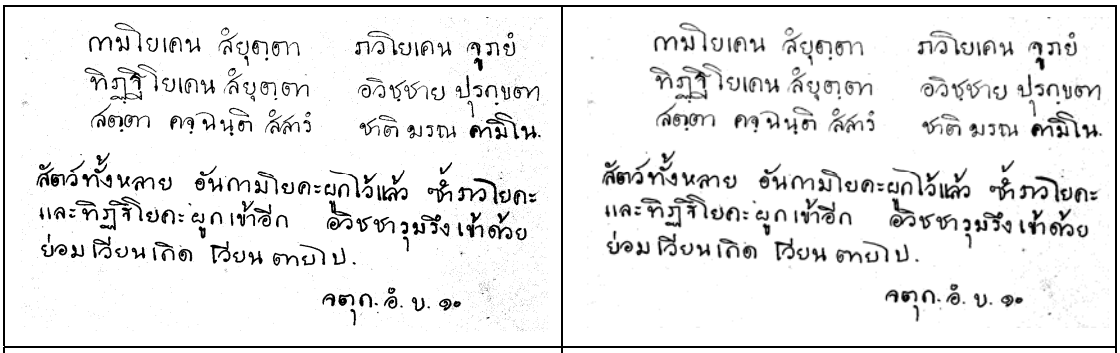
(a) Binary image (b) Sample result

Figure 5.7 Gaussian Filter 3x3 in Group 2



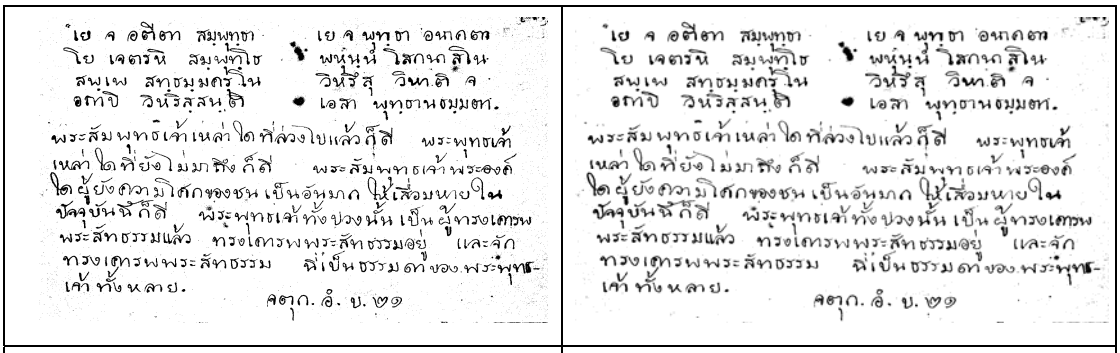
(a) Binary image (b) Sample result

Figure 5.8 Gaussian Filter 3x3 in Group 3



(a) Binary image (b) Sample result

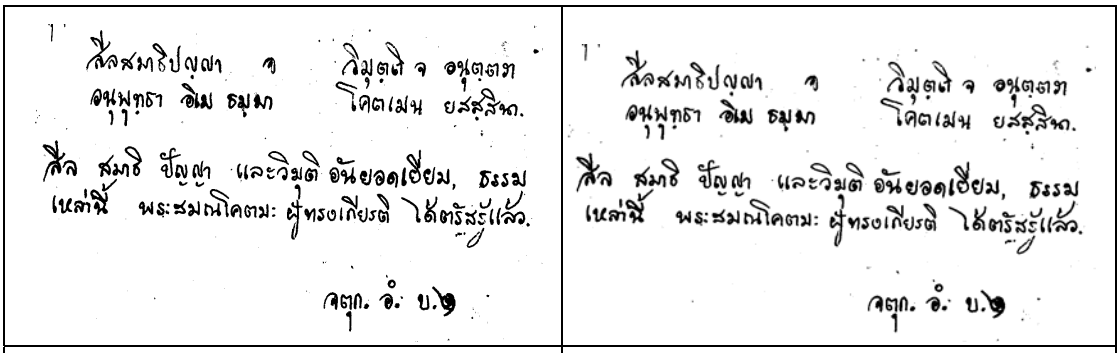
Figure 5.9 Gaussian Filter 3x3 in Group 4



(a) Binary image (b) Sample result

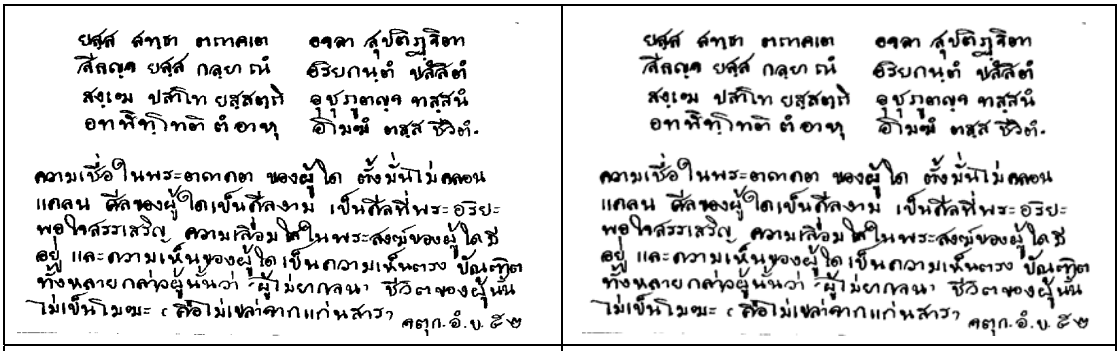
Figure 5.10 Gaussian Filter 3x3 in Group 5

When the Gaussian Filter with window size 3×3 is applied, the noises are not remained. The results are very similar to the binary images as shown in Figure 5.6 – 5.10.



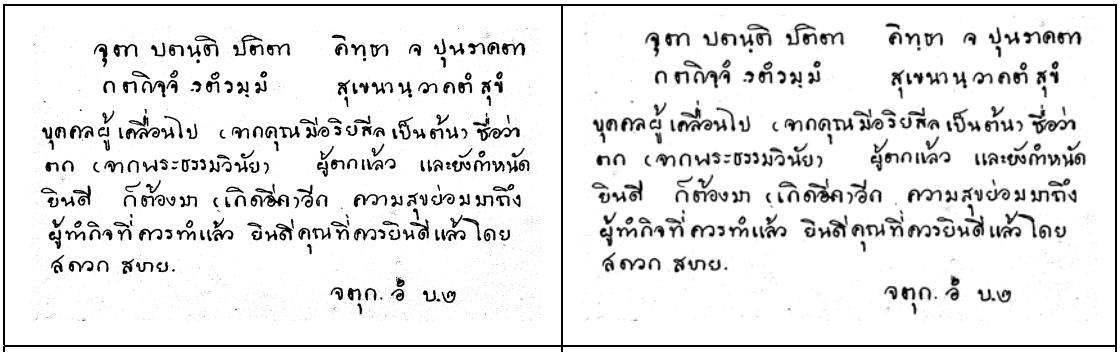
(a) Binary image (b) Sample result

Figure 5.11 Wiener Filter 3x3 in Group 1



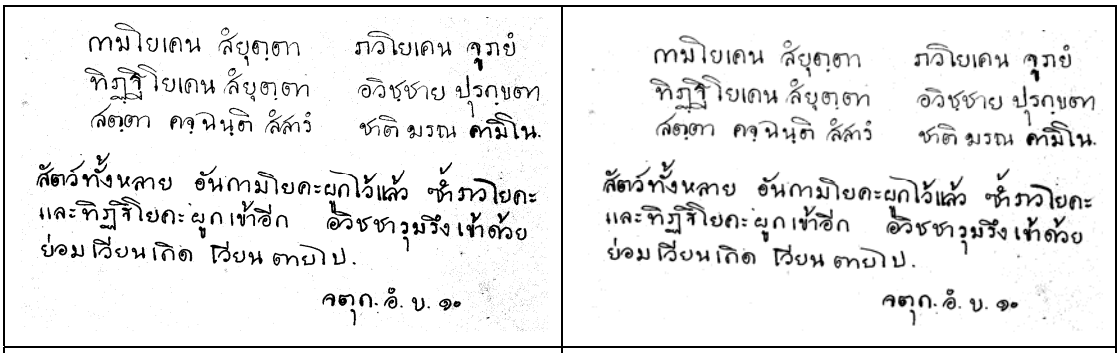
(a) Binary image (b) Sample result

Figure 5.12 Wiener Filter 3x3 in Group 2



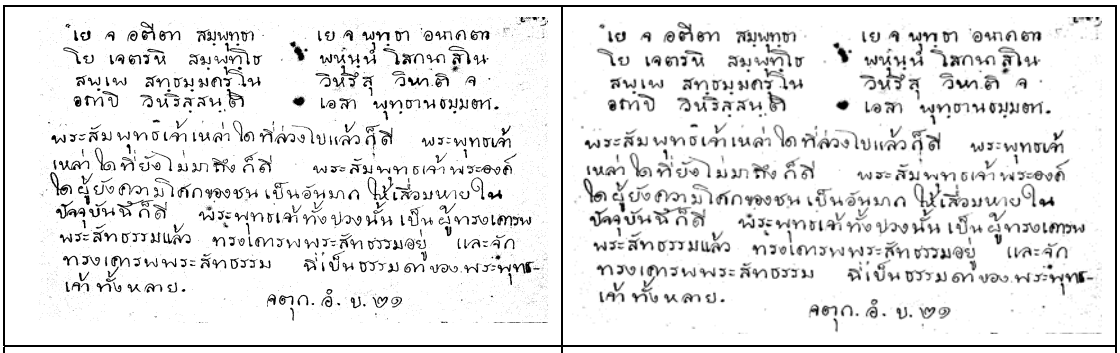
(a) Binary image (b) Sample result

Figure 5.13 Wiener Filter 3x3 in Group 3



(a) Binary image (b) Sample result

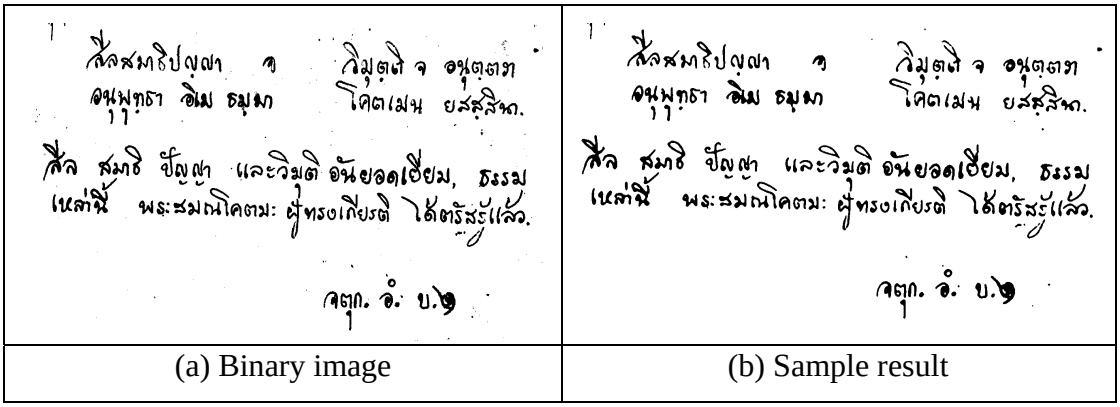
Figure 5.14 Wiener Filter 3x3 in Group 4



(a) Binary image (b) Sample result

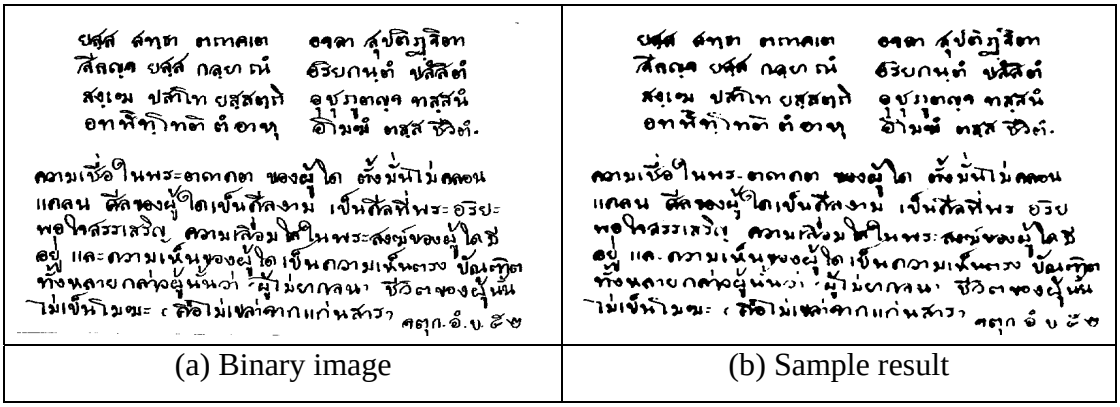
Figure 5.15 Wiener Filter 3x3 in Group 5

When the noises are removed by Wiener Filter with window size 3x3, the noises are still remained in these images. The results of Wiener Filter are not differed from Gaussian Filter.



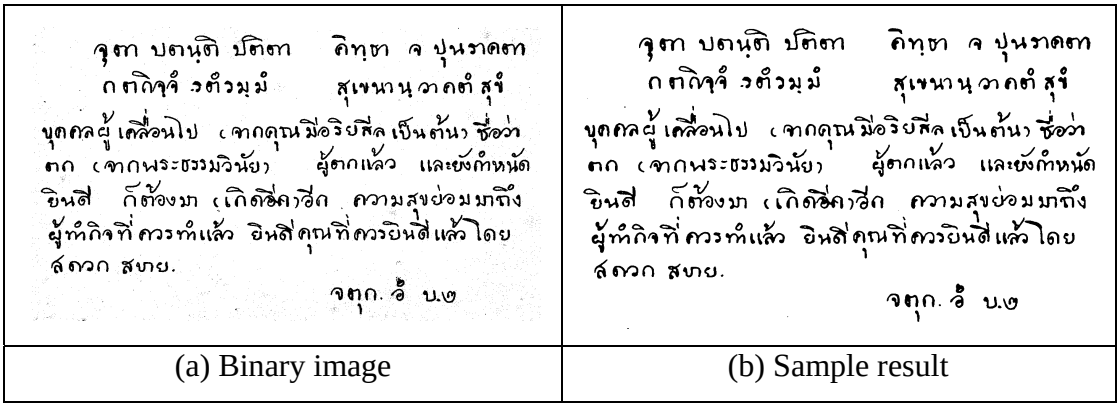
(a) Binary image (b) Sample result

Figure 5.16 Median Filter 5x5 in Group 1



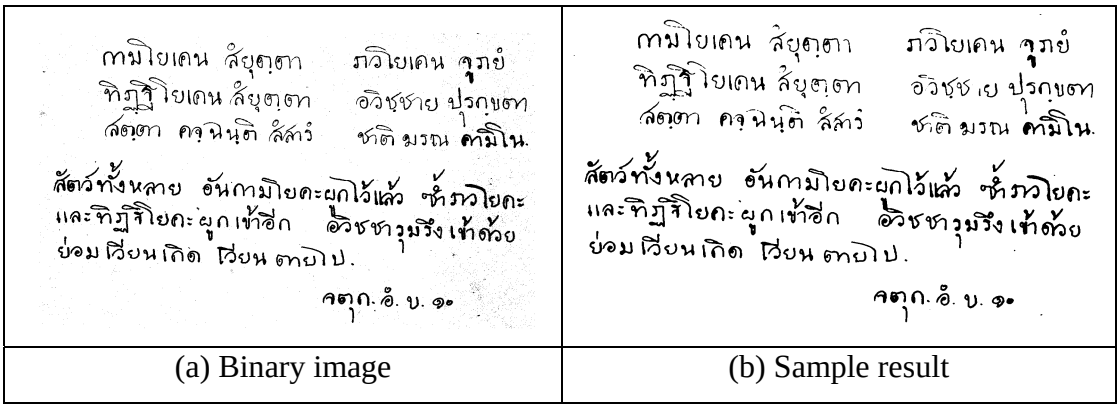
(a) Binary image (b) Sample result

Figure 5.17 Median Filter 5x5 in Group 2



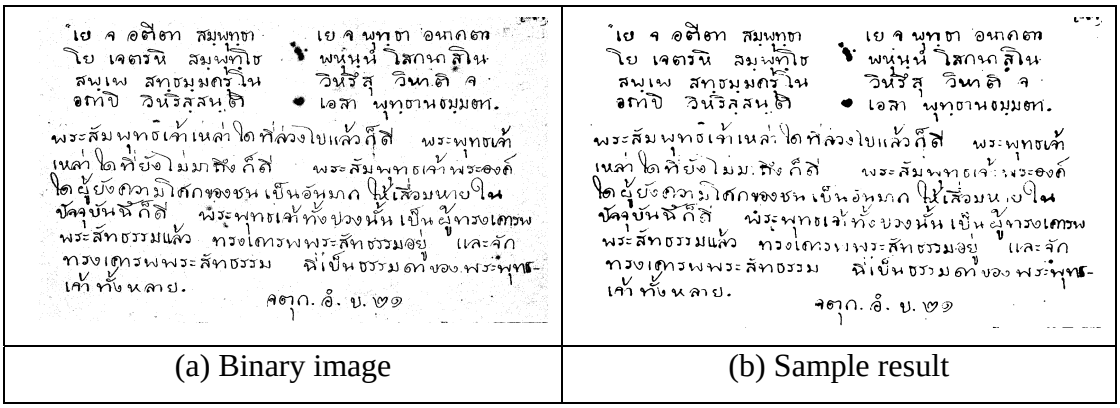
(a) Binary image (b) Sample result

Figure 5.18 Median Filter 5x5 in Group 3



(a) Binary image (b) Sample result

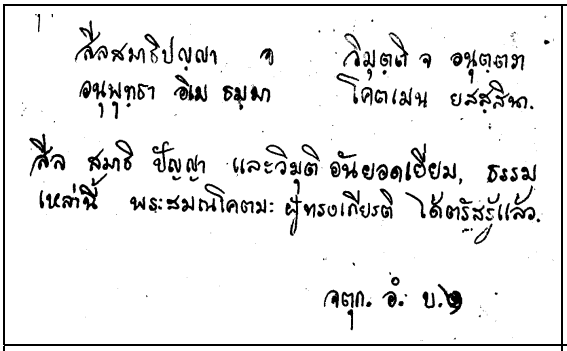
Figure 5.19 Median Filter 5x5 in Group 4



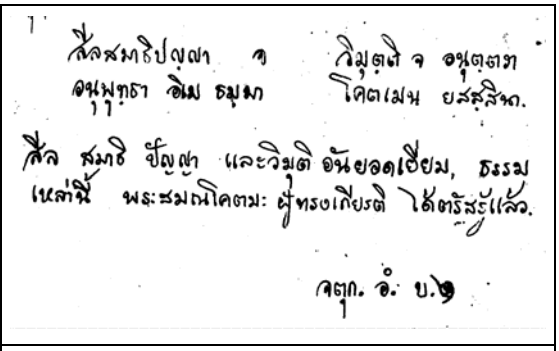
(a) Binary image (b) Sample result

Figure 5.20 Median Filter 5x5 in Group 5

When increase the window size to 5x5 in the Median Filter, this method can be removed can be removed the large noise. This method is not suitable for the smaller or thin character in the image. The detail of these characters is disappear as in the sample result image in group 2 and group 5.

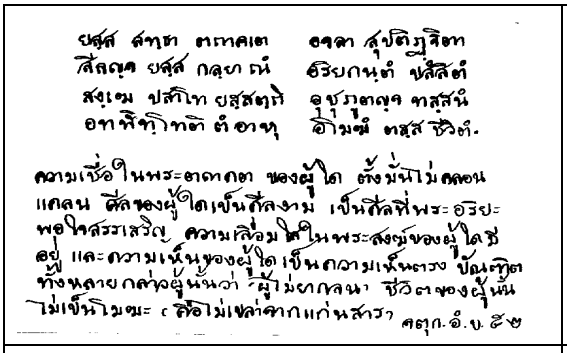


(a) Binary image

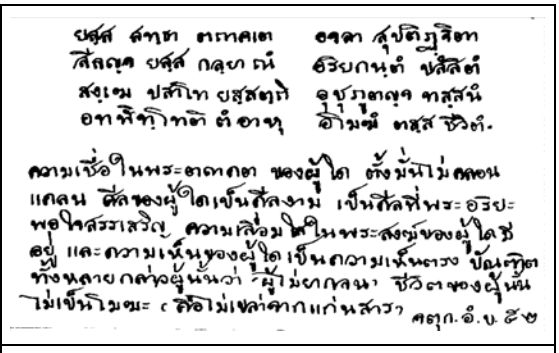


(b) Sample result

Figure 5.21 Gaussian Filter 5x5 in Group 1

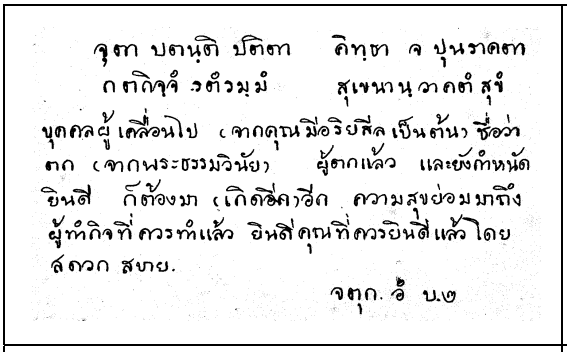


(a) Binary image

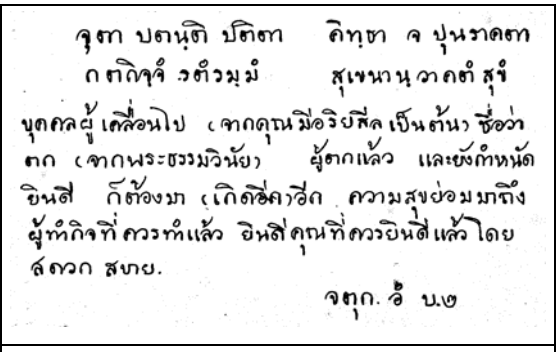


(b) Sample result

Figure 5.22 Gaussian Filter 5x5 in Group 2

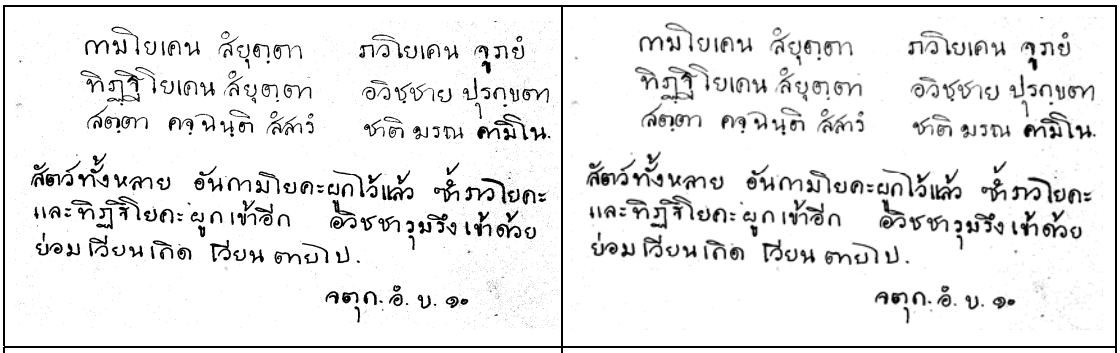


(a) Binary image



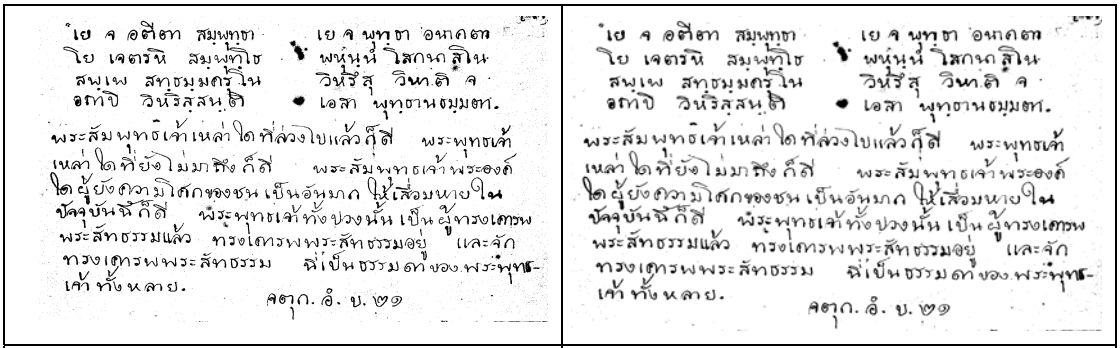
(b) Sample result

Figure 5.23 Gaussian Filter 5x5 in Group 3



(a) Binary image (b) Sample result

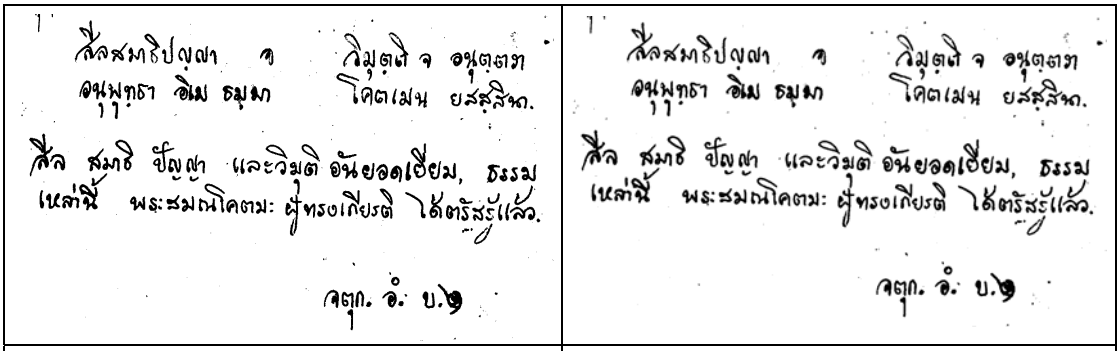
Figure 5.24 Gaussian Filter 5x5 in Group 4



(a) Binary image (b) Sample result

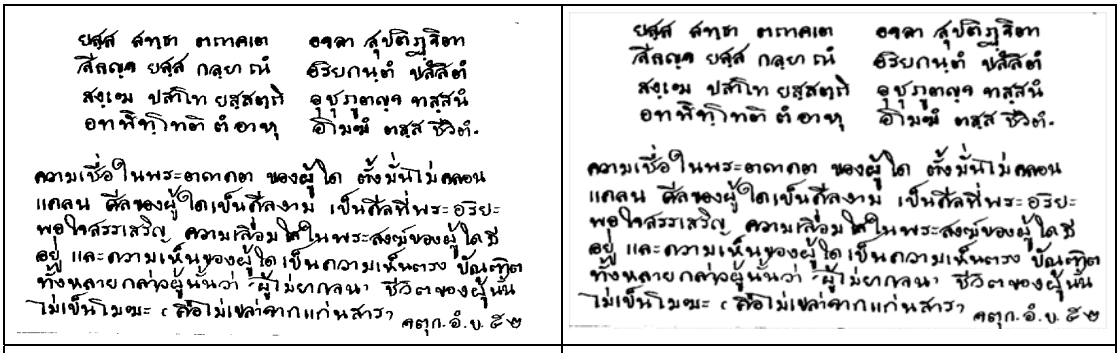
Figure 5.25 Gaussin Filter 5x5 in Group 5

When increase the window size to 5x5 in Gaussian Filter ,The binary image before remove noise and the result of binary image after remove noise does not difference.



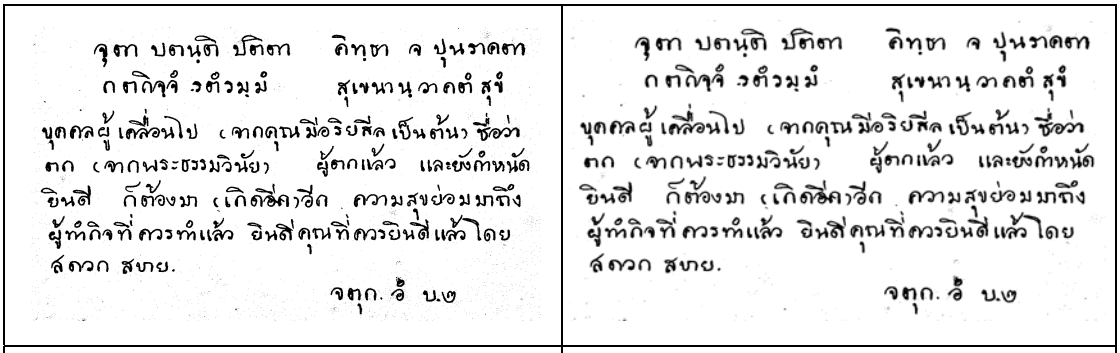
(a) Binary image (b) Sample result

Figure 5.26 Wiener Filter 5x5 in Group 1



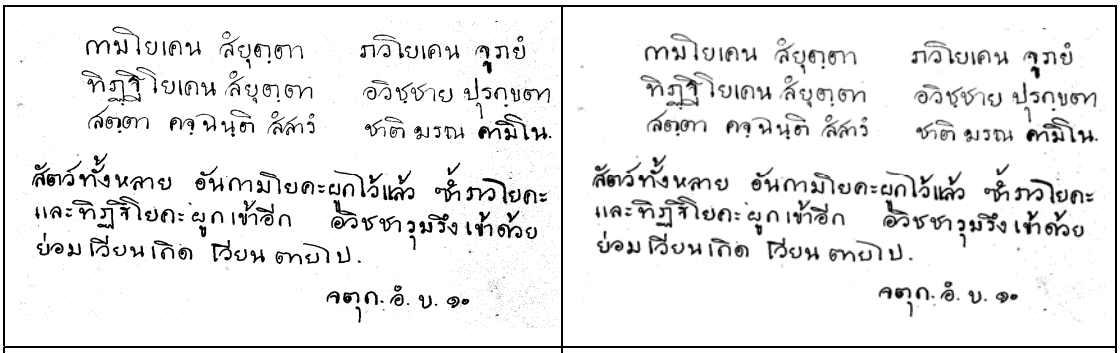
(a) Binary image (b) Sample result

Figure 5.27 Wiener Filter 5x5 in Group 2



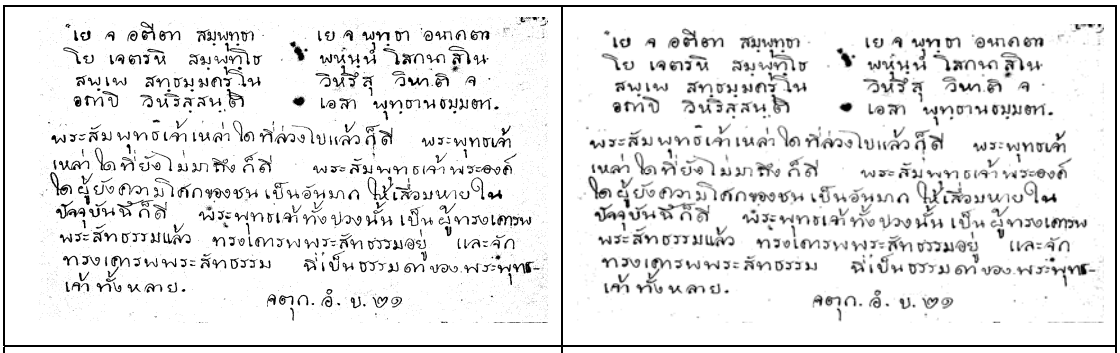
(a) Binary image (b) Sample result

Figure 5.28 Wiener Filter 5x5 in Group 3



(a) Binary image (b) Sample result

Figure 5.29 Wiener Filter 5x5 in Group 4



(a) Binary image (b) Sample result

Figure 5.30 Wiener Filter 5x5 in Group 5

When increases the window size to 5x5 in Wiener Filter, the noises are occurred in the binary image cannot reduced in this method.

From the results of noise removed in Figure 5.1 – 5.30, the Median Filter is the best method to remove noises are occurred on Buddhadasa Indapanno's handwritten image after converted grayscale image into binary image. The Median Filter with 5x5 can be removing the large noises more than Median Filter with 3x3. But the large window size is affected to the image after removed noise. The image is blurred and the detail of small and thin characters is disappeared. Therefore, the Median Filter with 3x3 is suitable to be used in this research. The percentage of noise removing in each technique is shown in Table 5.1.

Table 5.1 The percentage of noise removing

Group of image	Window size 3x3			Window size 5x5		
	Median	Gaussian	Wiener	Median	Gaussian	Wiener
1	0.354 %	48.028 %	48.028 %	1.214 %	48.028 %	85.569 %
2	-0.183 %	37.199 %	37.198 %	0.214 %	37.198 %	69.191 %
3	-0.007 %	37.663 %	37.663 %	0.376 %	37.663 %	70.305 %
4	1.731 %	46.304 %	46.304 %	1.927 %	46.304 %	81.629 %
5	6.037 %	60.589 %	60.589 %	7.185 %	60.589 %	91.804 %

From the Table 5.1, the percentages of noise removing with Median Filter are less than other technique. Due to, this technique is increasing the thick of character. The thin and fade character will be thicker. The Wiener Filter technique with window size 5x5 is the most of noise removed this technique can be remove some noise. However, some noises are still remained. The original of Binary image is shown in Figure 5.31. The Figure 5.32 and 5.33 is show the some sample result is sensitive to the noise removing with difference window size in Median Filter. The Figure 5.34 is shown the result of noise removing with Wiener Filter with window size 5x5.

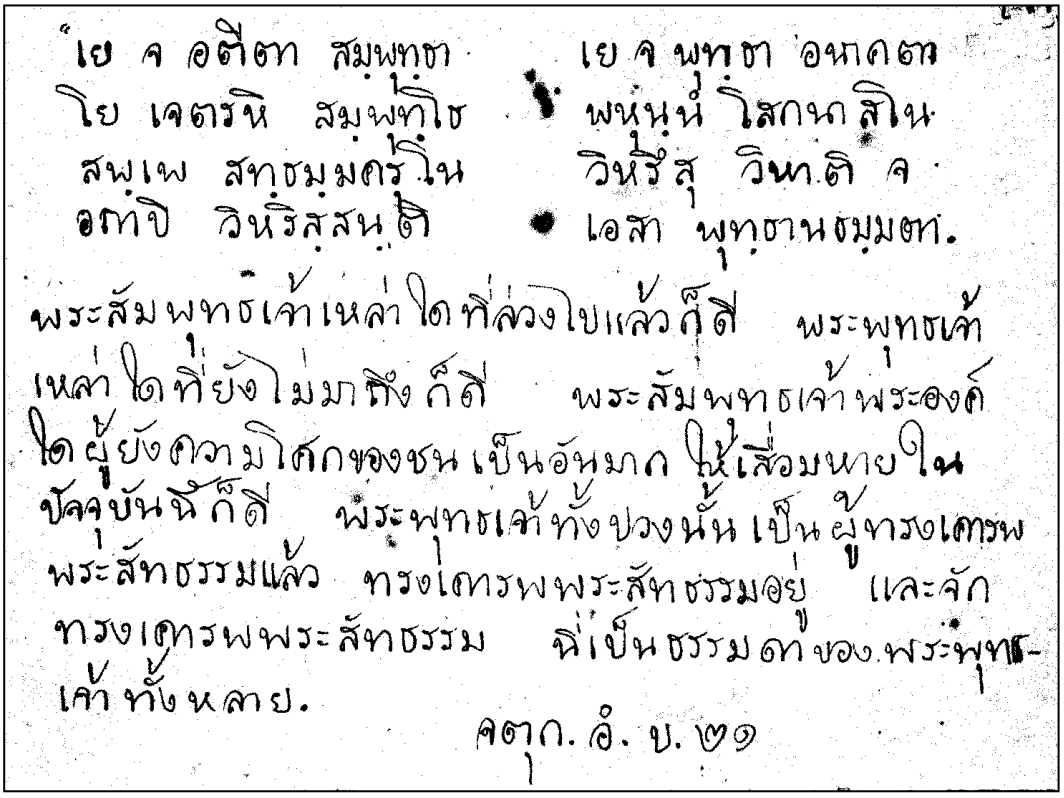


Figure 5.31 Original Binary Image

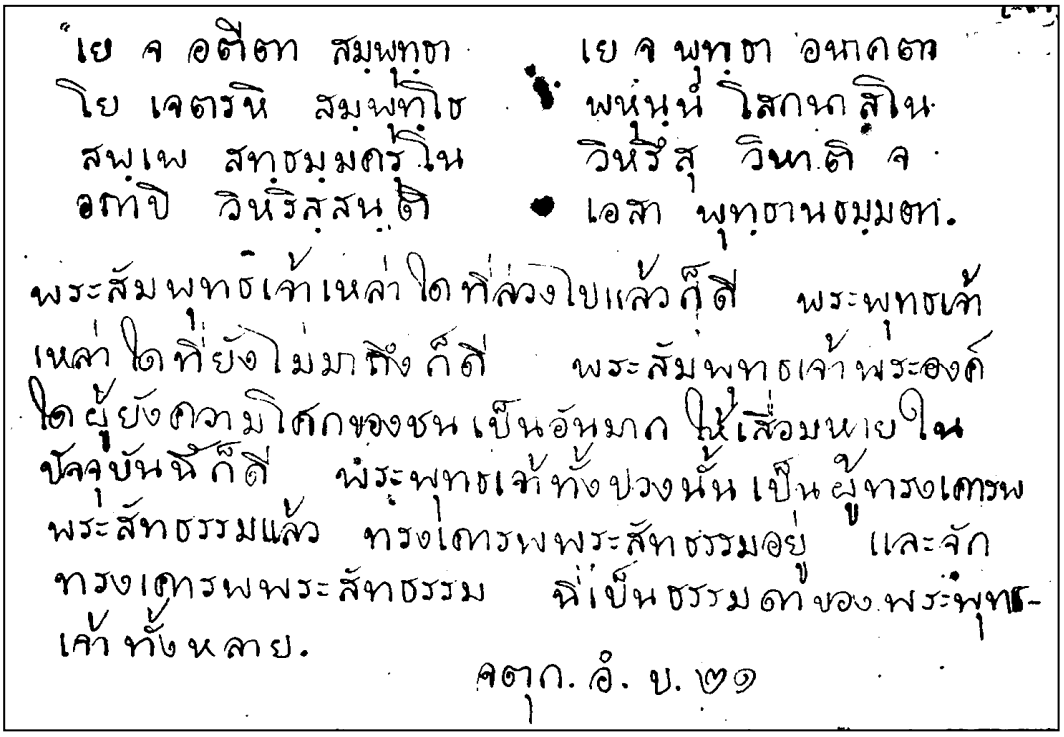


Figure 5.32 Result of Median Filter 3x3

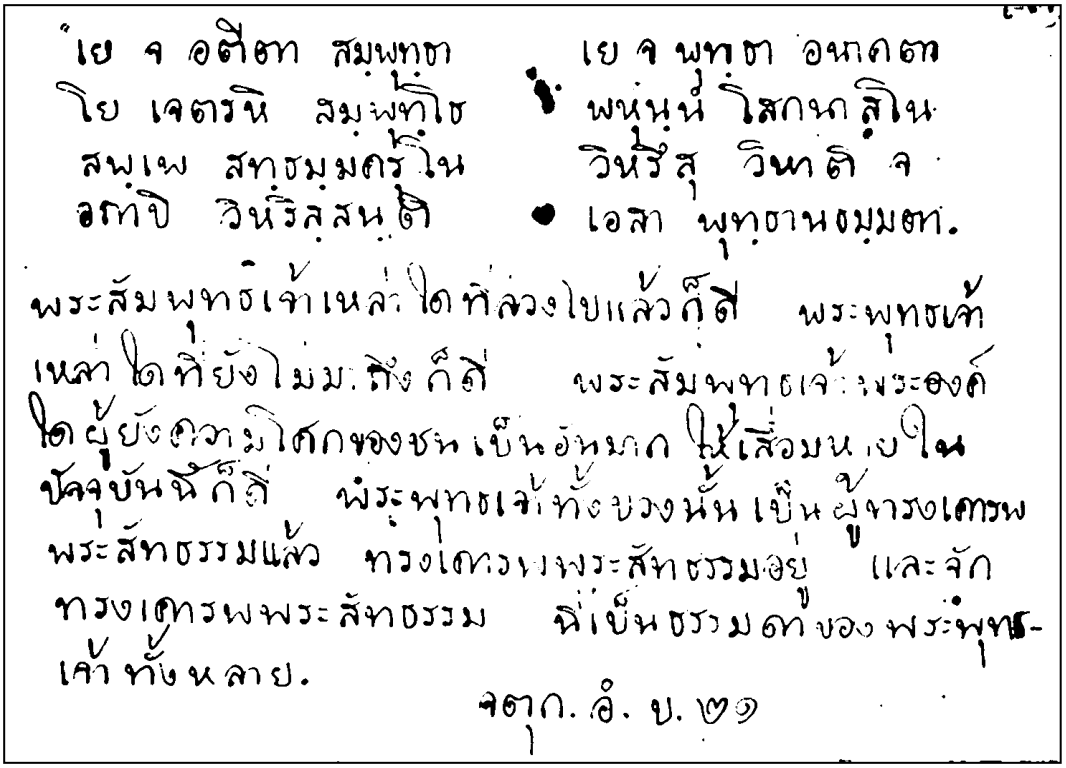


Figure 5.33 Result of Median Filter 5x5

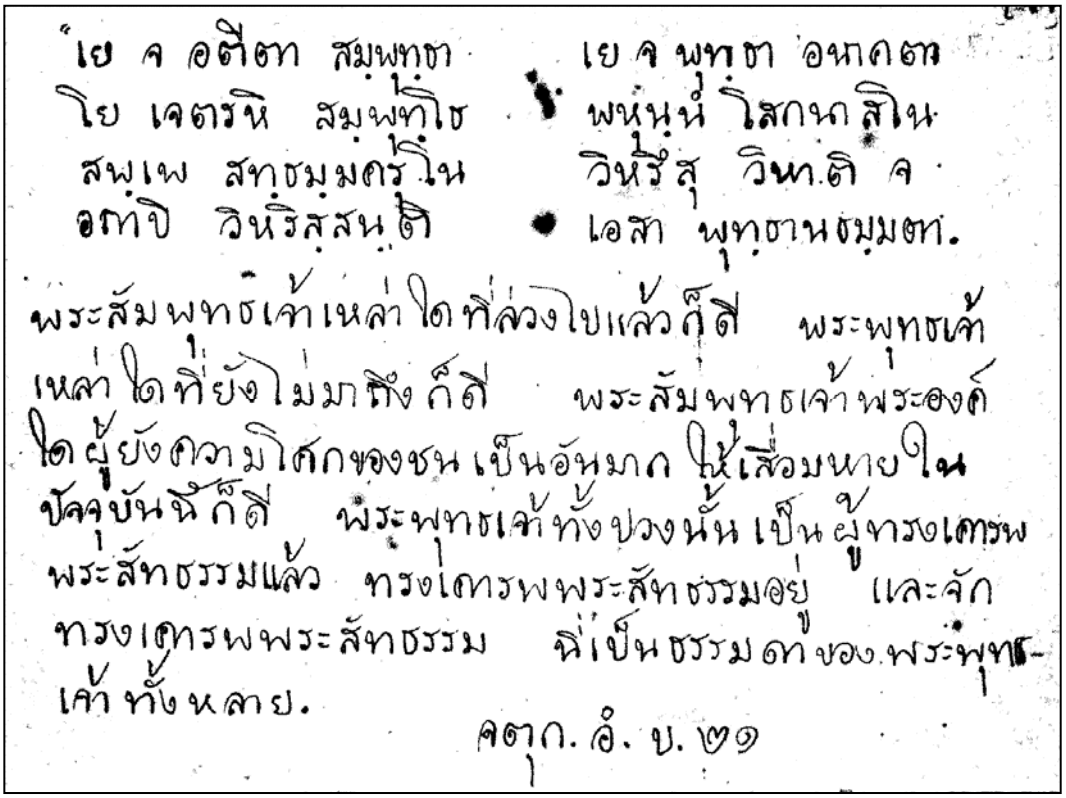


Figure 5.34 Result of Wiener Filter 5x5

The image result from Pre-processing Process is binary image. This image will be segmented the character into single character in character segmentation processes. The Figure 5.35 is shown the original images are input and the binary image results.

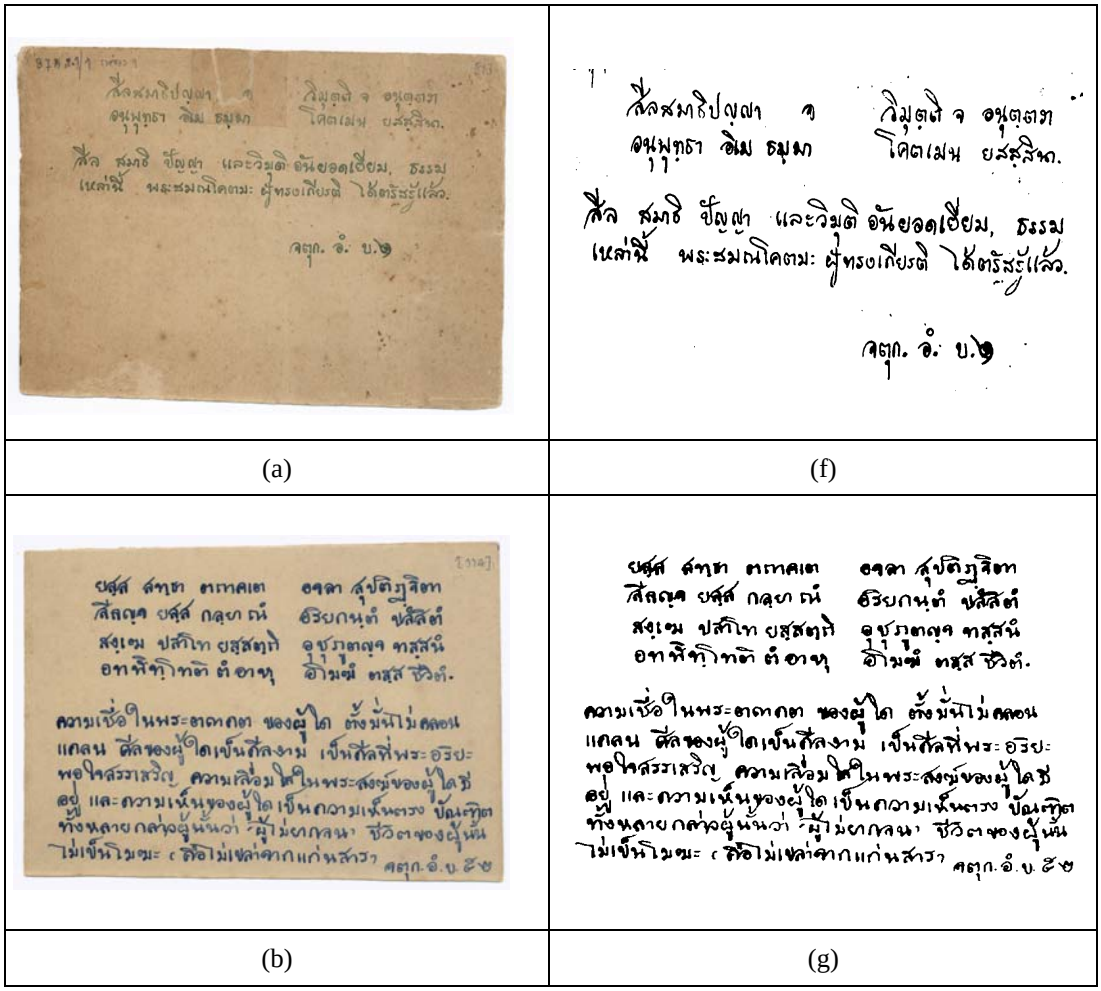


Figure 5.35 The original images and binary images of 5 groups

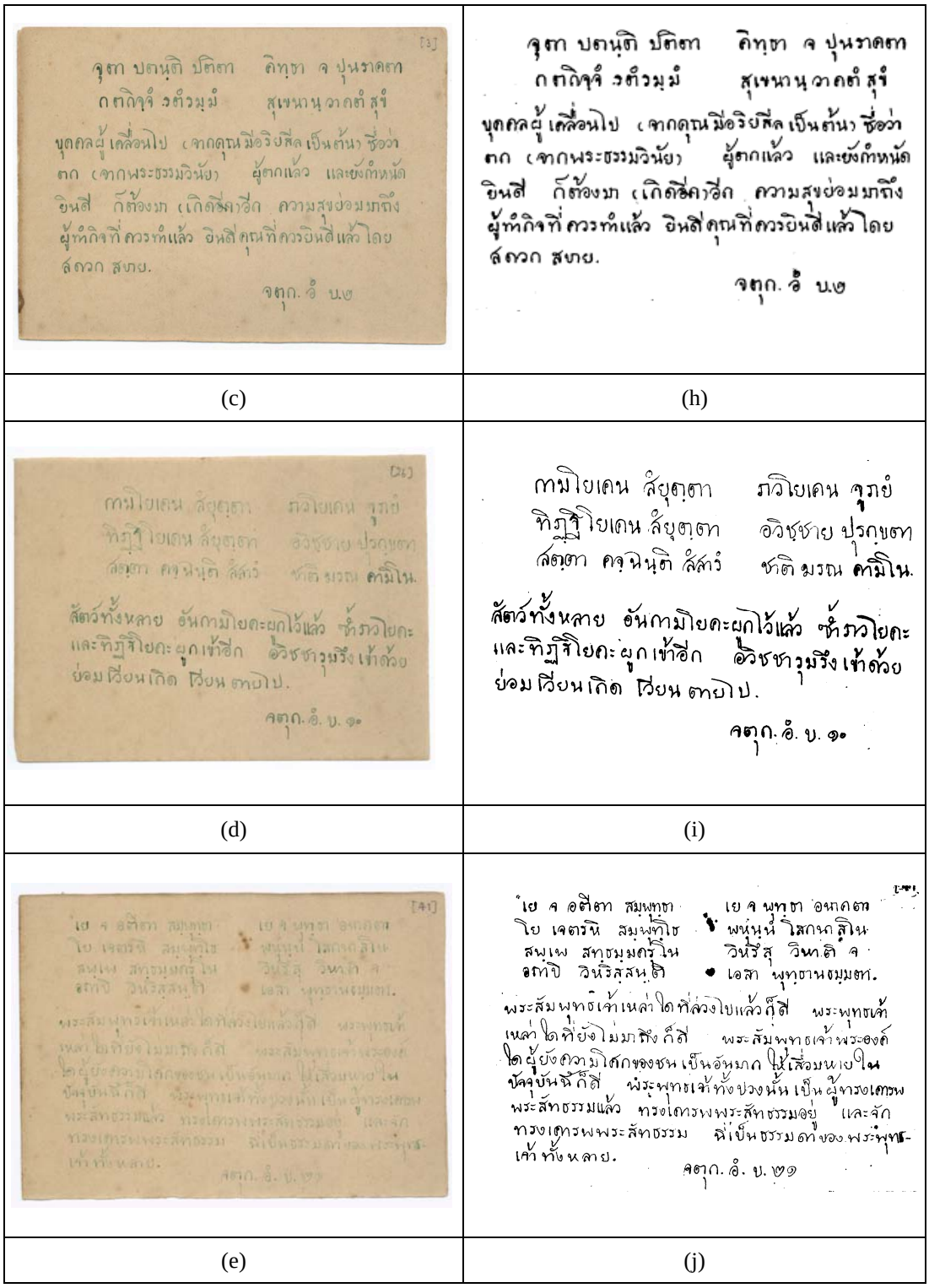


Figure 5.35 The original images and binary images of 5 groups (cont.)

From Figure 5.35, the Figure in 5.35 (a) , (b) , (c) , (d) , and (e) are original image group1 to group 5 respectively and Figure 5.35 (f), (g), (h) , (i) , and (j) are result of binary image in group1 to group5 respectively .

5.2 The Result of Character Segmentation

This section is the processes to divide the character into single characters, there are two sub-processes as connected component labeling process, and projection profile process. Then, the result of this section is separated into two parts which are: the result of image after segmented by connected component labeling process, and the result of image after segmented by projection profile process. A sample of input image in this section was show in Figure 5.36.

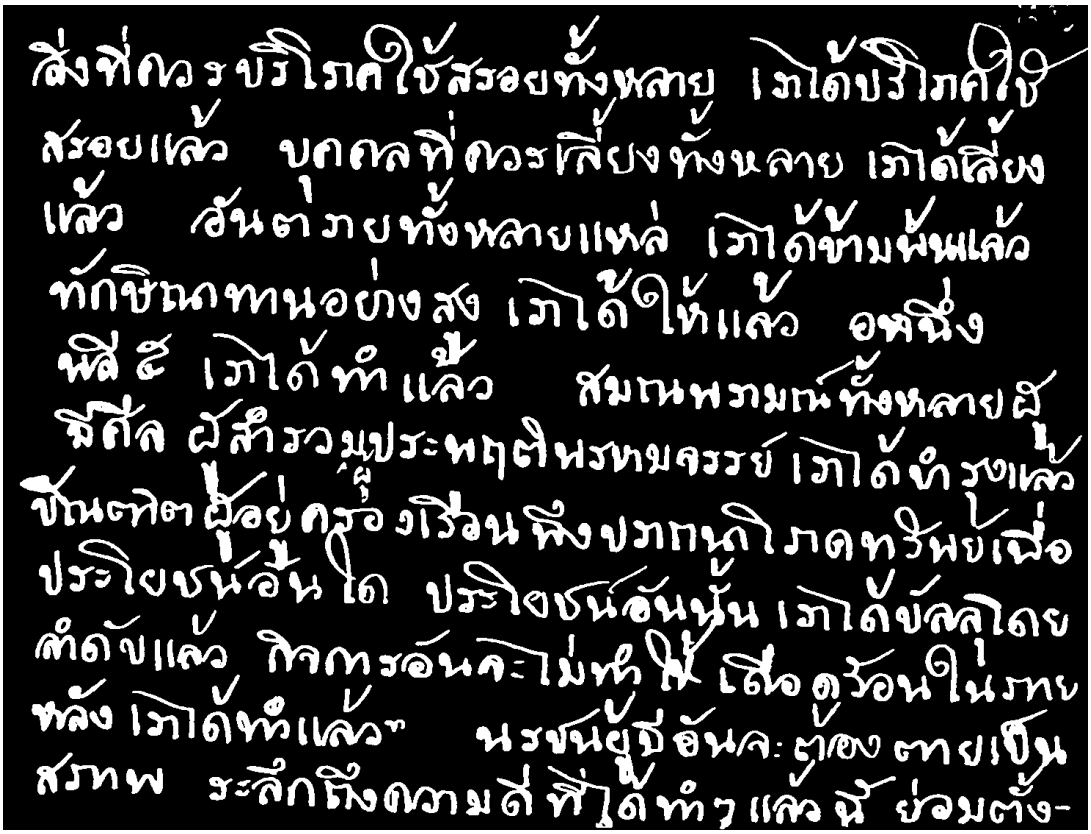


Figure 5.36 A sample of input image in character segmentation

5.2.1 Connected Component Labeling Process

The result of character images after segmented by connected component labeling process are single characters, touching character, and not character such as point, and line with edge of image. The result this method is shown in Figure 5.37-5.39, and the result is not character is show in Figure 5.40.

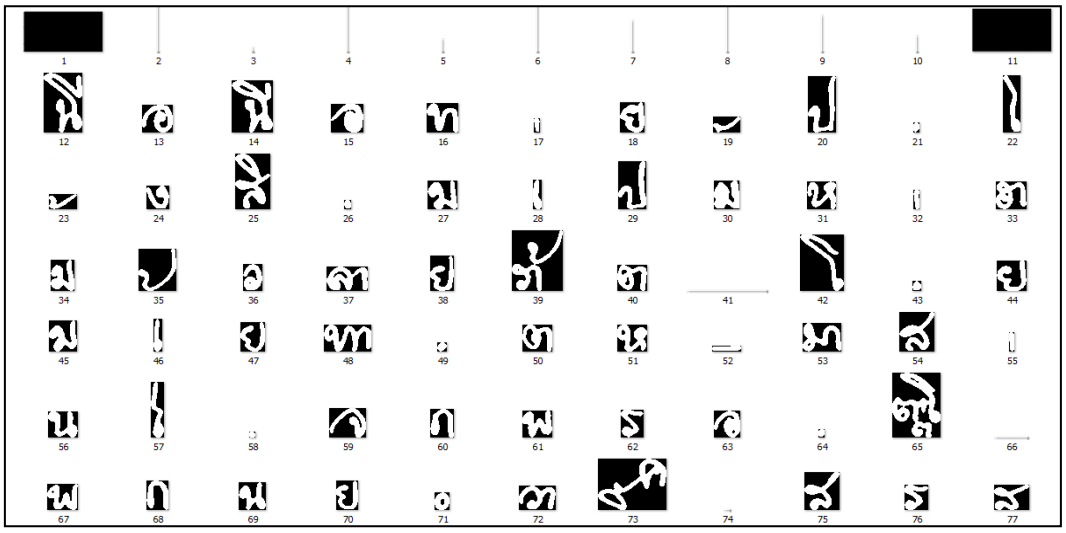


Figure 5.37 A sample of images after segmented by connected component labeling process

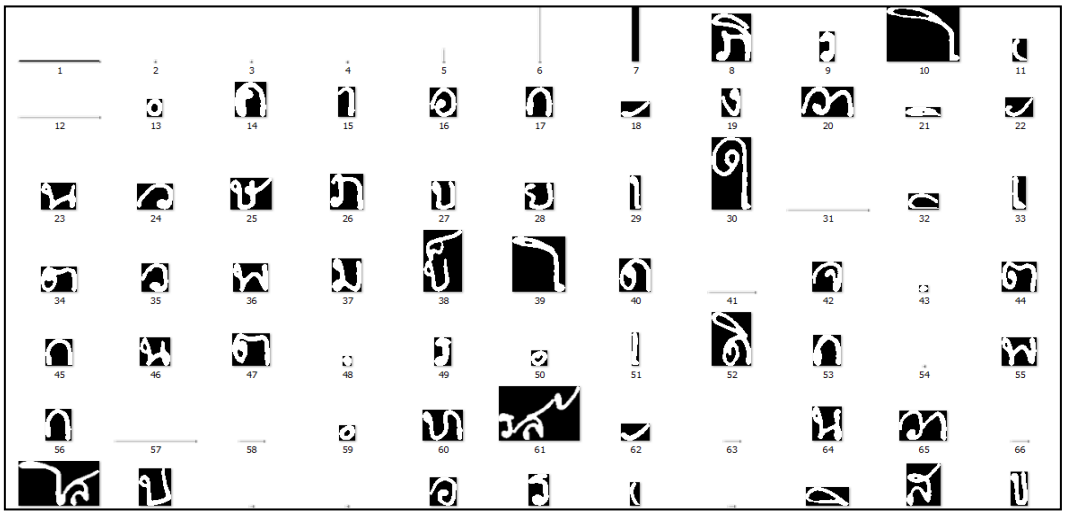


Figure 5.38 A sample of images after segmented by connected component labeling process

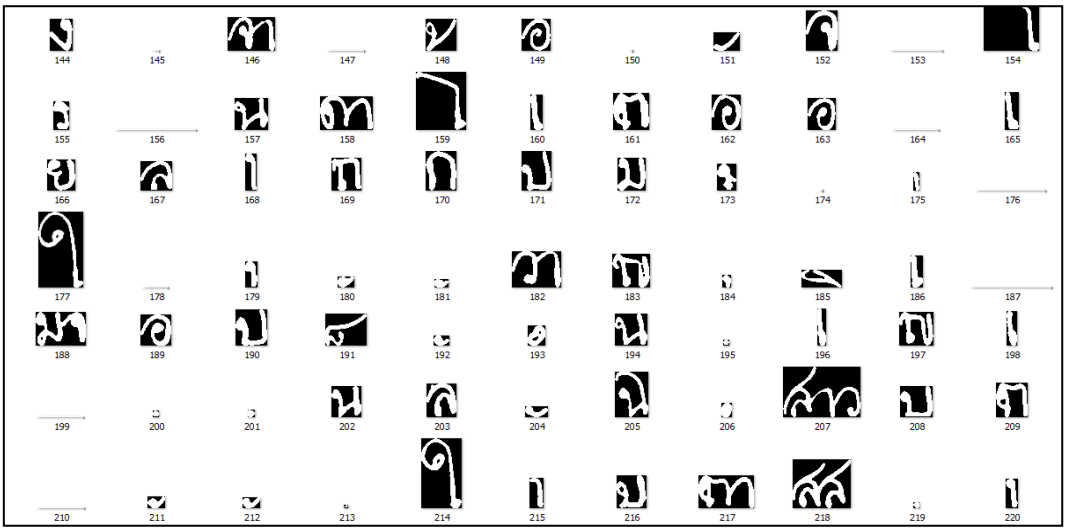


Figure 5.39 A sample of images after segmented by connected component labeling process

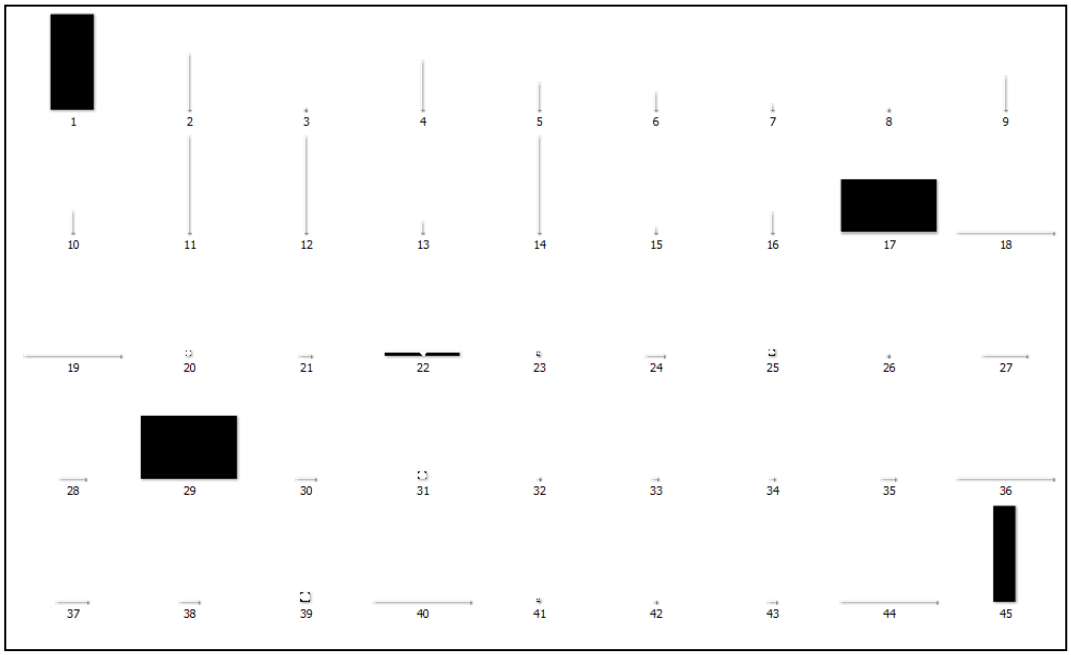


Figure 5.40 A sample result of image is not character image.

The image is not character was show the sample result in Figure 5.40 will be removed. The final result of this processes are single character and touching character, the sample result of image after remove the image is not character is show in Figure 5.41 - 5.43.

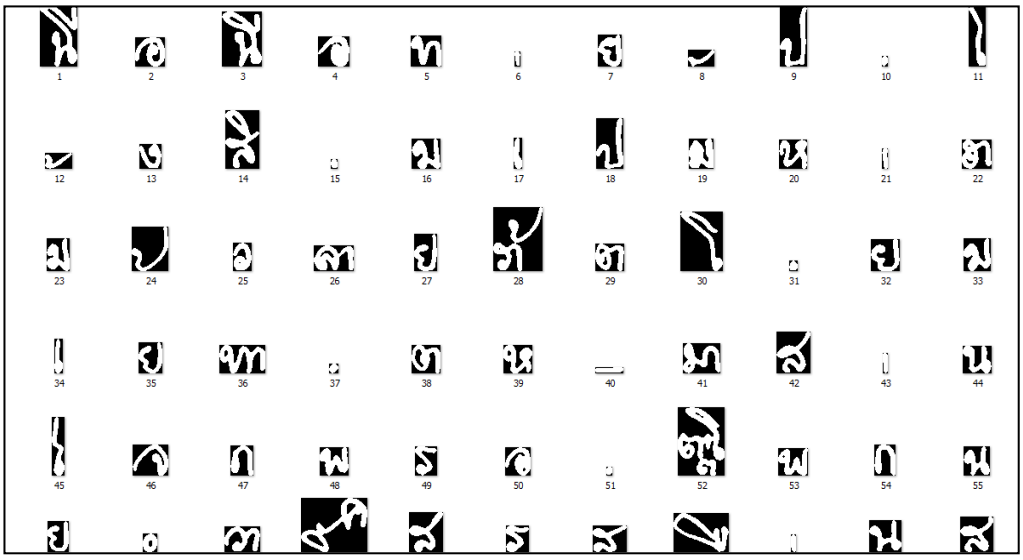


Figure 5.41 A sample of character image after removed not character

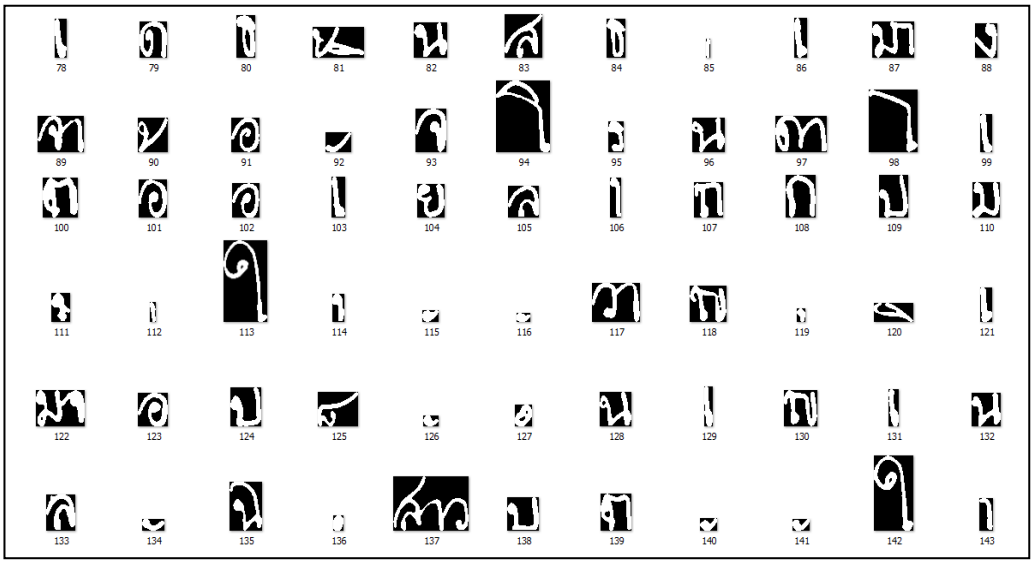


Figure 5.42 A sample of character image after removed not character

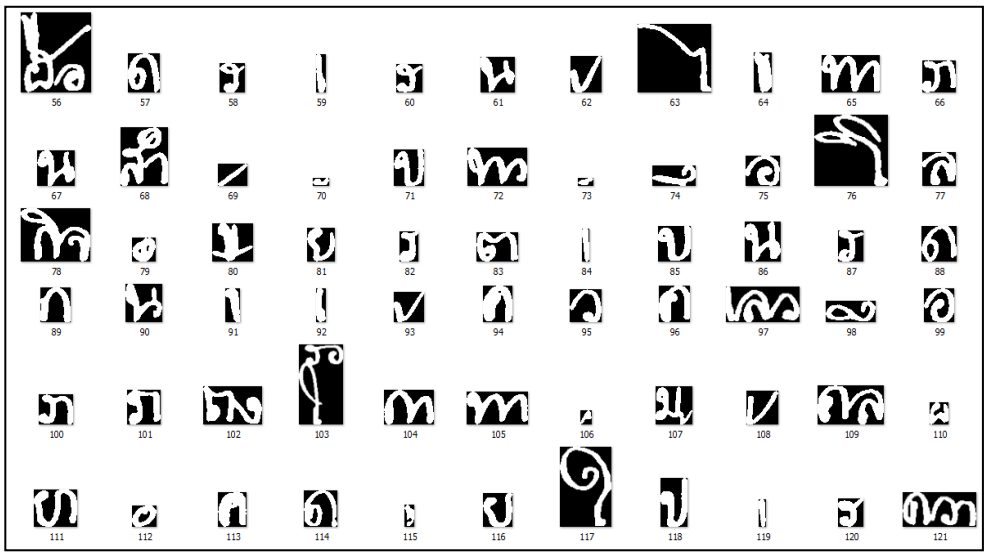


Figure 5.43 A sample of character image after removed not character

Figure 5.41 - 5.43, there are show the result after removed the not character image. After that, this result will be classified into two groups as single character group and touching character group. The criterion to classify the character image group will be using height and width of character image and proportion of black and white pixel of character to separate the single character from touching character. The result of single character image was separated from touching character is show in Figure 5.44, and 5.45.

Figure 5.44 A sample result of single character image

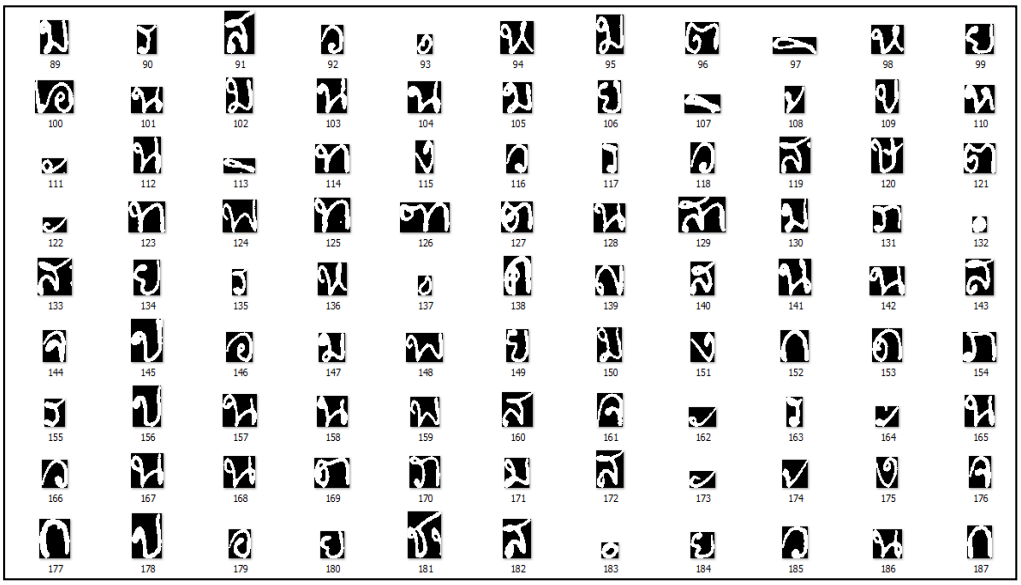


Figure 5.45 A sample result of single character image

Figure 5.44 and 5.45, the result after separate single character images from touching character images, there are some touching character images remain in single character images group as the touching character image group, there are some of single character images in this group. The error of this result is occurred from the height and width of character image, and proportion of black and white pixel of these character images are in the criterion in each group. The percentage of single character from connected component labeling process is shown in Table 5.2.

Table 5.2 The percentage of single character in each group

Group of image	Percentage of single character
1	87.14
2	79.29
3	80.13
4	84.06
5	**87.18**

From Table 5.2, there are shown the percentage of single characters in 5 groups are separated from Connected Component Labeling process. The most of single characters in group 5 can be separated, although the characters of image in this group are thin and fade. However, the characters in group 2 are small and thick. The mostly characters in this group are touching characters, that are effected to the percentage of single character segmentation by Connected Component Labeling process is lowest.

The images in touching character image group will be divided into single characters image by Projection Profile method. The sample of touching character images are show in Figure 5.46, and 5.47.

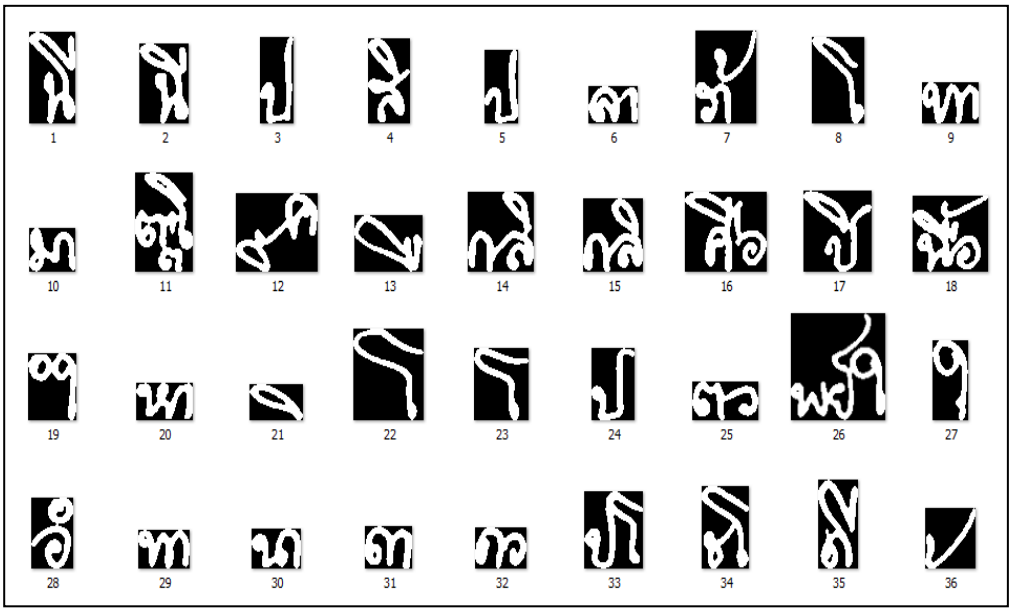


Figure 5.46 A sample of touching character images

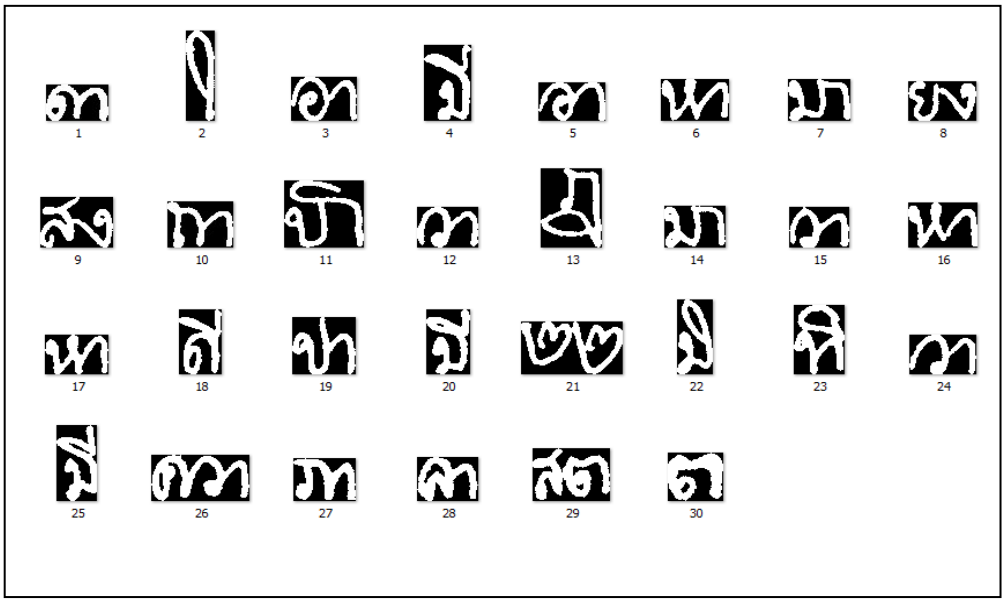


Figure 5.47 A sample of touching character images.

5.2.2 Projection Profile Process

The touching character images will be segmented into single character images, there are classify the segmentation into 3 patterns which are horizontal segmentation, vertical segmentation, and horizontal and vertical segmentation.

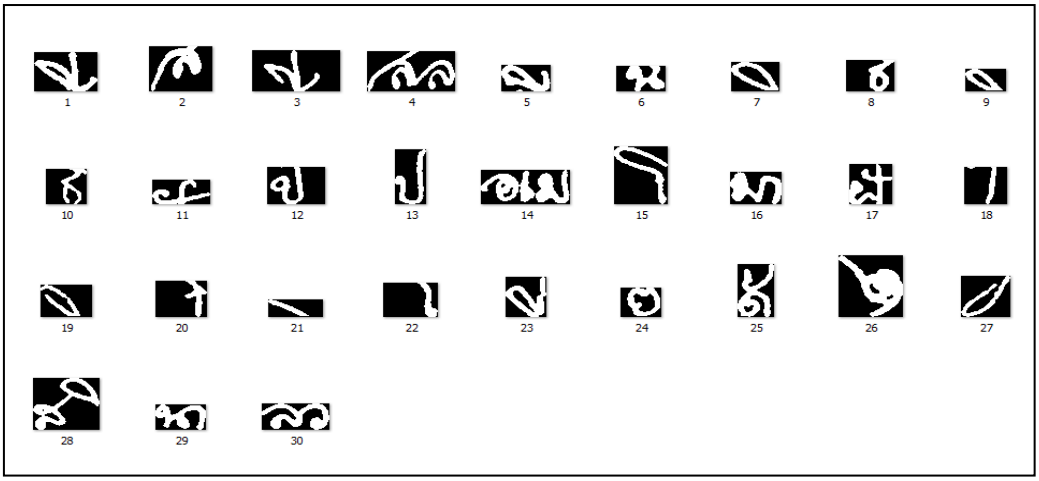


Figure 5.48 A sample result of character images after segmented by horizontal segmentation.

The Figure 5.48 was show some images after segmented in the horizontal, the result of this method is including of single character images and touching character images are touching in horizontal. The result of character images after segmented in the horizontal will be compare the width of character images with mean of width (MW) of character images in single character image group. If the width of character images is more than MW, These character images are segmented in the horizontal.

The final result of character segmentation is single character images are show in Figure 5.49 - 5.51.

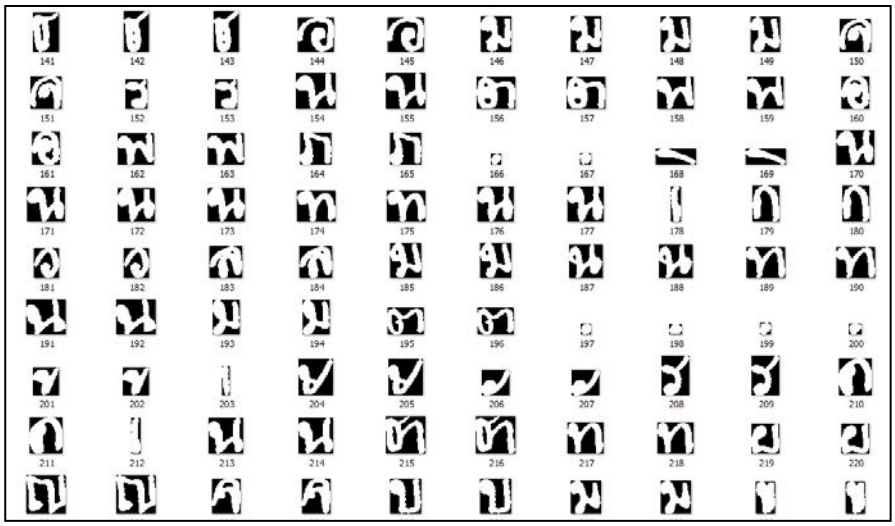


Figure 5.49 A sample of single character images result

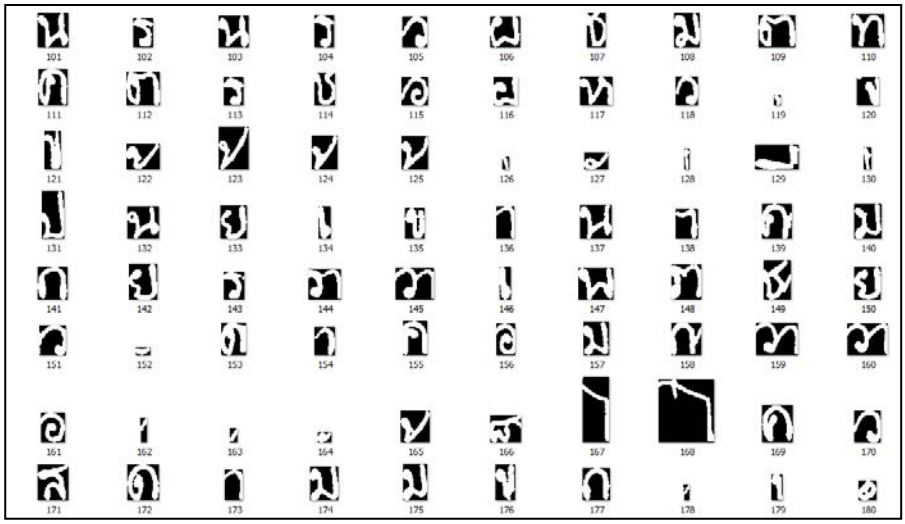


Figure 5.50 A sample of single character images result

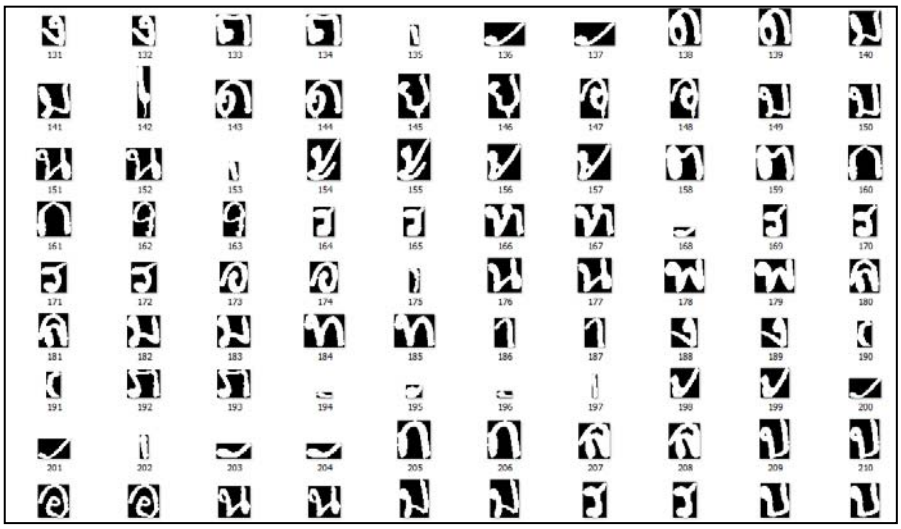


Figure 5.51 A sample of single character images result

The percentage of character segmentation correctness is shown in Table 5.3.

Table 5.3 The percentage of character segmentation

Image Group	Percentage of correctly Character Segmentation method	
	Connected Component Labeling	Connected Component Labeling and Projection Profile
1	87.14 %	**93.04 %**
2	79.29 %	88.44 %
3	80.13 %	87.14 %
4	84.06 %	90.80 %
5	87.18 %	90.57 %

Table 5.3 shows the character segmentation results. The correctness of images in group 1 is 93.04 %. That is the highest result. Projection can improve the correctness of segmentation, especially in group 2. The correctness can be improved

9.15%. These results show the increasing of correctness of character segmentation after Connected Component Labeling process.

5.3 The Result of Line Position Detection

The binary image from Pre-processing process will be used to find the text line with horizontal projection. The positions of each line are used to sort the character after character recognition process. The sample of text line are separated is shown in Figure 5.52, and 5.53

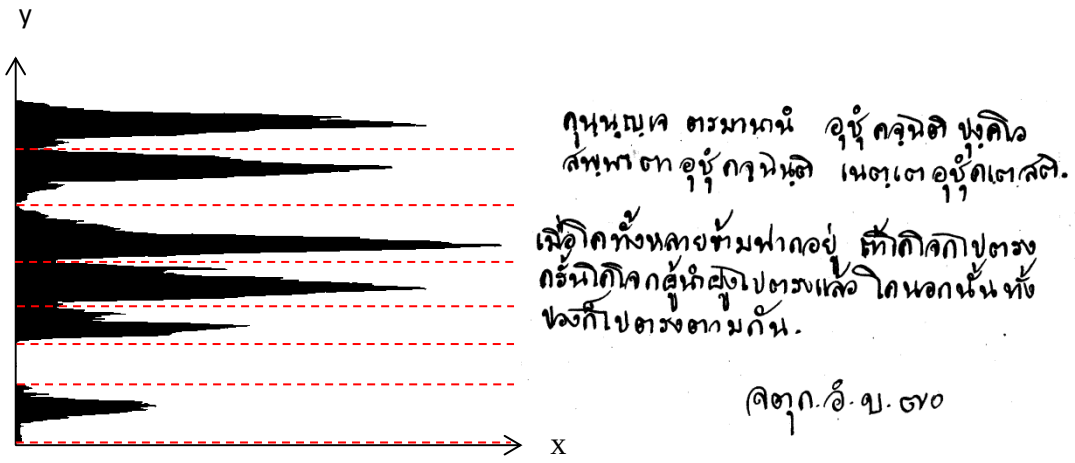


Figure 5.52 The sample of vertical projection

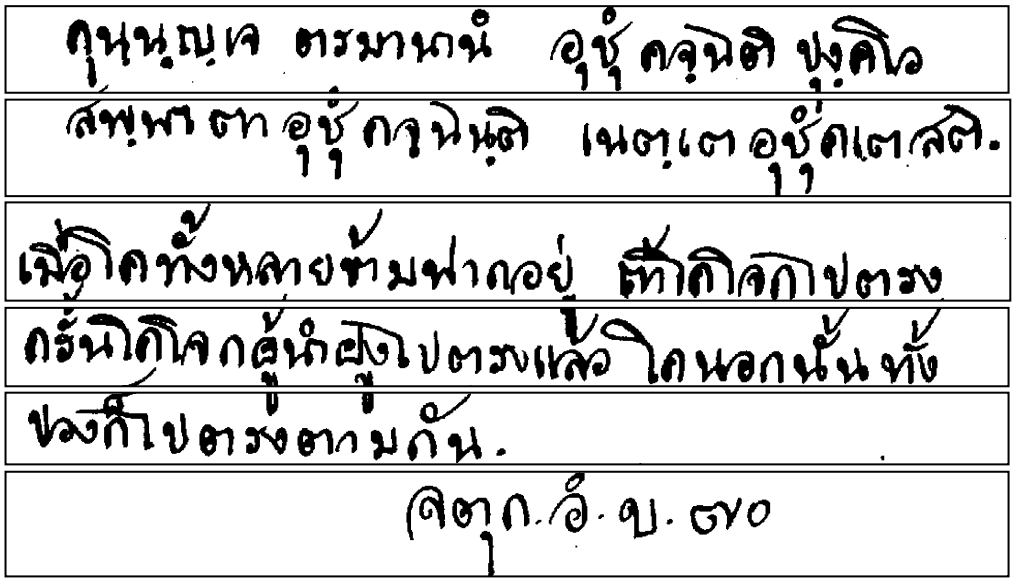


Figure 5.53 The text lines are separated by horizontal projection.

The technique to separate the text line in this research is compared to the Free OCR to word version 5.6.5 program. The comparison result is shown in Figure 5.54 and 5.55.

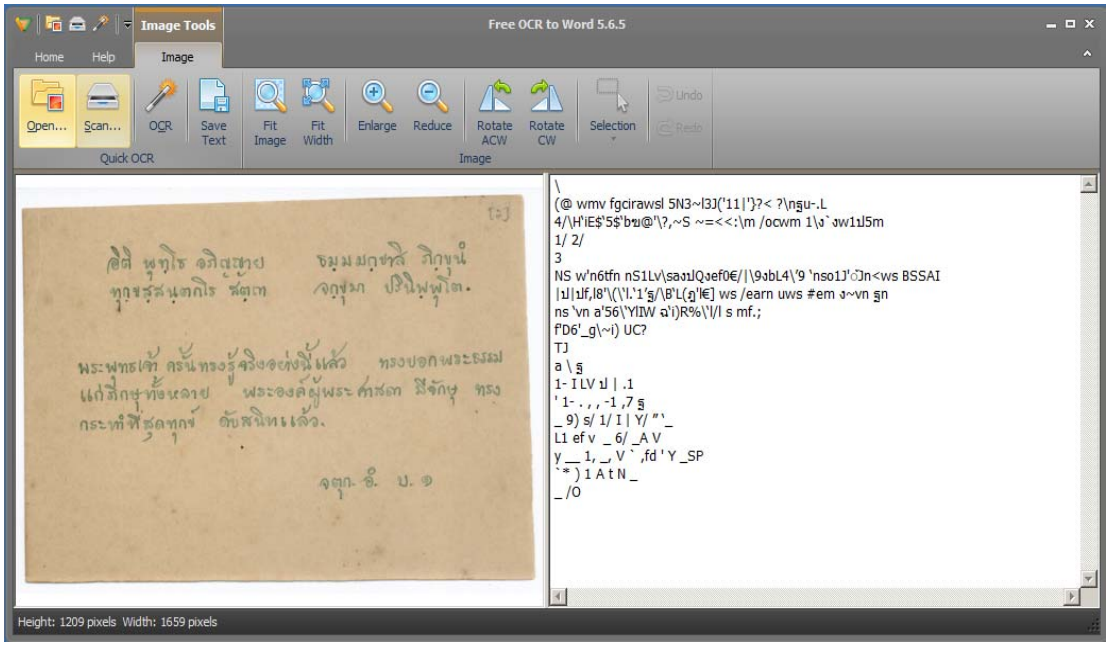


Figure 5.54 The result of line separation from Free OCR to word version 5.6.5 program

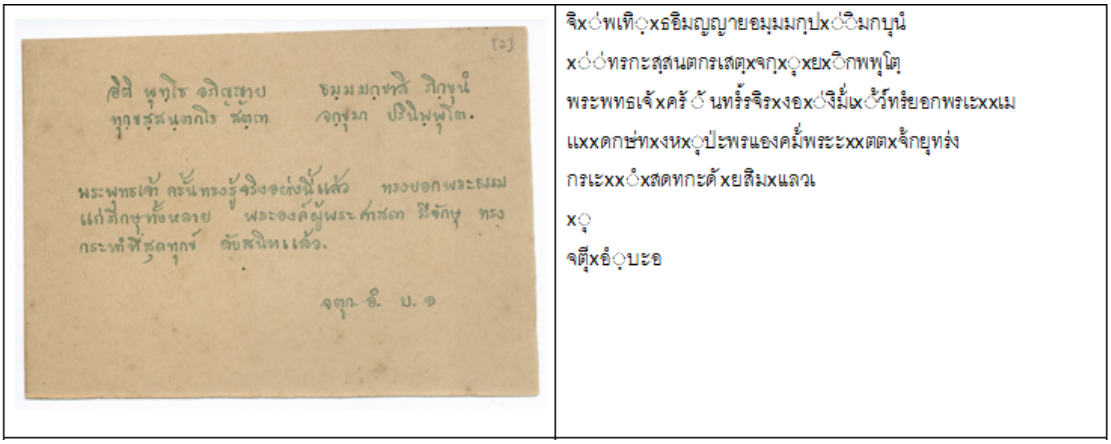


Figure 5.55 The result of line separation from this research

From the result in Figure 5.54 and 5.55, the proposed technique can separate the text line nearby to original image. The straight line of Buddhadasa Indapanno's handwritten image can be designated correctly, but the upper vowel and lower vowel between lines may be separated. The mostly of Buddhadasa Indapanno's handwritten image are slant. Therefore, this method may be un-suitable because the error of designation line is occurred. The error of line separation is affected to the error of the characters arrangement after recognition.

5.4 The Result of Zone Analysis

To find the suitable feature extraction with zoning method to increase the correctness of character recognition, this research is compared the zone with 4x4 and 8x8. The zone is divided on the character with 4x4 is bester result that is 86.78% and the zone is divided on the character with 8x8 is 78.09%. From the result, this research is selected the 4x4.

5.5 The Result of Character Recognition

5.5.1 10-Fold Cross Validation Test

This research is developed the Buddhadasa Indapanno's handwritten character recognition with feed-forward back-propagation neural network. The primary test to find the suitable model of neural network in this research is 10-Fold Cross Validation. The 69 class are used to train, there are included 42 consonants, 11 numbers, and 16 vowels class. The comparative of the maximum number of epochs to train is shown in Table 5.4. The increment of epochs is affected to the percentage of accuracy rate. However, the increment of epochs is direct variation with time. The tendency of increment with epochs and percentage of accuracy rate is shown in Figure 5.56.

Table 5.4 The comparative of parameter to train the neural network model

Training Parameters	**1**	**2**	**3**	**4**
net.trainParam.epochs	1000	3000	5000	10000
net.trainParam.goal	0	0	0	0
net.trainParam.lr	0.01	0.01	0.01	0.01
net.trainParam.lr_inc	1.05	1.05	1.05	1.05
net.trainParam.lr_dec	0.7	0.7	0.7	0.7
net.trainParam.max_fail	5	5	5	5
net.trainParam.max_perf_inc	1.04	1.04	1.04	1.04
net.trainParam.mc	0.9	0.9	0.9	0.9
net.trainParam.min_grad	1E-10	1E-10	1E-10	1E-10
net.trainParam.show	25	25	25	25
net.trainParam.time	inf	inf	inf	inf
Percent of accuracy (%)	80.14	91.11	93.58	**95.57**
Approximate Time	1.26 hr	4.08hr	6 hr	12hr

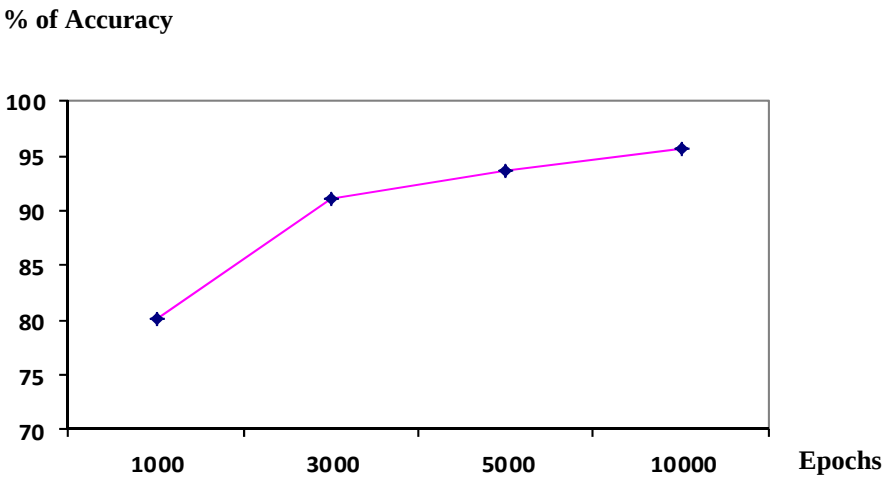


Figure 5.56 The tendency of accuracy rate

From Figure 5.56 is shown the graph of percentage of accuracy rate of Maximum number of epochs to train. At 10,000 of epochs in this experiment, the graph will be stable.

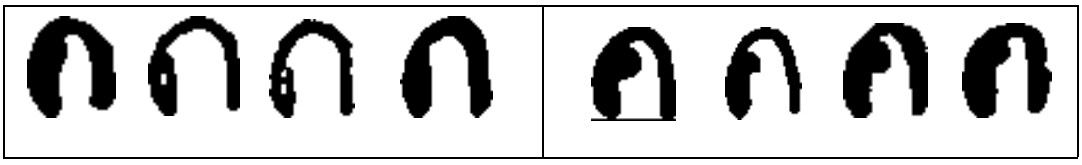
5.5.2 Creation of Neural Network Model

The total of character images was using to training the neural model are 28,584 characters, there are including of consonant 13,038 characters, number 252 characters, vowel 7,555 characters, tone 1,696 characters, symbol 901 characters, and unknown character 5,142 characters. The correctness of character recognition in each group is shown in Table 5.5.

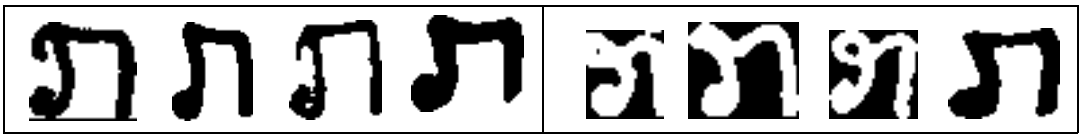
Table 5.5 The percentage of character recognition

Group of Image	Percentage of character recognition
1	75.80
2	**84.66**
3	79.79
4	83.86
5	84.53

The most of correctness in character recognition is image in group 2. Even though, the characters in group 2 are small and thick but the system can be recognized the character in this group more than other image group. The error of character recognition may be occurred from the character in some class are closely, that is difficult to define the class of character. The samples of character are similar show in Figure 5.57 and samples the false classified of characters are shown in Table 5.6.



ก ค



ภ รา

Figure 5.57 The similarity of consonant

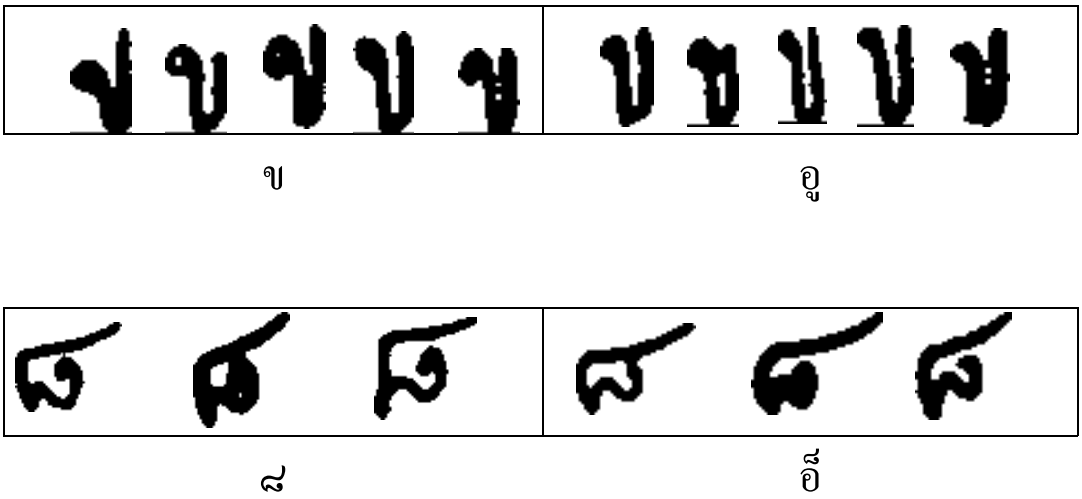

ข
ฺ
ุ
ิ
ี
ึ
ื
ั
ู
ิ
ี
ึ
ื
ั
ู
ิ
ี
ึ
ื
ั
ู

Table 5.6 The false of the character classification (cont.)

Character	Recognition Result
เอ	อ่ อะไอ ง
ไอ	โอ เอ อุ
อุ	x อ่ โอ ร อิ อั
อู	อู x ข บ ย
อุ	อะ อ่ เอ
อั	อะ อั เอ x อ้
อิ	x เอ ก อะล ร อำ
อ่	อุ เอ ร x
อ้	อั ย ฝ
x	อะ อั ภ อิ ง เอ ถ อ่ ธ ร ไอ น ห อุ ท

CHAPTER VI
CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

This research presented the Buddhadasa Indapanno's handwritten character recognition system. The handwritten images were ancient documents .The color of background was yellow–brown, and there are difference of brightness and contrast. This system is included 4 main processes which are pre-processing, character segmentation, feature extraction, and character recognition process. The first step, each handwritten image is enhanced the contrast in suitable value before convert the image into gray scale image and binary image respectively. The advantage of adjustment contrast in this research is improved the thin and fade characters are clearly. After convert the image into binary image, there are noise occurred. This research used median filter method to removed noise, because this method can remove noise and not removed the character detail in image.

In the next step, the binary image after removed noise is segmented into single characters. The segmentation process is included 2 sub-processes which are connected component labeling: the image is segmented into single characters, touching characters, and non-character image. The criterion to classify the single characters, touching characters and non-characters are height and width of character image and proportion of black and white pixel. The projection profile: the touching characters are segmented into single characters. The touching characters segmentation in this process is classified into 3 patterns as horizontal projection, vertical projection, and horizontal and vertical projection together. The correctness of segmentation the Buddhadasa Indapanno's handwritten image into single characters is 89.16%. The image in group 1 is the most of correctness is 93.04% and the lower of correctness is image in group 2, there are 86.23 % because the characters in this image group is small and thick than characters in the other image groups. The classification of Buddhadasa Indapanno's handwritten into 5 groups with the mean of pixel image,

which are increased to the efficiency of pre-processing process. The efficiency of pre-processing is affected to be convenient in character segmentation.

The last step, the single characters are extracted the important feature with zoning method. These important features are data input in the character recognition process. The neural network with feed-forward back-propagation is used to recognize the Buddhadasa Indapanno's handwritten characters. The average of accuracy rate is 81.09%. The accuracy of character recognition results in the research is satisfactorily. Because the size of consonant in handwritten of Buddhadasa Indapanno document are difference, the lines are slant , and the characters are touched in horizontal and vertical direction, which is difficult to segment the image into single characters.

6.2 Limitations

The handwritten of Buddhadasa Indapanno images are used in this research is a Pali card, which consisted of the Thai and Pali-Sanskrit. The style of Buddhadasa Indapanno is squiggly handwriting. The characters are touched in the horizontal and vertical direction and the line is slant and connected more than 1 line. Moreover, the character in each images are non-uniform appearance which are:

- The size of characters in each images are difference.
- The thick and thin characters in these images are difference.
- Some images, the size of lower vowel and consonant are not difference.

The above problems affect the error in character segmentation process. Furthermore, more noises are occurred in the binary image because the brightness of each image is difference. Therefore, this research must classify these images into 5 groups to reduce noise efficiently.

The error of this research is show in Figure 6.1, which is a sample of touching characters are not segmented in this systems.

Figure 6.1 The sample of character segmentation error

From the error of character segmentation, that is affected to accuracy rate in the character recognition process.

6.3 Recommendation and Future Work

The technique in pre-processing is used in this research, is a technique to solve the problem with the thin character in ancient documents, there are not occurred the broken character after convert the document image to binary image.

The proposed techniques are incorrectly to segment the complicated touching characters. The error of incorrectly segmentation is affected to accuracy rate in character recognition process.

The Line Position Detection process will be suitable for the straight line or adjusted the image to straight. The technique to adjust the line may use the detection the slant angle and rotate the image in the correct position.

The technique to extract the feature of characters is an important process. Due to, the result from this process is effected to the character recognition process. Then, to find the suitable of feature extraction method should be compare the efficiency of zone analysis and the other method such as Frequency Domain.

The dictionary of Thai and Pali will be used to increase the performance in character recognition process.

REFERENCES

1. J. Pradeep, E. Srinivasan, and S. Himavathi. Neural Network based Handwritten Character Recognition system without feature extraction. International Conference on Computer, Communication and Electrical Technology – ICCCET 2011, 18 -19 March, 2011.
2. Phoorat Pakawathana. Thai Character Recognition on Tax Form. M.Sc. (Computer Science. Mahidol University, 2009
3. Chomtip Pornpanomchai, Verachad Wongsawangtham, Satheanpong Jeungudomporn, and Nannaphat Chatsumpun. Thai Handwritten Character Recognition by Genetic Algorithm (THCRGA). IACSIT International Journal of Engineering and Technology. Vol.3, No.2, Apil 2011
4. Oivind Due Trier, Anil K. Jain, and Torfinn Taxt. Feature Extraction Method for Character Recognition – A survey. Pattern Recogniyion, Vol.29, No. 4, pp. 641 – 662, 1996.
5. Wikipedia. Optical character recognition – Wikipedia, the free encyclopedia,2008. [Online; accessed 8 – May -2008].
6. Kurt Alfred kluever. Breaking the PayPal HIP: A Comparison of Classifiers. Document and Pattern Recognition LabDepartment of Computer ScienceRochester Institute of TechnologyRochester, New York, USA
7. Nafiz Arica, and Fatos T. Yarman – Vural. Optical Character Recognition for Cursive Handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, no. 6, June 2000.
8. R. Plamondon and S. N. Srihari. On-line and off-line handwritten character recognition : A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no. 1, pp. 63-84, 2000.
9. Thai OCR: Thai Optical Character Recognition [cited 2010 Dec. 12]. Available from: http:// thaiocr.phaisarn.com /category. html

10. Supachai Tangwongsan, and Chaivatna Sumetphong. Optical Character Recognition Techniques for Restoration of Thai Historical Documents. International Conference on Computer and Electrical Engineering, 2008
11. Rafael C. Gonzalez, and Richard E. Woods. Digital Image Processing, Third edition. Pearson Prentice Hall, Upper Saddle River, New Jersey.,2010
12. Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. Digital Image Processing using MATLAB, Third edition. Pearson Prentice Hall, Upper Saddle River, New Jersey.
13. Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image Processing, Analysis, and Machine Vision ,Second edition. Brooks/Cole Publishing Company, Pacific Grove, California.
14. Guoping Qiu. An Improved Recursive Median Filtering Scheme for Image Processing. IEEE Transactions on Image Processing, Vol. 5, NO. 4, APRIL 1996
15. Khalid Saeed, and Majida Albakoor. Region growing based segmentation algorithm for typewritten and handwritten text recognition. Applied Soft Computing 9 (2009)
16. M. Blumenstein, and B. Verma. Neural – based solutions for the Segmentation and Recognition of Difficult Handwritten Words from a Benchmark Database. Fifth International Conference on Document Analysis and Recognition (ICDAR'99)
17. Olarik Surinta and Rapeeporn Chamchong. Image Segmentation of Historical Handwriting from Palm Leaf Manuscripts. IFIP Internation for Information Processing, Vol. 288, pp. 182 – 189
18. Douglas J. Kennard, and William A. Barrett. Separating Lines of Text in Free – Form Handwritten Historical Documents. Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)
19. Richard G. Casey and Eric Lecolinet. A Survey of Methods and Strategies in Character Segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7, July 1996

20. N. Tripathy and U. Pal. Handwriting Segmentation of Unconstrained Oriya Text. Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR – 9 2004)
21. Yi Lu. Machine Printed Character Segmentation– An Overview. Pattern Recognition, Vol.28, No.1, pp.67 – 80, 1995
22. Berrin Yanikoglu, and Peter A. Sandon. Segmentation of Off – Line Cursive Handwriting using Linear Programming. Pattern Recognition. Vol.31, No.12, pp. 1825 – 1833,1998
23. Nafiz Arica and Fatos T. Yarman – Vural. An Overview of Character Recognition Focused on Off- Line Handwriting. IEEE Transactions on Systems, Man and Cybernetics – Part C: Application and Reviews, Vol. 31, No.2, MAY 2001
24. Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. Text line segmentation of historical documents: a survey. IJDAR ,2007
25. Prabir Bhattacharya. Connected component labeling for binary images on a reconfigurable mesh architecture. Journal of Systems Architecture 42 (1996) pp. 309 - 313.
26. Liang XU, Fei YIN, and Cheng – Lin LIU. Touching Character Splitting of Chinese Handwriting using Contour Analysis and DTW. Pattern Recognition (CCPR), Chinese Conference ,2010
27. E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. Kokkinakis. An Integrated System for Handwritten Document Image Processing. Internation Journal of Pattern Recognition and Antificial Intelligence. Vol. 17, No.4, 2003
28. Sergios Theodoridis, and Konstantinos Koutroumbas. Pattern Recognition, Fourth edition. Elsevier Academic Press, London,2009
29. Sing-Tze Bow. Pattern Recognition and Image Preprocessing, Second Edition. Marcel Dekker, New York, 2002.
30. J. Pradeep, E.Srinivasan, and S.Himavathi. Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System using Neural Network. International Journal of Computer Science & Information Technology (IJCSIT). Vol. 3, No. 1, Feb 2011
31. Tom M. Mitchell,Machine learning. McGraw-Hill, New York,1997

32. N.P. Padhy. Artificial Intelligence and Intelligent Systems. Oxford University Press, New York, 2005
33. Gradient descent momentum and adaptive learning rate. [cited 2011 May. 05]. Available from: http://www.mathworks.com/help/nnet/ref/traingdx.html
34. S.V. Rajashekararadhya, and Dr. P. Vanaja Ranjan. Handwritten Numeral/Mixed Numerals Recognition of South-Indian Scripts: The Zone-Based Feature Extraction Method. Journal of Theoretical and Applied Information Technology, Vol. 7, No. 1, pp. 063 – 079, 2005-2009
35. Omaima N. Ahmad AL-Allaf, Abdelfatah Aref Tamimi, and Shahlla A. AbdAlKader. Artificial Neural Networks for Iris Recognition System: Comparisons between Different Models, Architectures and Algorithms. International Journal of Information and Communication Technology Research, Vol. 2, No. 11, Nov.2012
36. Sigmoid Activation Function [cited 2011 Apr. 29]. Available from: http://annet.eeng.nuim.ie/intro/course/chpt3/sigmoid.shtml
37. Fundamentals of Neural Networks [cited 2013 May. 14]. Available from: http://www.yukool.com/nn/fundamental.htm

APPENDICES

APPENDIX A
CHARACTER CODES

The character outputs are classified into 87 classes. These classes are 87 nodes of output layers in Neural Network model.

Class	Output	Character
1	1000000000

Class	Output	Character
17	00000000000000000100000000000000000

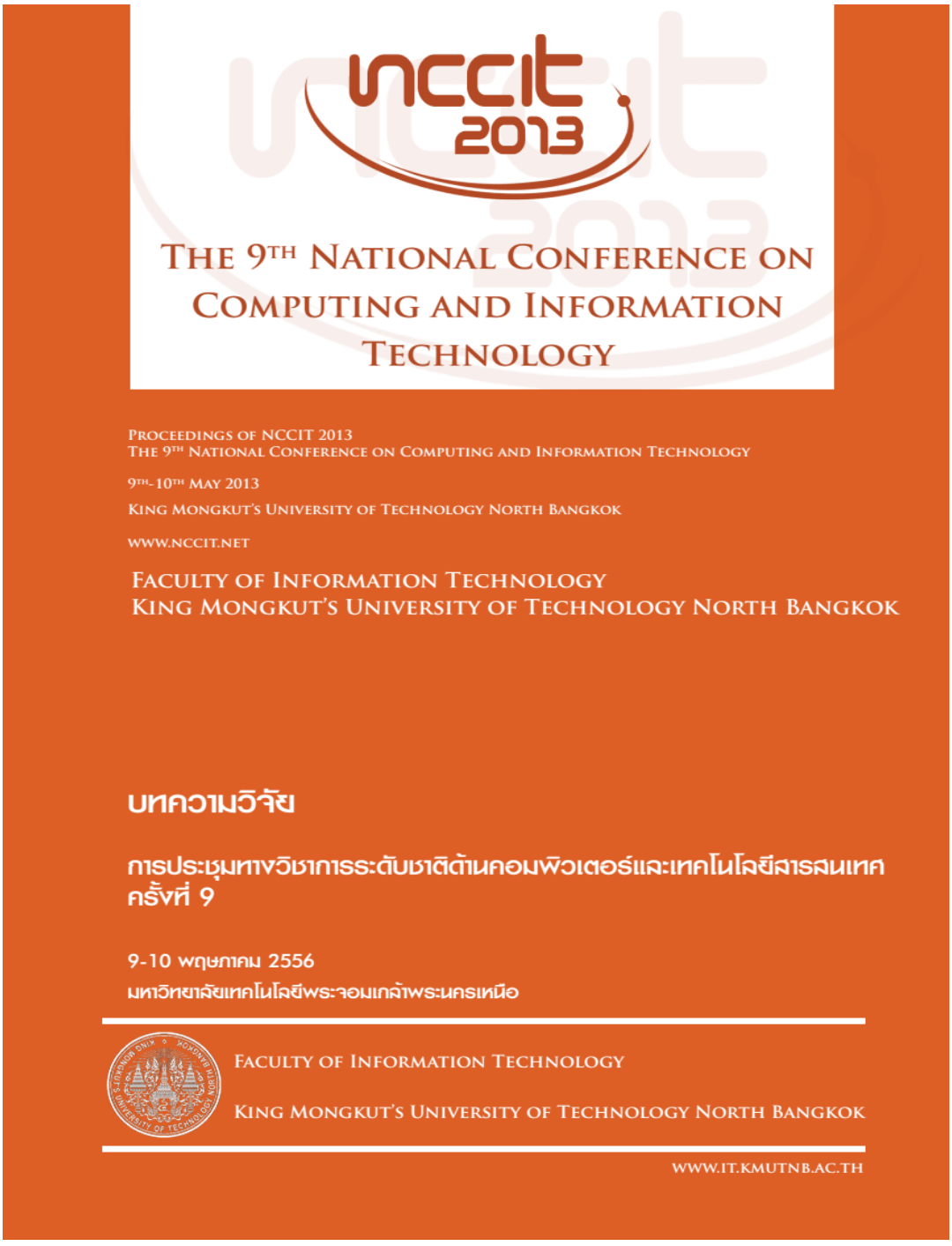
Class	Output	Character
33	00000000000

Class	Output	Character
47	00000000000

Class	Output	Character
62	00000000000

Class	Output	Character
77	00000000000

APPENDIX B
PUBLICATION



The 9th National Conference on Computing and Information Technology NCCIT2013

การตัดตัวอักษรลายมือเขียนจากเอกสารภาพของท่านพุทธทาส อินทปัญโญ
Handwritten Character Segmentation on Buddhadasa Indapanno Documents

วันวิสาข์ ชีวกุลมงคล (Wanwisa Chiwakulmongkhon)[1] และธนัสนี เพียรระกูล(Tanasanee Phientrakul)[2]
[1]สาขาวิชาเทคโนโลยีการจัดการระบบสารสนเทศ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล
[2]ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล
[1]tism.wanwisa@gmail.com, [2]tanasanee.phi@mahidol.ac.th

บทคัดย่อ

กระบวนการตัดตัวอักษร เป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งในการเพิ่มความถูกต้องให้กับระบบรู้จำตัวอักษร งานวิจัยนี้ได้นำเสนอวิธีการตัดตัวอักษรลายมือของท่านพุทธทาอินทปัญโญ โดยการประยุกต์ใช้วิธีการกำหนดสัญลักษณ์การเชื่อมต่อขององค์ประกอบ (Connected Component Labeling) และ วิธีการโปรเจคชั่น(Projection Profile) โดยผลลัพธ์จากการตัดตัวอักษรด้วยเทคนิคที่นำเสนอนี้มีความถูกต้องเฉลี่ย89.16%และเทคนิคที่นำเสนอนี้สามารถนำไปประยุกต์ใช้กับภาพตัวอักษรลายมืออื่นๆต่อไปได้

คำสำคัญ: การกำหนดสัญลักษณ์การเชื่อมต่อองค์ประกอบ การโปรเจคชั่น การตัดตัวอักษรลายมือ

Abstract

The character segmentation processes is a most necessary and important step for character recognition process. The correctness of character segmentation is affected to the accuracy of recognition. This paper proposes an algorithm for Buddhadasa Indapanno's handwritten character segmentation. This algorithm is based on the connected component labeling and projection profile. The correctness of this algorithm is 89.16%. Moreover, this algorithm may be applied to other character images

Keyword: Connected Component Labeling, Projection Profile, Handwritten Character Segmentation

1. บทนำ

กระบวนการตัดตัวอักษรเป็นกระบวนการหนึ่งที่มีความสำคัญและมีผลต่อความถูกต้องในกระบวนการรู้จำตัวอักษร ซึ่งในปัจจุบันงานวิจัยจำนวนมาก ได้นำเสนอเทคนิคการตัดตัวอักษร เพื่อแก้ปัญหาที่เกิดขึ้นกับตัวอักษรลายมือ อาทิ เช่น การติดกันของตัวอักษรในแนวนอน แนวตั้ง หรือการติดติดกันของตัวอักษร ทั้งในแนวนอน และแนวตั้ง และ รวมถึงการลาดเอียงของบรรทัดข้อความ ซึ่งเทคนิคการตัดตัวอักษรต่างๆที่จะถูกเลือกนำมาใช้ให้เหมาะสมกับเอกสารลายมือนั้น ก็ขึ้นอยู่กับรูปแบบการเขียนของผู้เขียนแต่ละคน

ในงานวิจัยนี้ได้นำเอกสารภาพลายมือของท่านพุทธทาส อินทปัญโญ มาใช้ในการศึกษาเทคนิคการตัดแยกตัวอักษรลายมือแบบออฟไลน์ โดยการประยุกต์ใช้วิธี การกำหนดสัญลักษณ์การเชื่อมต่อขององค์ประกอบ (Connected component labeling) และ วิธีการโปรเจคชั่น (projection profile) เนื่องจาก หลักธรรม คำสอนของท่านพุทธทาส ล้วนแต่เป็นผลงานที่ทรงคุณค่า สามารถนำมาปรับ และประยุกต์ใช้ได้อย่างเหมาะสมกับคนทุกยุค ทุกสมัย และทุกเหตุการณ์ อีกทั้งยังได้รับการยกย่องจาก องค์การศึกษาวิทยาศาสตร์และวัฒนธรรมแห่งสหประชาชาติ หรือ UNESCO ให้เป็นบุคคลสำคัญของโลก ในด้านส่งเสริมขันติธรรม สันติธรรม วัฒนธรรม ความสัมพันธ์ของโลก และความเข้าใจอันดีของมวลมนุษย์ โดยเอกสารภาพลายมือ ที่นำมาใช้ในการศึกษาวิจัยนี้ ได้รับความอนุเคราะห์ข้อมูลมาจาก หอจดหมายเหตุพุทธทาส อินทปัญโญ ซึ่งในเบื้องต้นได้เลือกภาพลายมือ ที่อยู่ในหมวด "*อฏฺกนิปาต องฺคุตฺตรนิกาย*" ในการเริ่มต้นศึกษา

2. วรรณกรรมที่เกี่ยวข้อง

โดยทั่วไปการตัดแยกตัวอักษรจะประกอบด้วยกระบวนการหลักๆ 2 กระบวนการ คือ กระบวนการประมวลผลภาพเบื้องต้น และ กระบวนการตัดแยกตัวอักษร ซึ่งกระบวนการประมวลผลภาพเบื้องต้นนั้นมีมามายหลายวิธี อาทิเช่น การปรับปรุงคอนทราส (Enhance contrast) การกำจัดสัญญาณรบกวน (Noise Reduction) และ การแปลงภาพให้เป็นภาพไบนารี่ (Binarization) เป็นต้น ซึ่งวิธีการปรับปรุงภาพที่ถูกเลือกใช้ในแต่ละงานวิจัยนั้นโดยส่วนมากจะถูกเลือกใช้เพื่อให้เหมาะสมกับภาพแต่ละประเภท ส่วนกระบวนการตัดแยกตัวอักษร โดยทั่วไปแล้วจะประกอบด้วย 2 กระบวนการคือ การตัดแยกภาพออกเป็นบรรทัดข้อความ (Line Segmentation) และ การตัดแยกตัวอักษรออกเป็นตัวอักษรเดี่ยว(Character Segmentation) [1], [2] และ [3]

Surinta และ Chamchong [1] ได้นำเสนอการตัดตัวอักษรลายมือจากเอกสารใบลาน โดยการใช้ เทคนิคตัวกรองแบบเกาส์เซียน (Gaussian Filter) ในการกำจัดสัญญาณรบกวนที่เกิดขึ้นกับภาพหลังจากแปลงภาพสีไปเป็นภาพระดับเทา และ ใช้วิธี Otsu's algorithm ในการหาค่า เธรดโชลด์ (Threshold) ที่เหมาะสม เพื่อแปลงภาพระดับเทาให้เป็นภาพไบนารี่โดยการแยกพื้นหลังออกจากตัวอักษร (background)และใช้วิธีโปรเจคชั่น (projection profile) ในการตัดแยกภาพออกเป็นบรรทัดข้อความ ก่อนตัดแยกบรรทัดข้อความออกเป็นตัวอักษรด้วยวิธีการทำ โปรเจคชั่น

Ntogas Nikolaos และ Ventzas Dimitrios [4] ได้นำเสนอ วิธีการแปลงภาพให้เป็นภาพไบนารี่ (binarization algorithm) สำหรับเอกสารลายมือทางประวัติศาสตร์ ด้วยการแยกข้อความออกจากพื้นหลัง โดยใช้พื้นฐานของ การทำเธรสโชลด์ และ ตัวกรอง (Filtering) ในการประมวลผลภาพ ซึ่งวิธีการทำเธรสโชลด์ ด้วยวิธี Otsu's นั้น ให้ผลลัพธ์ของภาพไบนารี่ ดีที่สุดกับภาพในกลุ่มที่มีลักษณะของเส้นตัวอักษรที่บาง (thin strokes) ซึ่งจากงานวิจัยที่ [1] และ [4] แล้ว ยังมี งานวิจัยที่ [5] นำเสนอโดย J. Pradeep E. Srinivasan และ S.Himavathi ได้ใช้วิธีการแยกตัวอักษรออกจากพื้นหลังโดยใช้วิธีการทำเธรสโชลด์ ด้วยวิธี Otsu's เช่นกัน ดังนั้น จะเห็นได้ว่าวิธีแยกตัวอักษรออกจากพื้นหลัง โดยการทำเธรสโชลด์ ด้วยวิธี Otsu's เป็นวิธีที่ได้รับความนิยม ส่วนในการตัดแยกตัวอักษรด้วยวิธีโปรเจคชั่นนั้นพบว่ายังมีข้อเสียอยู่บ้างเนื่องจากวิธีการนี้ไม่เหมาะสมกับภาพตัวอักษรลายมือที่ไม่มีช่องว่างระบรรทัด หรือตัวอักษร[6] แต่ในงานวิจัยนี้ได้นำวิธีการทำโปรเจคชั่น มาใช้ร่วมกับการทำตัวอักษรให้บาง (thin image) จึงทำให้จุดในการตัดแยกตัวอักษรได้ชัดเจนยิ่งขึ้น

3. การตัดแยกตัวอักษรลายมือเขียน

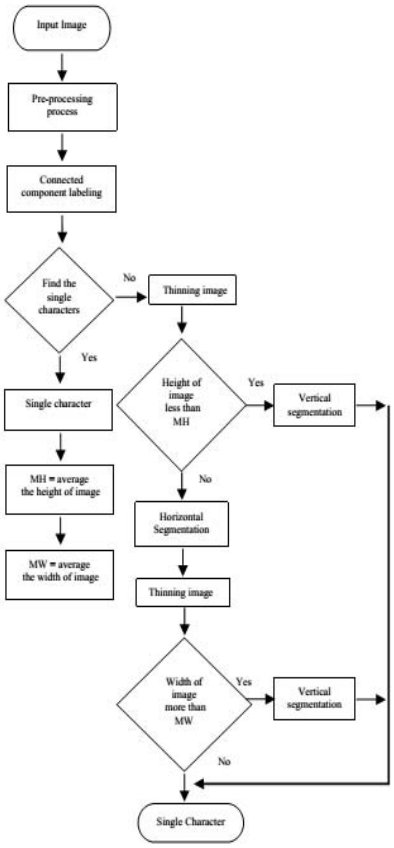
กระบวนการตัดตัวอักษรลายมือออกจากเอกสารภาพลายมือ ของท่านพุทธทาส อินทปัญโญนั้น ประกอบด้วย 2 ขั้นตอนหลักๆ คือ ขั้นตอนการประมวลผลภาพ และ กระบวนการตัดแยกตัวอักษร แสดงดังภาพที่ 1

3.1 กระบวนการประมวลผลภาพ (Pre-processing process)

เนื่องจากภาพลายมือที่นำมาใช้ในการทำวิจัยนี้ เป็นภาพสีโดยแต่ละภาพนั้นมีความมืด และความสว่างของภาพที่แตกต่างกัน ดังนั้นในขั้นตอนนี้จึงเป็นขั้นตอนปรับปรุงภาพแต่ละภาพให้มีความเหมาะสม และช่วยเพิ่มความถูกต้องให้กับกระบวนการตัดแยกตัวอักษร ซึ่งในขั้นตอนนี้ ประกอบด้วย 4 ขั้นตอนดังนี้

3.1.1 กระบวนการตัดกรอบรูปภาพ (Crop Image)

ภาพลายมือจะถูกตัดขอบภาพให้เหลือเฉพาะบริเวณที่เป็นตัวอักษรเท่านั้น หลังจากนั้นภาพจะถูกนำมาคำนวณเพื่อหาค่าเฉลี่ยของพิกเซล (pixel) ซึ่งค่าเฉลี่ยที่ได้จะถูกนำมาใช้เป็นหลักเกณฑ์ในการจัดกลุ่มภาพ โดยภาพที่มีลักษณะคล้ายๆกันจะถูกจัดอยู่ในกลุ่มเดียวกันซึ่งสามารถแบ่งได้เป็น 5 กลุ่มคือ



ภาพที่ 1: ภาพรวมของระบบการตัดตัวอักษรลายมือ ออกจากเอกสารภาพลายมือของท่านพุทธทาสอินทปัญโญ

กลุ่มที่ 1 เป็นกลุ่มของภาพที่มีรอยเปื้อนกระจายอยู่ทั่วบริเวณภาพ อีกทั้งยังเป็นภาพที่มีความสว่างของภาพน้อยที่สุด

กลุ่มที่ 2 โดยภาพในกลุ่มนี้จะมีขนาดของตัวอักษรที่เล็กและหนา แต่มีความสว่างของภาพที่สม่ำเสมอ

กลุ่มที่ 3 ตัวอักษรของภาพในกลุ่มนี้ มีลักษณะที่อ่านง่าย เนื่องจากตัวอักษรมีขนาดและความหนาที่เหมาะสม

กลุ่มที่ 4 ลักษณะของภาพในกลุ่มนี้เป็นภาพที่ประกอบด้วยตัวอักษรที่มีทั้งตัวหนา และบาง ปะปนกันภายในภาพ

กลุ่มที่ 5 ลักษณะของภาพในกลุ่มนี้เป็นภาพที่ประกอบด้วยตัวอักษรที่มีลายเส้นที่บาง ไม่ชัดเจน

3.1.2 กระบวนการปรับปรุงคอนทราส (Adjustment Contrast)

ภาพหลังจากถูกตัดให้เหลือเฉพาะบริเวณข้อความตัวอักษรแล้ว จะถูกนำมาปรับปรุงคอนทราสของภาพตามกลุ่มที่ได้แบ่งไว้ เนื่องจากภาพในแต่ละกลุ่มมีลักษณะของความมืดและความสว่างของภาพที่แตกต่างกัน ดังนั้นเพื่อให้ภาพแต่ละภาพยังคงรายละเอียดของตัวอักษรในภาพให้มีความสมบูรณ์มากที่สุด ในงานวิจัยนี้จึงได้เลือกทำการปรับปรุงคอนทราสของภาพตามกลุ่มภาพ ก่อนที่จะแปลงภาพให้เป็นภาพในระดับเทา (Grey image) ด้วยวิธีการยืดคอนทราส (Contrast Stretching) [7] โดยการกำหนดค่าให้เหมาะสมกับภาพในแต่ละกลุ่ม

3.1.3 กระบวนการแปลงภาพไบนารี่ (Binarization)

ภาพในระดับเทาที่ได้หลังจากกระบวนการปรับปรุงคอนทราส จะถูกแปลงไปเป็นภาพไบนารี่ (Binary Image) ด้วยสมการที่ 1

$$g(x,y) = \begin{cases} 1 ; f(x,y) \neq 0 \\ 0 ; f(x,y) = 0 \end{cases} \quad (1)$$

เมื่อ $g(x,y)$ คือ ค่าพิกเซลในแต่ละตำแหน่ง (x,y) ของภาพใหม่ และ $f(x,y)$ คือ ค่าพิกเซลในแต่ละตำแหน่ง (x,y) ของภาพเดิม โดยค่าพิกเซล เท่ากับ 1 แทนพื้นหลัง(สีขาว) และ ค่าพิกเซล เท่ากับ 0 แทน ตัวอักษร (สีดำ)

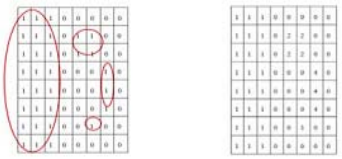
3.1.4 กระบวนการกำจัดสัญญาณรบกวน (Noise Reduction)

เมื่อภาพถูกแปลงจากภาพระดับเทาไปเป็นภาพไบนารี่ จะมีสัญญาณรบกวน (Noise) เกิดขึ้น ดังนั้นเพื่อกำจัดสัญญาณรบกวนที่เกิดขึ้นบนภาพไบนารี่ ในงานวิจัยนี้จึงได้เลือกใช้ ตัวกรองแบบมัธยฐาน (Median Filter) โดยใช้หน้าต่างขนาด 3×3

3.2 กระบวนการกำหนดสัญลักษณ์การเชื่อมต่อขององค์ประกอบ (Connected Component Labeling)

วิธีการกำหนดสัญลักษณ์การเชื่อมต่อขององค์ประกอบ เป็นการกำหนดสัญลักษณ์ให้กับตัวอักษรที่มีองค์ประกอบต่อเนื่องกัน ซึ่งในงานวิจัยนี้ได้กำหนดการเชื่อมต่อขององค์ประกอบแบบ 4-connected แสดงดังภาพที่ 2 โดย ภาพที่ 2

(ก) แสดงภาพไบนารี่ที่มีการเชื่อมต่อขององค์ประกอบ แบบ 4 – connected ส่วนภาพที่ 2 (ข) แสดงการกำหนดสัญลักษณ์ ที่สอดคล้องกับภาพ 2 (ก) [7]



(ก) ภาพไบนารี่ ที่มีการเชื่อมต่อของ องค์ประกอบแบบ 4 - connected

(ข) การกำหนดสัญลักษณ์แบบ 4 - connected

ภาพที่ 2: แสดงภาพไบนารี่ แบบ 4-components [7]

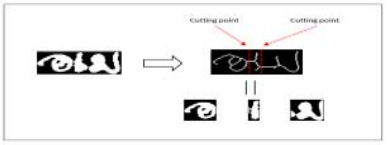
ซึ่งในขั้นตอนนี้ ภาพตัวอักษรลายมือจะถูกตัดแบ่งออกเป็นตัวอักษรเดี่ยว ตัวอักษรที่ติดกัน และชิ้นส่วนที่ไม่ใช่ตัวอักษร (อาทิเช่น เส้นขอบ หรือ สัญญาณรบกวนที่ยังคงเหลืออยู่ในภาพ) ปะปนกันอยู่ โดยในการแยกตัวอักษรเดี่ยว ออกจากตัวอักษรที่ติดกัน และชิ้นส่วนที่ไม่ใช่ตัวอักษรนั้น จะใช้ค่าความสูง และ ความกว้างของภาพตัวอักษร รวมถึง สัดส่วนของพิกเซลสีขาว และพิกเซลสีดำ (proportion of black and white pixel) มาเป็นเงื่อนไขในการแบ่งแยก โดยภาพตัวอักษรเดี่ยวที่ได้จากขั้นตอนนี้จะถูกนำไปหาค่าเฉลี่ยความสูง และความกว้างของตัวอักษร ตัวอักษรที่ติดกันจะถูกนำไปตัดแยกตัวอักษรเพื่อให้เหลือเฉพาะตัวอักษรเดี่ยวในกระบวนการโปรเจคชั่น และชิ้นส่วนที่ไม่ใช่ตัวอักษร จะถูกกำจัดทิ้ง

3.2.1 กระบวนการโปรเจคชั่น (Projection profile)

ภาพตัวอักษรที่ติดกันจะถูกตัดแยกให้เป็นตัวอักษรเดี่ยวในขั้นตอนนี้ ซึ่งก่อนที่ภาพตัวอักษรที่ติดกันจะถูกตัดแยกตัวอักษรให้เป็นตัวอักษรเดี่ยวนั้น ภาพตัวอักษรที่ติดกันจะถูกทำให้บาง (Thinning image)ก่อน จากนั้นนำมาหาผลรวมของพิกเซลในแต่ละแถว เพื่อใช้ในการพิจารณาหาจุดตัดที่เหมาะสม

โดยในเบื้องต้น ภาพตัวอักษรที่ติดกันจะถูกแบ่งออกเป็น 2 กลุ่ม ใหญ่ๆ คือ

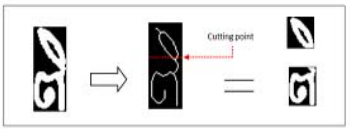
- กลุ่มของภาพที่มีความสูง น้อยกว่าความสูงเฉลี่ยของตัวอักษรเดี่ยว ซึ่งภาพในกลุ่มนี้โดยส่วนใหญ่จะเป็นภาพที่ติดกันในแนวนอน เพียงอย่างเดียว ซึ่งจะทำการตัดแยกตัวอักษรแค่ในทิศทางแนวตั้ง (Vertical Projection) เพียงทิศทางเดียว ซึ่งแสดงดังภาพที่ 3



ภาพที่ 3: ตัวอย่างการตัดภาพตัวอักษรที่ติดกันในแนวตั้ง

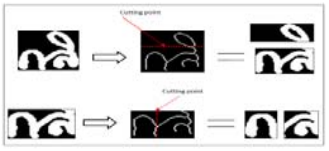
- กลุ่มของภาพที่มีความสูง มากกว่าความสูงเฉลี่ยของตัวอักษรเดี่ยว โดยภาพที่อยู่ในกลุ่มนี้จะมีทั้งตัวอักษรที่ติดกันทั้งในแนวตั้ง เพียงอย่างเดียว หรือ ติดกันทั้งในแนวตั้ง และแนวนอน ซึ่งภาพในกลุ่มนี้จะถูกตัดแยกตัวอักษรในแนวตั้งก่อน (Horizontal Projection) แล้วหลังจากนั้น ภาพที่ถูกตัดแล้วจะถูกนำมาพิจารณาเปรียบเทียบระหว่างความกว้างของตัวอักษร กับค่าเฉลี่ยความกว้างของตัวอักษรเดี่ยวอีกครั้ง โดย

- ภาพตัวอักษรที่มี ความกว้างของตัวอักษร น้อยกว่าความกว้างเฉลี่ยของตัวอักษรเดี่ยว ซึ่งโดยส่วนมากภาพที่อยู่ในเงื่อนไขนี้ จะเป็นตัวอักษรเดี่ยว ดังแสดงในภาพที่ 4



ภาพที่ 4: ตัวอย่างการตัดภาพตัวอักษรที่ติดกันในแนวนอน

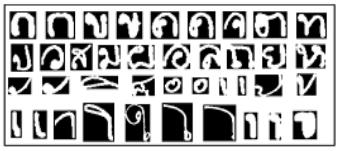
- ภาพตัวอักษรที่มีความกว้างของตัวอักษร มากกว่าความกว้างเฉลี่ยของตัวอักษรเดี่ยว จะถูกนำมาตัดแยกตัวอักษรในแนวตั้งอีกครั้ง (Vertical Projection) ดังภาพที่ 5



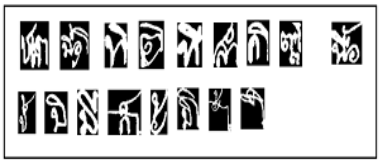
ภาพที่ 5: ตัวอย่างการตัดภาพตัวอักษรที่ติดกัน ทั้งในแนวนอน และแนวตั้ง

4. ผลการทดลอง

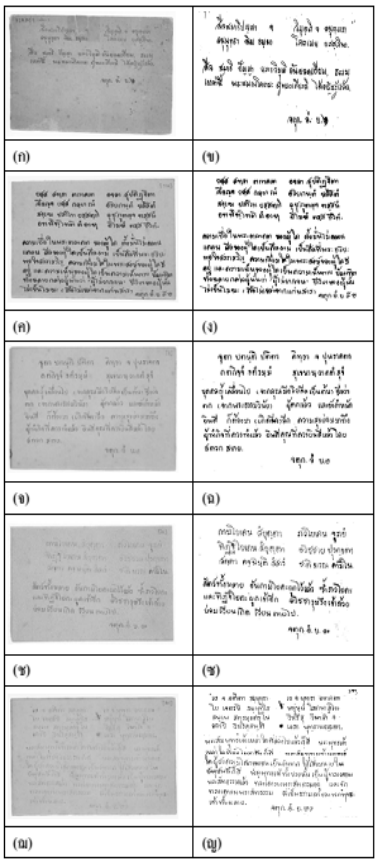
การตัดตัวอักษรลายมือออกจากเอกสารภาพลายมือของท่านพุทธทาส อินทปัญโญ ได้ใช้โปรแกรม MATLAB เวอร์ชั่น R2010a ในการดำเนินการ โดยจำนวนภาพที่นำมาใช้ในงานวิจัยนี้มีทั้งสิ้น 148 ภาพ ซึ่งเป็นภาพสี ที่ถูกสแกนเก็บด้วยความละเอียด 300 จุดต่อนิ้ว และบันทึกเก็บเป็นไฟล์ที่มีนามสกุล .TIFF โดยสามารถตัดแยกเป็นตัวอักษรได้ 34,020 ตัวอักษร โดยมีเปอร์เซนต์ความถูกต้องเฉลี่ย ของการตัดแยกตัวอักษร เท่ากับ 89.16 % โดยตัวอย่างผลลัพธ์ของภาพที่ระบบสามารถตัดได้ถูกต้องแสดงดังภาพที่ 6 และภาพที่ 7 แสดงภาพตัวอย่างที่ระบบตัดผิดพลาด ส่วน ผลลัพธ์ของภาพไบนารี่ ที่ได้จากกระบวนการประมวลผลภาพ (Pre-processing process) ซึ่งจำแนกตามกลุ่มภาพ ได้แสดงดังภาพที่ 8 โดยภาพที่ 8 (ก), 8 (ค) , 8 (จ), 8 (ช) และ 8 (ฌ) คือ ตัวอย่างภาพต้นฉบับของภาพในกลุ่มที่ 1,2 ,3 ,4 และ 5 ตามลำดับ ส่วนภาพ ที่ 8 (ข), 8 (ง), 8 (ฉ), 8 (ซ) และ 8 (ญ) คือตัวอย่างของภาพไบนารี่ของภาพต้นฉบับในกลุ่มที่ 1,2 ,3 ,4 และ 5 ตามลำดับตัว



ภาพที่ 6: ตัวอย่างภาพตัวอักษรที่สามารถตัดได้ถูกต้อง



ภาพที่ 7: ตัวอย่างภาพตัวอักษรที่ระบบไม่สามารถตัดได้



(ก)	(ข)
(ค)	(ง)
(จ)	(ฉ)
(ช)	(ซ)
(ฌ)	(ญ)

ภาพที่ 8: แสดงผลลัพธ์ของภาพไบนารี่ โดยจำแนกตามกลุ่มภาพ

ตารางที่ 1: แสดงจำนวนตัวอักษรที่ตัดได้ในแต่ละกลุ่มภาพ

กลุ่มภาพ	จำนวนเอกสารภาพ	จำนวนตัวอักษรที่ตัดได้
1	1	158
2	19	5,395
3	74	17,262
4	47	9,686
5	7	1,519
ผลรวม	148	34,020

ตารางที่ 2: เปอร์เซนต์ความถูกต้องของการตัดแยกตัวอักษร

กลุ่มภาพ	ความถูกต้องของการตัดแยกตัวอักษร		เปอร์เซนต์ความถูกต้อง
	ถูกต้อง	ผิดพลาด	
1	147	11	**93.04%**
2	4,782	623	86.23%
3	15,073	2,189	87.14%
4	8,807	879	88.84%
5	1,351	168	90.57%
ค่าเฉลี่ย			89.16%

จากตารางที่ 1 และ ตารางที่ 2 ได้แสดงจำนวนตัวอักษรที่ตัดได้ทั้งหมด และ ผลความถูกต้องที่ได้จากการตัดแยกตัวอักษรในแต่ละกลุ่มภาพ ตามลำดับ

5. สรุปผล

จากการตัดตัวอักษรลายมือออกจากเอกสารภาพลายมือของท่านพุทธทาส อินทปัญโญ นั้นพบว่า ภาพไบนารี่ที่ได้จากในกระบวนการประมวลผลภาพนั้น ให้ผลเป็นที่น่าพอใจ โดยระบบสามารถแก้ปัญหาที่เกิดขึ้นกับภาพลายมือได้ทุกภาพ อีกทั้งยังคงรายละเอียดของตัวอักษรได้อย่างครบถ้วน เนื่องจากไม่เกิดแตกหักของตัวอักษร หลังการแปลงภาพระดับเทา ไปเป็นภาพไบนารี่ ส่วนกระบวนการตัดแยกตัวอักษรนั้นได้ประยุกต์ใช้ วิธีการกำหนดสัญลักษณ์การเชื่อมต่อขององค์ประกอบ (Connected Component Labeling) และวิธีการโปรเจคชั่น (Projection Profile) โดยมีความถูกต้องในการตัดแยกตัวอักษรเฉลี่ย เท่ากับ 89.16 % ซึ่งภาพตัวอักษรลายมือในกลุ่มที่ 1 คือกลุ่มภาพที่มีรอยเปื้อน บริเวณพื้นหลัง และ มีความสว่างของภาพน้อยที่สุดนั้น มีความถูกต้องในการตัดแยกตัวอักษรมากที่สุด คือ 93.04% รองลงมาคือภาพในกลุ่มที่ 5 คือ กลุ่มภาพที่มีลักษณะตัวอักษรที่บาง โดยมีความถูกต้องอยู่ที่ 90.57%

เอกสารอ้างอิง

[1] Surinta, O. and Chamchong, R., "Image Segmentation of Historical Handwriting from Palm Leaf Manuscripts," *IFIP International Federation for Information Processing*, Vol. 288, pp. 182-189, 2008.

[2] Rejean Plamondon, and Sargur N. Srihari, "On – Line and Off – Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, January 2000,

[3] Berrin Yanikoglu, and Peter A. Sandon, "Segmentation of Off – Line Cursive Handwriting Using Linear Programming," *Pattern Recognition*, Vol. 31, No. 12, pp. 1825 – 1833, 1998.

[4] Ntogas Nikolaos and Ventzas Dimitrios, "A Binarization Algorithm for Historical Manuscripts," *The 12 th WSEAS International Conference on Communications*, Heraklion, Greece, July 23 – 25, 2008.

[5] J. Pradeep, E. Srinivasan, and S.Himavathi, "Diagonal based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network," *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol. 3, No 1, Feb 2011.

[6] Zaidi Razak, Khansa Zulkiflee, Mohd Yamani Idna Idris, Emran Mohd Tamil, Mohd Noorzaily Mohamed Noor, Rosli Salleh, Mohd Yaakob @ Zulkifli Mohd Yusof, and Mashkuri Yaacob, "Off – line Handwriting Text Line Segmentation : Review," *IJCSNS Internatioonal Journal of Computer Science and Network Security*, Vol. 8, No. 7, July 2008.

[7] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, "*Digital Image Processing using Matlab*" Pearson Prentice Hall: Upper Saddle River, New Jersey, 2004.

BIOGRAPHY

NAME	Miss Wanwisa Chevakulmongkol
DATE OF BIRTH	15 May 1984
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Rajamangala University of Technology Krungthep, 2002-2006 Bachelor of Engineering (Textile Chemical Engineering) Mahidol University, 2008-2013 Master of Science (Technology of Information System Management)
HOME ADDRESS	202/139 moo 1, Theparak Road, Bangsaothong, Bangsaothong, Samuprakarn 10540 Tel 089-793-9239 Email: tism.wanwisa @gmail.com
PUBLICATION / PRESENTATION	NCCIT 2013 (National Conference on Computing and Information Technology 2013)