UNIVERSITY OF SOUTHAMPTON

A Systematic Study of Offline Recognition of Thai Printed and Handwritten Characters

by

Sutat Sae-Tang

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical and Applied Sciences
School of Electronics and Computer Science

November 15, 2011

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Sutat Sae-Tang

Thai characters pose some unique problems, which differ from English and other oriental scripts. The structure of Thai characters consists of small loops combined with curves and there is an absence of spaces between each word and sentence. In each line, moreover, Thai characters can be composed on four levels, depending on the type of character being written. This research focuses on OCR for the Thai language: printed and offline handwritten character recognition. An attempt to overcome the problems by simple but effective methods is the main consideration. A printed OCR developed by the National Electronics and Computer Technology Center (NECTEC) uses Kohonen self-organising maps (SOMs) for rough classification and back-propagation neural networks for fine classification. An evaluation of the NECTEC OCR is performed on a printed dataset that contains over 0.6 million tokens. Comparisons of the classifier, with and without the aspect ratio, and with and without SOMs, yield small, but statistically significant differences in recognition rate. A very straightforward classifier, the nearest neighbour, was examined to evaluate overall recognition performance and to compare with the classifier. It shows a significant improvement in recognition rate (about 98%) over the NECTEC classifier (about 96%) on both the original and distorted data (rotated and noisy), but at the expense of longer recognition times. For offline handwritten character recognition, three different classifiers are evaluated on three different datasets that contain, on average, approximately 10,000 tokens each. The neural network and HMMs are more effective and give higher recognition rates than the nearest neighbour classifier on three datasets. The best result obtained from the HMMs is 91.1% on ThaiCAM dataset. However, when evaluated on a different dataset, the recognition rates drastically reduce, due to differences in many aspects of online and offline handwritten data. An improvement in classification rates was obtained by adjusting the stroke width of a character in the online handwritten dataset (12 percentage points) and combining the training sets from the three datasets (7.6 percentage points). A boosting algorithm called AdaBoost yields a slight improvement in recognition rate (1.2 percentage points) over the original classifiers (without applying the AdaBoost algorithm).

Contents

List of Figures

List of Tables

DECLARATION OF AUTHORSHIP

I, Sutat Sae-Tang, declare that the thesis entitled `A Systematic Study of Offline Recognition of Thai Printed and Handwritten Characters` and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: .....
Date: .....

Acknowledgements

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I am sincerely and heartily grateful to my supervisors, Dr. John N Carter and Professor RI "Bob" Damper, for the support, guidance, patience and encouragement throughout my time at the University of Southampton; where I have learnt much about research, academia and myself. I am sure it would have not been possible without their help.

I would like to acknowledge the financial support of the Royal Thai Government and particularly everyone in the Office of Educational Affairs, Royal Thai Embassy in London for their assistance and administrative support. I am most grateful to all of my fellow colleagues at the ISIS research group and Thai friends for their kindness, friendship and support. Thanks for all of the nice conversations and for being such an integral part of my postgraduate school experience.

Finally, I am extremely indebted to my parents and siblings, who have supported and encouraged me through their kindness and affection, so that I could concentrate on my studies. Without their encouragement and support, it would have been impossible to complete this thesis.

For any errors or inadequacies that may remain in this work, of course, the responsibility is entirely my own.

To my parents and siblings

Chapter 1

Introduction

This research focuses on optical character recognition (OCR) for the Thai language. First, printed character recognition is considered, followed by offline handwritten character recognition. Thai serves as the official national language and the mother tongue of the Thai people. More than 65 million people speak and write Thai as their native language. It is the language used in schools, the media and government affairs. An attempt to overcome the problems of OCR by simple but effective methods is the main consideration. The remaining sections of this chapter are introduced below. The motivation for this research is presented in Section 1.1. An overview of OCR and its applications are introduced in the next section (1.2) and contributions and original work are described in Section 1.3. Finally, the outline of the thesis is presented in Section 1.4.

1.1 Motivation

Machine replication of human functions, for instance reading, is an ancient dream. Nowadays, machine reading has grown from dreams to reality. Character recognition has become one of the most successful technologies in the area of pattern recognition and artificial intelligence. There have been increasing demands for applications to process automatically the content of documents as a supplement to tasks performed 2by humans. Many commercial systems exist, although machines are still not able to compete with human reading capabilities.

Character recognition is generally concerned with the problem of recognising any forms of character (Mori *et al.*, 1992). The ultimate goal of character recognition research should be to develop systems that are able to read and understand any text with the same or better recognition accuracy than humans, but at a faster rate (Lorette, 1999).

Character recognition is a difficult task, because it is generally associated with several problems across several research areas. Image processing, for example, is required for

(a)

(b)

FIGURE 1.1: Examples of the Thai phrase: (a) printed style (AugsanaUPC font), (b) handwritten style.

solving the problems of low quality and the noise in input documents, while knowledge of artificial intelligence and machine learning are needed to tackle the classification problem. Moreover, linguistic knowledge is essential for improving performance in post-processing. There have been many character recognition research projects attempting to tackle different aspects of the problem (Mori *et al.*, 1992; Miletzki, 1997; Arica and Yarman-Vural, 2001). The performance of character recognition systems has improved dramatically.

In the past, most character recognition research has been centred on the Roman alphabet. Over the past two decades, the demands for research in other languages, especially Asian languages, such as Chinese, Japanese and Arabic, have increased. The Thai language, the official language of Thailand, is among the many Asian languages that have experienced increasing interest, as seen from the dramatic growth and increasing reports on Thai language processing (Sornlertlamvanich *et al.*, 2000) and the number of computer and Internet users in Thailand (Bhattarakosol, 2003). With the development of the information revolution in Thailand, Thai OCR has been considered to play an important role. Compared to the Roman alphabet, Thai is considered to be more complex in visual representation. The structure of Thai characters consists of small loops combined with curves, zigzags and there is an absence of spaces between each word and sentence. In each line, moreover, Thai characters can be composed on four levels, depending on the type of character being written. Figure 1.1 shows samples of a Thai phrase in printed and handwritten styles. It has been considered as an interesting and intellectually challenging topic. Compared to Chinese OCR, with its enormous number and complexity of characters (Shi, 2002), Thai OCR seems not to be a difficult task. However, Thai characters pose some unique problems that differ from English and Chinese. Many studies have thus been conducted, proposing several different techniques to solve the problem, which will be discussed in Section 2.4.

The National Electronics and Computer Technology Center (NECTEC), a government national research organisation in Thailand, considered that Thai OCR is one of the

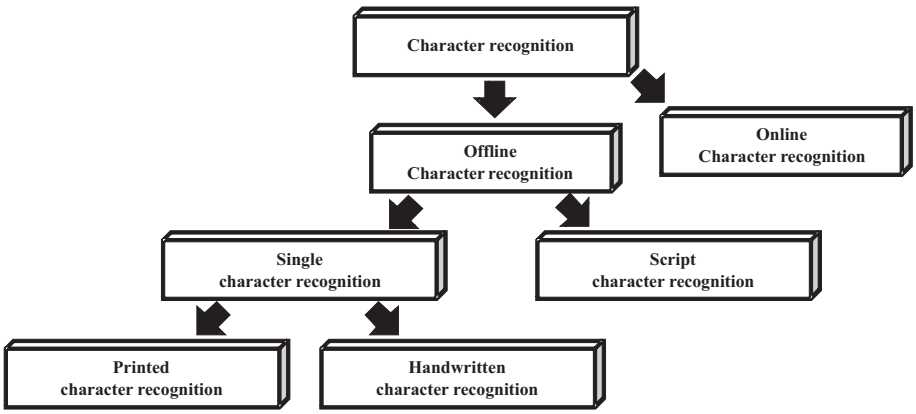


FIGURE 1.2: Different areas of character recognition (Arica and Yarman-Vural, 2001).

most desirable computer applications. Hence, a Thai OCR project was created to deal with the problem, myself being one of the members of the team. We proposed a method for tackling the printed Thai language's OCR problem, using a two-step neural network classification (Tanprasert and Koanantakool, 1996; Tanprasert *et al.*, 1997): Kohonen Self-Organising Maps (SOMs) for rough classification and a set of standard back-propagation neural networks for final classification. More details are provided in Section 2.5.

1.2 Optical Character Recognition

Optical character recognition is a field of research in pattern recognition. It may be defined as a computer method, or computer software, designed to convert document images into an editable text form that a computer can manipulate, for example, ASCII[1] or Unicode[2]. OCR deals with the problem of recognising optically processed characters. Optical recognition is performed offline after writing or printing has been completed, as opposed to online recognition where the computer recognises the characters as they are drawn (Tappert *et al.*, 1990). Figure 1.2 shows these two main categories of character recognition (Arica and Yarman-Vural, 2001). This research focuses entirely on offline Thai character recognition.

1. Offline (paper-based) character recognition refers to the recognition of pre-written or pre-typed text. Thus, the information that can be used is only that observed after the writing has been done. Although the development of electronic media is

[1]American Standard Code for Information Interchange (ASCII) is a code for representing symbols as numbers, with each symbol assigned a number from 0 to 255 (an 8-bit dataspace). For example, the ASCII code for uppercase A is 65. Most computers use ASCII codes to represent text, which makes it possible to transfer data from one computer to another.

[2]Unicode is a code similar to ASCII, used for representing commonly used symbols in a digital form. Unlike ASCII, however, Unicode uses a 16-bit dataspace, and so can support a wide variety of non-Roman alphabets including Chinese, Japanese, Arabic, Thai, and so on.

entering daily life more prolifically, the use of paper-based media is still prominent. An optical scanner converts the image of the writing into a bitmap file (Tappert *et al.*, 1990). Offline character recognition can be divided into two subgroups, single and script character recognition, while single character recognition consists of two types, printed and handwritten character recognition. Single character recognition is the recognition of isolated characters, while script character recognition is the recognition of connected or cursive characters. The rest of the thesis considers both printed and handwritten isolated character recognition.

2. Online (pen-based) character recognition, by contrast, means that the machine recognises the writing as the user writes, so dynamic information can be used to recognise characters. Dynamic information is the temporal information obtained from being able to observe characters as they are written. This information includes the number of strokes taken to write a character, the order of strokes, the direction of the writing for each stroke and the speed of writing within each stroke. Online character recognition requires a transducer that captures the writing as it is written. The most common of these devices is the electronic tablet or digitiser (Tappert *et al.*, 1990).

The main principle in the automatic recognition of patterns is first to teach the machine to classify patterns that may occur and what they look like. In OCR, the patterns are letters, numbers and some special symbols, like commas, question marks, etc., while different classes correspond to different characters. Teaching the machine is performed by showing the machine examples of characters of all the different classes. Based on these examples, the machine builds a prototype or a description of each class of character. Then, during recognition, the unknown characters are compared to the previously obtained descriptions, and assigned the class that gives the best match.

The performance of an OCR system is directly dependent on the quality of the input documents. The more constrained the input is, the better the performance of the OCR system will be (Lorette, 1999) . Furthermore, when it comes to totally unconstrained styles, OCR machines are still a long way from reading as well as humans. However, the advance of computer technology is continually bringing OCR closer to its ideal.

The common components of an OCR system are illustrated in Figure 1.3. The first step in the process is to digitise the document using an optical scanner. Pre-processing is then important to prepare the inputs to be suitable for later recognition stages. Examples of the pre-processing stage include the processes to identify text regions within the documents, eliminate noise, and segment the text into isolated characters. Next, feature extraction is carried out to construct a feature vector from an input image. The goal of feature extraction is to obtain a compact description (a feature vector) that can be used to represent the character uniquely. Recognition is the main decision-making stage in which the extracted features are classified into one of several categories. Finally,

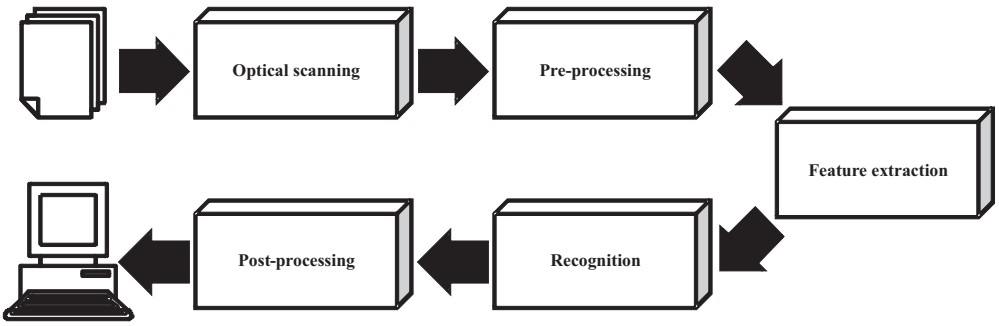


FIGURE 1.3: Components of an OCR system (Arica and Yarman-Vural, 2001).

post-processing typically forms a verification step. Language models and contextual information are used to reconstruct and verify the characters and words of the original text (Arica and Yarman-Vural, 2001).

At present, there are many areas of application in which OCR has been successful and widely used (Mori *et al.*, 1992), for example, for data entry and text entry.

- Data entry, one of the commonly used applications, covers technologies for entering restricted data. Document reading machines are used for banking applications. The systems are characterised by reading only an extremely limited set of printed characters, usually numerals and a few special symbols. They are designed to read data, such as account numbers, customer identification and amount of money. The paper formats are constrained, with a limited number of fixed lines to read per document.

- Text entry is mainly used in office automation. Reading machines are used to enter large amounts of text, often in a word processing environment. Compared with data entry, the set of printed characters read by these machines is rather large.

- Additionally, many other areas of OCR applications exist, such as automatic mail systems and information retrieval in libraries. Some of these are mentioned. For example, automatic vehicle license plate recognition is a system for the automatic reading of car number plates. In contrast to other applications of OCR, the input image is not a file digitised from a scanner, but must be captured by a camera. This creates its own special problems and difficulties, although the character set is limited and the syntax restricted. OCR, moreover, can merge with other research to create sophisticated applications. A reading machine for the blind is another interesting OCR application. Combined with a speech synthesis system, it can help a blind reader to understand printed documents.

1.3 Contributions and Original Work

The main contribution of this thesis is the development of Thai optical character recognition methods (both printed and offline handwritten characters), based on very simple, but effective algorithms. These can briefly be summarised as follows:

- For machine printed Thai character recognition:
  1. The NECTEC Thai OCR system was reviewed, and it was found that the previous training and evaluation seemed to use inappropriate methods. To continue further work, evaluation with the appropriate method, n-fold cross-validation, was performed. An evaluation of the NECTEC Thai OCR that focuses on aspect ratio, the ratio between the width and the height of a character, and SOMs is performed. The results indicate that the classifiers, with and without the aspect ratio, and with and without SOMs, yield small, but statistically significant, differences in recognition rates (approximately by 0.6 and 1.6 percentage points, respectively), as presented in Section 3.4.
  2. A very straightforward method based on the nearest neighbour algorithm with several distance metrics was examined. The results showed a significant improvement in recognition rates over the complicated NECTEC method (by approximately 2.1 percentage points), as demonstrated in Section 3.5.3.
  3. The robustness of the classifiers was investigated. The results showed that the classifier based on the nearest neighbour algorithm proved more robust than the NECTEC classifier, as described in Section 3.5.5.
- For offline handwritten Thai character recognition:
  1. Evaluations on three different classifiers were performed: the nearest neighbour algorithm, with exclusive-OR distance metric (Hamming distance) showed a significant improvement in recognition rates of printed Thai character recognition; the standard back-propagation neural network used in NECTEC Thai OCR; the hidden Markov models applied successfully in offline handwritten Thai character recognition (Nopsuwanchai, 2005). However, HMMs, with 70 states, using the simplest feature, pixel distribution, achieved the best accuracy (91.1% on ThaiCAM dataset, as shown in Section 4.5).
  2. Evaluations on three different databases were performed. Details of the three different databases can be found in Section 4.2. Classifiers yielded a good recognition rate for the 'within database' (each dataset was divided into disjoint training and testing sets of approximately equal size.), but not for cross databases (trained and tested on different datasets). This indicates that the capability of classifiers to identify correctly samples not encountered during training is not good enough. The experimental results can be found in Section 5.2.

3. The recognition performance among the three different databases was worse than the recognition rates of the 'within database' due to many aspects. One is a difference between online (pseudo) and offline (real) handwritten data. An improvement in classification rates was obtained by adjusting the stroke width of characters. The best achievement improved the recognition rates by 5.6 and 12 percentage points for the back-propagation neural network and HMMs respectively (as shown in Section 5.3).
4. Another improvement in classification rates was obtained by merging the training sets of the three databases together for teaching the classifiers (as shown in Section 5.4). An improvement of 18 and 7.6 percentage points in the classification rate for the back-propagation neural network and HMMs was obtained over single database training, respectively.
5. An attempt to improve overall recognition performance among three different databases by applying a boosting algorithm called AdaBoost was made. This boosting has been successfully applied in many research areas in the past, such as face detection (face or non-face) and character recognition. The evaluations in Section 6.3 show that only a slight improvement (sometimes a small decrease) in recognition rates was achieved, while other researchers obtained an apparently large increase in recognition performance using the AdaBoost algorithm. Usually AdaBoost applies to a weak rather than a strong classifier. For this reason, it seems to be unsuccessful in improving recognition accuracy when applied to a well-designed classifier. An investigation into AdaBoost performance is made in Chapter 6.

1.4 Outline of Thesis

The thesis is put together in seven chapters. Chapter 1 has provided an introduction to the research objectives and background to character recognition, together with applications. Chapter 2 gives a survey of character recognition in English, Chinese, Arabic and Thai. The advantages and disadvantages of the work are discussed. A background to the Thai language is provided, in order to understand Thai character recognition. Descriptions of the machine printed OCR developed by NECTEC are also reviewed. Chapter 3 presents the NECTEC printed character image database, followed by the evaluation of NECTEC OCR. A comparison of NECTEC and a very simple classifier, based on the nearest neighbour algorithm, is shown and the results are discussed. The nearest neighbour algorithm with various distance metrics is used to perform classification. The performance and robustness of the classifiers are also examined. Chapter 4 describes and examines offline handwritten Thai character recognition on three different datasets. Results and discussion are also provided. Chapter 5 presents the improvements in the recognition rates of handwritten

Thai OCR. Differences between the online and offline handwritten data are addressed and investigated to improve overall recognition performance. In Chapter 6, an evaluation of the AdaBoost algorithm on our handwritten databases, a machine learning algorithm for improving the accuracy of a classifier, are reported and compared with the original classifiers. Then, the performance of the AdaBoost algorithm applied to our classifiers on synthesised databases is presented and discussed. The conclusions of this thesis are given in Chapter 7, followed by future research perspectives.

Chapter 2

Background and Literature Review

This chapter introduces essential background to the research. First, the basic concepts of the Thai language are presented, followed by its sound system and writing style. Then, a survey of OCR, as applied to English, Chinese and Arabic is described. Next, a review of Thai OCR is presented (Section 2.4). Finally, the NECTEC OCR system is presented in detail (Section 2.5), as it forms the basis for the work carried out to date.

2.1 Introduction

OCR is one of the most successful and oldest applications in the field of automatic pattern recognition. Since the mid 1950's, OCR has been a very active field for research and development (Trier *et al.*, 1996). In the early days of pattern recognition research, many researchers took up the subject of OCR. One reason was that characters were very handy to deal with and were regarded as a problem that could be solved easily. However, contrary to the expectations of many, after some initial progress, great difficulties in solving the problems surfaced (Mori *et al.*, 1992).

A review of OCR research is presented. Offline, rather than online, character recognition for English, Chinese and Arabic OCR is selected as the focus of attention for this work. Because of the vast number of papers published on OCR since the mid 1950s, it is impossible to include all the available methods in this section. Instead, I have tried to make a representative selection to illustrate the different principles that can be used. For Thai OCR, previous research, including machine printed, offline and online handwritten character recognition, is detailed.

To be able to solve the Thai OCR problem efficiently, some knowledge of the Thai language is helpful. Section 2.2 provides an overview of the Thai language and script, as

a background for understanding Thai character recognition, including its sound system and writing system.

2.2 Basic Concepts of the Thai Language

Thai serves as the official national language and the mother tongue of the Thai people. It is the language used in schools, the media and government affairs. More than 90% of its population speak and write Thai as their native language. It consists of 44 consonants, 32 vowel sounds and five phonemic tones that combine to form syllabic sounds. Furthermore, 10 Thai numerals and some extra symbols are used in documents. Their details will be described in the next section.

Unlike a non-tonal language such as English, tone and stress are used to change the semantics of a sentence. In Thai, as with Chinese and other Asian languages, the same word can have a completely different meaning, depending on its prosody.

On the other hand, Thai, like English and other European languages, has an alphabetic script. All syllables must contain a vowel sound, but may begin or end with a consonant. Each syllable consists of one or more consonants, a simple or compound vowel and a tone marker. One or more syllables are combined to form a word. It is written from left to right.

The grammar of Thai is considerably simpler than the grammar of most Western languages. Most significantly, words are not modified or conjugated for tense, plural, gender, or subject-verb agreement. Articles, such as 'a', 'an' or 'the' are not used either. The fundamental structure of a Thai sentence is Subject + Verb + Object with adjectives following nouns.

Thai writing does not use spaces between the words in a sentence. It has no fixed rules about how to space, but generally space (or an equivalent character) is used to indicate the end of a phrase, clause or sentence.

2.2.1 Thai Sound System

The Thai alphabet is syllabic and includes consonants and vowels that are grouped separately. The consonants nominally consist of 44 letters (graphemes), but these represent only 21 consonant sounds (phonemes). Two letters are practically obsolete: ฃ (/k^h/) and ฅ (/k^h/). However, they still appear on many keyboards and in character sets. The characters are divided into three classes: nine middle class consonants (M), 11 high class consonants (H), and 24 lower class consonants (L). The classes are important,

Symbol	IPA Sound	Class	Symbol	IPA Sound	Class
ก	/k/	M	ท	/t^h/	L
ข	/k^h/	H	ธ	/t^h/	L
ฃ	/k^h/	H	น	/n/	L
ค	/k^h/	L	บ	/b/	M
ฅ	/k^h/	L	ป	/p/	M
ฆ	/k^h/	L	ผ	/p^h/	H
ง	/ŋ/	L	ฝ	/f/	H
จ	/c/	M	พ	/p^h/	L
ฉ	/c^h/	H	ฟ	/f/	L
ช	/c^h/	L	ภ	/p^h/	L
ซ	/s/	L	ม	/m/	L
ฌ	/k^h/	L	ย	/y/	L
ญ	/y/	L	ร	/r/	L
ฎ	/d/	M	ล	/l/	L
ฏ	/t/	M	ว	/w/	L
ฐ	/t^h/	H	ศ	/s/	H
ฑ	/t^h/	L	ษ	/s/	H
ฒ	/t^h/	L	ส	/s/	H
ณ	/n/	L	ห	/h/	H
ด	/d/	M	ฬ	/l/	L
ต	/t/	M	อ	/ʔ/	M
ถ	/t^h/	H	ฮ	/h/	L

TABLE 2.1: Thai consonants, IPA sounds and classes.

as they determine the tone in which a syllable should be spoken. Table 2.1 illustrates the symbols, IPA[1] sounds and classes of Thai consonants.

The Thai language has 32 different vowel sounds. Table 2.2 shows the symbols and IPA sounds for each of the vowels. They are divided into short and long vowel sounds or classified into simple and compound vowels. Two of the vowels, ฤ (/rɯ/) and ฤๅ (/rɯː/) are obsolete and are no longer used in written Thai. A dash (−) indicates the position of the initial consonant after which the vowel is pronounced.

Note that vowels can stand above, below, left or right of the consonant, or combinations of these places. Figure 2.1 displays the possible positions of a vowel relative to a consonant in Thai script. However, no matter where the vowels are placed, Thai syllables, unlike English, are always spoken in the same order: initial consonant + vowel + final consonant (optional). For example, the pronunciation of เรือ (*boat*) is /r/ + /ɯəː/.

[1]The International Phonetic Alphabet (IPA) is a system of phonetic notation, devised by the International Phonetic Association as a standardised representation of the sounds of all spoken languages.

Short		**Long**	
Symbol	**IPA Sound**	**Symbol**	**IPA Sound**
–ะ, –ั	/a/	–า	/aː/
–ิ	/i/	–ี	/iː/
–ึ	/ɨ/	–ื	/ɨː/
–ุ	/u/	–ู	/uː/
เ–ะ, เ–็	/e/	เ–	/eː/
แ–ะ	/ɛ/	แ–	/ɛː/
โ–ะ	/o/	โ–	/oː/
เ–าะ	/ɔ/	–อ	/ɔː/
เ–อะ	/ə/	เ–อ	/əː/
เ–ียะ	/iə/	เ–ีย	/iəː/
เ–ือะ	/ɨə/	เ–ือ	/ɨəː/
–ัวะ	/uə/	–ัว	/uəː/
ฤ	/rɨ/	ฤๅ	/rɨː/
ฦ	/lɨ/	ฦๅ	/lɨː/
		–ำ	/am/
		ไ–	/ay/
		ใ–	/ay/
		เ–า	/aw/

TABLE 2.2: Thai vowels and IPA sounds ('–' indicates a consonant grapheme).

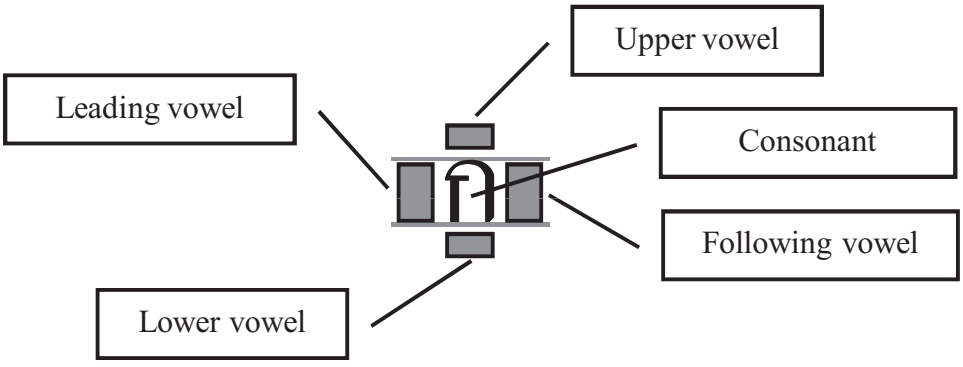


FIGURE 2.1: Possible positions of a vowel associated with a consonant.

Symbol	IPA Sound	Name
none	none	middle tone
◌่	[`]	low tone
◌๊	[ˆ]	falling tone
◌๋	[´]	high tone
◌๎	[ˇ]	rising tone

TABLE 2.3: Thai tone markers, IPA sounds and names.

Symbol	IPA Sound	Numeral
๐	/sǔːn/	0
๑	/nɨ̀ŋ/	1
๒	/sɔ̌ːŋ/	2
๓	/sǎːm/	3
๔	/siˀ/	4
๕	/hâː/	5
๖	/hòk/	6
๗	/cèt/	7
๘	/pɛ̀ːt/	8
๙	/kâːw/	9

TABLE 2.4: Thai numerals, IPA sounds and Western numerals.

Symbol	◌์	ๆ	ฯ	ฯลฯ	฿
Name	silence	repetition	abbreviation	et cetera	Baht

TABLE 2.5: Some special symbols used in Thai.

Since Thai is a tonal language, different tones in each word can produce different meanings. Thai tonal marks represent five different tonal sounds: middle, low, high, rising and falling. Table 2.3 displays the symbols and IPA sounds for each of the tone markers.

Thai numerals, Table 2.4, are traditionally used only for official documents, such as statutes, enactments and the law. For general documents, such as newspapers and business documents, Arabic numerals (also known as Western numerals) are more common.

In addition to the Thai consonants, vowels, tone marks and numerals, there are a number of extra symbols in common use. These are similar in function to certain punctuation symbols that are found in English. Although, in the past, there were several special symbols used in the Thai language, at present some of them are not used in general documents. Table 2.5 shows the special symbols used only in Thai general documents.

FIGURE 2.2: Four levels of a Thai sentence compared with one level of an English sentence.

2.2.2 Thai Writing System

The structure of most Thai letters consists of small loops combined with curves. The writing of a Thai character always starts with a loop followed by curves. The direction of writing in the Thai language is horizontally from left to right, with lines going from the top to the bottom of a page, as in English. The standard Thai writing style is in block letters, and is not cursive, as it is in Arabic and many English writing styles. This means each Thai character is well separated from its neighbours.

However, Thai words are normally written contiguously, without intervening space between words, as opposed to separating each word with a space as in English. Extra space in the Thai writing style indicates only the beginning of a new phrase or sentence.

Figure 2.2 shows the four levels of a Thai sentence compared with one level of an English sentence. Vowels are written above, below, before, or after the consonant they modify, but tone markers stand only above the consonant. Tonal marks cannot be displayed without consonants and cannot be displayed above vowels either. An above or below vowel and a tone marker will be written after the initial consonant.

Generally speaking, the basic structure of sentences consists of Subject + Verb + Object. Nouns are neither singular nor plural. They are not inflected, have no articles and no gender. Nonetheless, some specific nouns are reduplicated or may be used as a prefix to form collective or plural nouns. For an example of reduplication, เด็ก (/dèk/) means 'child', while เด็กเด็ก or เด็กๆ (/dèkdèk/) means 'children'. For an instance of using prefix, ฉัน (/c^{h}ăn/) means 'I', but พวกฉัน (/p^{h}uə̂ːkc^{h}ăn/) is 'we'. Verbs are not inflected and do not change with person, tense, mood or number.

There is no morphological distinction between adjectives and adverbs, and many words can be used in either function. They follow the word they modify, which may be a noun, verb, or another adjective or adverb.

2.3 OCR in Other Languages

This section is arranged to review the OCR for other languages with respect to the state of the art, rather than surveying complete solutions. From the enormous amount of research, the methods for recognising characters can be divided into two main groups. The first group of methods extract feature vectors from training data, compute statistical

distributions, and classify an unknown object by extracting its feature vector and evaluating the most likely character class to which it might belong. This approach is recognises characters holistically as complete characters. The second group of methods decompose a character into low-level features, such as strokes, and recognise it by precisely matching the individual strokes between the input character and a number of template characters (Shi *et al.*, 2003).

English OCR is described in Section 2.3.1. As described in Section 2.2, Thai is an alphabetic script like English, although it has been considered to be more complicated in its visual representation. A technique in English OCR could be useful for solving Thai OCR. Chinese OCR is then reviewed in Section 2.3.2. The sheer number and complexity of the characters to be recognised makes a holistic approach impractical, whereas it can and does work well for English, with its much smaller number of simpler characters. In fact, Chinese characters form an open set, in that it is always possible to invent new ones, whereas the Roman alphabet used by English is closed. A different technique from English will be reviewed, and could be helpful in tackling the Thai OCR problem. Compared to English, the structure of Thai characters is more complicated, but it is less complicated than Chinese. Arabic OCR is described in Section 2.3.3. Arabic text (printed or handwritten) is cursive in general, and characters are normally connected on the baseline (Lorigo and Govindaraju, 2006). Although Thai is normally in block letters, and is not cursive as in Arabic, a touching character can occur when writing is performed at high speed, such as in note-taking. A technique to handle cursive characters will be necessary to improve the recognition performance of Thai OCR.

2.3.1 English OCR

According to Mori *et al.* (1992) and Trier *et al.* (1996), earlier works used template matching methods. Accordingly, a similarity (or dissimilarity) measure between a template and the character image is computed. The template that has the highest similarity (or the lowest dissimilarity) is identified and if this similarity is above (or below) a specified threshold, then the character is assigned to that class. Otherwise, the character remains unclassified. At this time, research focused only on a number of specific typewritten fonts. Such fonts are OCR-A, OCR-B, Pica, Elite, etc. These are characterised by fixed spacing between each character. OCR-A and OCR-B are the American and European standard fonts specially designed for optical character recognition, where each character has a unique shape to avoid ambiguity with other characters similar in shape. Using these character sets, it is quite common for commercial OCR machines to achieve a very high recognition rate at high reading speed (Miletzki, 1997).

Subsequently, in the mid 1960's, OCR research and development opened up from fixed font to multi-font. Multi-font OCR recognises more than one font, as opposed to a fixed

font OCR, which could only recognise symbols in one specific font. However, the fonts recognised by multi-font OCR are usually of the same type as those recognised by fixed font OCR. Multi-font OCR was able to read up to about ten fonts. The limit in the number of fonts was due to the pattern recognition algorithm and template matching, which required that a library of bitmap images from each character for each font was stored. The accuracy is quite good, even on degraded images, as long as the fonts in the library are selected with care.

At the same time, some feature extractions, such as projection of a bitmap character (Mori *et al.*, 1992), zoning (Trier *et al.*, 1996), image transformation (Trier *et al.*, 1996), series expansion (Mori *et al.*, 1992) and statistical moments (Flusser and Suk, 1994; Chim *et al.*, 1999) were applied to represent a character rather than its bitmap image. Additionally, the most usual structural features were incorporated. The features that describe the geometric and topological structure of a character were extracted such as stroke, loop, curl, and start and end points. Then, a matching technique was employed as a recogniser. Since a structure can be broken into parts, it can be described by the features of these parts and by the relationships between them. The problems are then how to choose features and their relationships, so that the description gives each character a clear identification. Usually some pre-processing techniques, such as thinning and contour analysis, are applied before detecting an important structure of the character. Compared to other techniques, structural analysis assigns features with high tolerance to noise and style variations. However, the features are only moderately tolerant to rotation and translation. With this technique, OCR was able to recognise more regular machine printed characters. This led to the opening up of the new challenge of an omni-font OCR and handwritten character recognition. An omni-font OCR can recognise most non-stylised fonts without having to maintain huge databases of specific font information. Usually an omni-font OCR is characterised by the use of feature extraction. The database will contain a description of each symbol class instead of the symbols themselves. This gives flexibility in the automatic recognition of a variety of fonts.

In the mid 1970's, the challenge focused on documents of poor quality, and handprinted and handwritten character recognition. When handprinted characters were considered, the character set was constrained to numerals, a few letters and symbols. The principle underlying template matching is really only appropriate for the recognition of machine printed characters. However, another set of handprinted and handwritten characters need consideration. The variation of shape in handwritten characters is so large that it is difficult to create templates for them. The structural analysis method has now been applied to handwritten character recognition.

In the 1980's, structural analysis approaches were broadly applied in many systems, in addition to statistical methods (Lorette, 1999; Arica and Yarman-Vural, 2001). OCR research focused basically on shape recognition techniques, without using any

semantic information. This led to an upper limit in the recognition rate, which was not sufficient in many practical applications. Hence, in the 1990's, image processing and pattern recognition techniques were efficiently combined with artificial intelligence (AI) methodologies. Researchers developed complex OCR algorithms that receive high resolution input data and require complex methods in the classification phase. In addition to more powerful computers and more accurate electronic equipment, such as scanners, cameras, and electronic tablets, we have the efficient and modern use of methodologies such as artificial neural networks (ANNs), hidden Markov models (HMMs), fuzzy set reasoning and natural language processing. The systems for machine printed OCR are quite satisfactory for restricted applications. However, there is still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained offline handwriting (Arica and Yarman-Vural, 2001; Bunke, 2003).

From the year 2000 until now, much progress has been made in this area. For example, commercial systems for the tasks of handwritten address reading and amount recognition on bank cheques have become available (Bunke, 2003). Nevertheless, there is a clear need to develop the field further, to include applications such as address and cheque reading and work in narrow domains with limited vocabularies, where task specific knowledge and constraints are available. Examples are the relation between zip code and city name in address reading, or courtesy (numerical format) and legal amount (textual format) on a cheque. However, as regards general word or sentence recognition, where no constraints exist, and one is faced with a large, possibly open lexicon, the state of the art is quite limited and recognition rates are rather low. Therefore, the challenge of unconstrained cursive handwritten recognition is important in a number of future applications, for example, the transcription of personal notes, faxes, and letters, or the electronic conversion of historical handwritten archives in the context of the creation of digital libraries (Tomai *et al.*, 2002; Manmatha and Rothfeder, 2005).

2.3.2 Research into Chinese OCR

According to Shi (2002), the first research on Chinese character recognition was in 1966. After that, many studies on Chinese character recognition were published (Suen *et al.*, 2003). A large portion of the research work is divided into two main groups The first group of methods extract feature vectors from training data, compute the statistical distributions and classify unknown objects by extracting their feature vectors and evaluating the most likely character class to which they belong. The second group of methods decompose a character into strokes and recognise a character by precisely matching individual strokes with the input character and a number of template characters. Because Chinese characters represent images that incorporate rich structural information, a combination of the two methods may result in an efficient and practical scheme for pattern recognition.

Liu *et al.* (2001) proposed a model-based stroke matching algorithm. The strokes for each reference character are extracted from the pen trajectories of an online character input system. A database of stroke attributes and inter-stroke relations for each reference character is built up manually. The information includes stroke type, stroke length, orientation, tolerance for variations in length and orientation, and twelve types of inter-stroke relations. Stroke types include horizontal, vertical, dot, slash, hook, etc. Inter-stroke relations include information such as the start point of stroke A is close to the end point of stroke B, the start point of stroke C is near the mid-portion of stroke D, etc. In actual recognition, the unknown input character is pre-processed to give a graph of line segments. An elaborate search algorithm is used to obtain candidate strokes from the input character that correspond to strokes in the reference character. An objective function is defined that depends on the distance between the input stroke and reference stroke, as well as on the compatibility between this pair of strokes and other related stroke pairs. The algorithm tries to search for a solution with minimum cost, such that the constraints imposed by the types of relations are satisfied. Experiments on a database of 783 character classes with 200 examples per class produced an error rate of 1.5% and a rejection rate of 0.6%.

The work of Shi *et al.* (2003) introduced a novel approach, that of decomposing a character into radicals[2] on the basis of image information without first decomposing into strokes. The radical recognition approach based on nonlinear active shape modelling was called active radical modelling. Chamfer distance minimisation was applied to match radicals within a character. Then, the dynamic tunnelling algorithm was used to search for optimal shape parameters to describe the deformation of an active model to fit a test image. Finally, the Viterbi algorithm was employed to combine radicals in a character. The results were tested on a set of 430,800 characters from 2,154 character classes, composed of 200 radical categories collected from 200 different writers. The recognition rate was obtained 96.5% for radicals and 93.5% for characters.

2.3.3 Research into Arabic OCR

Arabic is similar to English, in that it uses letters, numerals and punctuation marks, as well as spaces and special symbols. However, it is written from right to left, unlike Roman text. Arabic letters are normally connected on the base line. Unlike most other languages, such as Chinese, both printed and handwritten Arabic characters are cursive. Furthermore, Arabic characters can take more shapes than Roman characters. Arabic has 28 letters, each of which can be linked in three different ways, or separated, depending on the case. Therefore, each character has up to four different forms depending on its position. Most letters can be connected from both sides: right and left. Some of them have one, two or three dots, and dots can be above, below or in the middle of the letter (Amin, 1997).

[2] A minimal meaningful unit of the Chinese writing system.

The first Arabic OCR was a survey of 1981 newspaper headlines presented by Lorigo and Govindaraju (2006), including the local features (concavities, loops and connectivity) used in language recognition. After the first Arabic OCR had been introduced, many new approaches were proposed. Many features have been applied to Arabic OCR, for example, pixel densities, Fourier descriptors and structural components, such as loops, dots and curves. In the recognition stage, many classifiers are used, for instance, rule-based, ANNs, HMMs and combinations of these. There are two approaches applied to Arabic character recognition: the analytical and global (Amin, 1997). In the analytical approach, a word is segmented into individual characters or sub-characters. A character is then fed into a classifier, and a combination of results produces a ranked list of possible words. In the global approach, a word or sub-word is recognised without segmentation. An advantage of this approach is that it avoids the difficult segmentation stage.

Mozaffari *et al.* (2005) proposed a method for the recognition of Arabic numeric characters, which is structural and also uses statistical features. Endpoints and intersection points were detected on a skeleton, which was then used to separate them into primitives. Eight statistical features were computed for each primitive, the features of which were concatenated, and the results normalised for length. Nearest-neighbour was used for classification. Eight digits were tested; 280 images of each were used for training and 200 for testing. The digits were collected from over 200 different writers. The recognition rate was 94.44%.

Many papers have applied HMM to Arabic OCR since 2000, perhaps due to the power of the frame-based HMM strategy in which the features computed on vertical strips of the image are fed into an HMM. It seems that a trend in Arabic OCR research is towards word-based recognition rather than character-based recognition.

Pechwitz and Margner (2003) used 160 semi-continuous HMMs representing characters or shapes. Thinning was applied to each word. Features used columns of pixels in blurred, thinned images. The models were combined into a word model for each of 946 valid city names. The system obtained an 89 percent word-level recognition rate using the IFN/ENIT database. This database was developed to advance research and development of Arabic handwritten word recognition systems. It consisted of 26,459 handwritten Arabic names of 937 towns/villages by 411 different Tunisian writers.

Khorsheed (2003) applied HMM with skeleton images to recognise text in ancient manuscript. The method did not require segmentation into characters. Training was given in single HMM with structural features. HMM is composed of multiple character models, where each model represents one letter of the alphabet. The recognition rate was 87% and 72% with and without spell-checking. The rate for correct results being in the top 5 choices was 97% and 81%, respectively. The test set was 405 character samples in a single font, extracted from a single manuscript.

In 2005, the first international Arabic handwriting recognition competition was announced in the eighth International Conference on Document Analysis and Recognition (ICDAR, 2005). Five groups submitted systems trained on the IFN/ENIT database. Several feature extractions and classifications were used by five competitors, for example, ANNs and HMMs. The best system using HMM obtained 75.93%, 87.99% and 90.88% in recognition rate for correct answers in the top 1, 5 and 10 choices, respectively (Margner *et al.*, 2005; Lorigo and Govindaraju, 2006).

2.4 Thai OCR

The development of Thai optical character recognition is important in Thailand at the moment. Many studies have been conducted on this topic, proposing several different techniques to solve the problem. However, development has progressed slowly, as can be seen from the small number of published reports and papers over the past 20 years. There are two commercial products for Thai OCR: ARNThai software, by the National Electronics and Computer Technology Center (NECTEC), which first appeared on the market in 1996, and Atrium ThaiOCR by Atrium Technology, launched on the market in 1997. Both focus on recognising a limited set of fonts in printed documents. Section 2.4.1 details previous research on machine printed Thai OCR, followed by a review of offline and online handwritten Thai OCR in Section 2.4.2 and 2.4.3.

2.4.1 Printed Thai Character Recognition

Research on Thai character recognition started relatively late. The first published paper was in 1982 (Hiranvanichakorn *et al.*, 1982), but work on the Roman alphabet did not begin until the 1950s (Mori *et al.*, 1992), and on Chinese and Japanese until the 1960s (Suen *et al.*, 2003). Most of the earlier work used template-based recognition systems (Kimpan and Walairacht, 1993; Hiranvanichakorn and Boonsuwan, 1993), while some applied an artificial neural network (Tanprasert and Koanantakool, 1996; Tanprasert *et al.*, 1996). By 2000, most of the work was in the area of machine-printed Thai character recognition, while only a small number of reports addressed the problem of handwritten Thai character recognition. This review is categorised into two fields: pre-processing techniques and feature extraction and classification techniques.

Pre-processing Techniques

The raw data of a scanned text image cannot usually be processed directly. Pre-processing techniques are required in order to convert the image into a suitable format for the recognition process. The result of the pre-processing stage should be an image containing the words or characters to be recognised without any disturbing

elements. Examples of pre-processing techniques include binarisation or thresholding, noise reduction or smoothing, document skew detection and correction, document decomposition and slant normalisation. In addition, the processes of thinning, skeletonisation and segmentation are sometimes embodied in the pre-processing stage. A number of research reports introduced novel methods to improve the performance of pre-processing techniques in Thai character recognition. A review of these techniques is given below.

Often, the document to be recognised is not placed correctly on the scanner, for example, it may be skewed on the scanner bed, which results in a skewed image. The skewed document has a disadvantageous effect on character segmentation and recognition. Automatic skew angle detection and correction processes should be done prior to the recognition process, in order to attain optimal performance. Duangphasuk and Premchaiswadi (1999) reported a method to detect and correct skew angle in documents by using a linear regression algorithm. This method is claimed to be faster than the conventional Hough transform algorithm (Parker, 1996), while maintaining comparable accuracy. On the other hand, document skew correction can also be done manually, as described in Santinanalert (1999).

The thresholding process, which is required to convert scanned images into bi-level images, may unintentionally make errors and subsequently result in broken parts in the characters. A mending algorithm for broken printed Thai characters was presented by Limmaneewichid and Premchaiswadi (1999). The information of the overlapping areas of the broken image, the specific features of the character and its projection profile are employed in this repairing process. Understanding of Thai writing characteristics is required to solve this problem efficiently. Premchaiswadi *et al.* (2003) presented a scheme for this consisting of two techniques used to identify broken characters: an overlapping area and character code. The specific characteristics of Thai graphemes are also employed in this scheme, such as the position of the head, leg and endpoint of characters. These authors claim that broken characters, both in normal and bold font, are identified and repaired efficiently.

Other pre-processing techniques, for example Thai document layout analysis, have been studied. An approach to eliminate that characteristic by removing non-middle-level characters from the image, based on heuristic rules derived from Thai language properties, was presented by Yingsaeree and Kawtrakul (2005). Non-middle-level characters are usually smaller than middle-level characters and the space between each level is smaller than the space between two consecutive lines of text. After applying this approach, any existing document layout analysis methods can be used with Thai documents without any modification.

Method	Accuracy	Data constraint
Hiranvanichakorn *et al.* (1984)	99.4%	- Thai
- Structural analysis		- 1 font
- Template matching		
Tanprasert *et al.* (1997)	97.1%	- Thai and English
- 8×8 feature matrix		- 23 fonts
- SOMs, Back-propagation NN		
Thammano and Ruxpakawong (2002)	97.8%	- Thai
- Global and local features		- 7 fonts
- Fuzzy, Back-propagation NN		- only Thai characters
Thammano and Duangphasuk (2005)	83.8%	- Thai
- Directional code		- 12 no-head fonts
- Hierarchical cross-correlation ARTMAP		- only Thai consonants

TABLE 2.6: Summary of Thai character recognition research.

Feature Extraction and Classification Techniques

Table 2.6 summarises Thai character recognition research. None of the published work shares the same dataset or samples used for evaluation. Hence, the results are not practically comparable. However, details of research will be described in the following paragraphs.

At the beginning of Thai character recognition research, Hiranvanichakorn *et al.* (1982) used structural analysis of the character contour. The digital contours of the characters are encoded according to their directional differences. Arithmetic operations are then applied to extract the concavities and convexities of the contours. In their recognition process, several geometric features of concave and convex arcs are used to calculate similarities in the arcs. Recognition of an unknown character is made by detecting similar arc pairs. A modification of this method was reported by Hiranvanichakorn *et al.* (1984), who proposed a more effective method to extract concavities, convexities and features for the recognition of low-resolution characters. A Freeman chain code and the directional differences of contour tracing of the characters are utilised to extract concavities and convexities. However, detecting similar arc pairs by this method is rather complex. It is difficult to obtain an optimum result in recognition, and it also takes long computational time for matching. A recognition rate of 99.4% was obtained on the training data by this method on 69 Thai characters (345 tokens).

Another attempt was based on an artificial neural network. Tanprasert and Koanantakool (1996) applied a back-propagation neural network to solve the Thai character recognition problem. A normalisation technique was used to change each isolated character into a standard image size, because the back-propagation neural network must receive the input image in a fixed size. Many different sizes of input matrix

were considered. The experimental results showed that an 8×8 feature matrix gave a better result than other cases. The recognition rate on a real document of training fonts (only Thai characters) is about 90%. An improvement to their method was presented in Tanprasert *et al.* (1997). They applied Kohonen self-organising maps (SOMs) for rough classification before feeding into a back-propagation neural network. An improved recognition rate was obtained. This approach can improve the recognition rate of the one-step neural network classification from 90% to 97%. All of their work used the character image database described in Section 3.2. More details of this research are presented in Section 2.5.

Some studies combine several techniques to recognise printed Thai characters. Premchaiswadi (2001) uses a hybrid character recognition scheme. It consists of the feature matching method and an adaptive resonant theory (ART) neural network. The recognition system uses information obtained from pre-processing for classification of characters into a number of smaller groups. Each group has a limited number of members. Some characteristics of those characters in each group are known in advance. The feature extraction and matching method is used for the recognition of characters that could clearly be discriminated by their features. Ambiguous characters are recognised by using the ART neural network.

Thammano and Ruxpakawong (2002) present an approach to classifying printed Thai characters, using a hybrid of global and local features, the fuzzy membership function and a neural network. The global feature classifies all characters into seven main groups. The local features and fuzzy membership function, combined with some rules, are applied to identify a character. Characters that cannot be identified by these techniques are fed into the back-propagation neural network for recognition. Their experiments were tested only with characters created by the authors. The dataset consisted of seven fonts, different font sizes, styles and scanning resolutions. Experimental results yielded an average recognition rate of 97.8% for their dataset. However, the details of the experimental set-up, such as number of character tokens in training and testing, and evaluation methods were not described clearly.

Thammano and Duangphasuk (2005) use the hierarchical cross-correlation ARTMAP to recognise no-head Thai characters. Their method is evaluated against the fuzzy ARTMAP neural network and two of the Thai OCR software products (ARNThai and Atrium ThaiOCR), available on the market. Experimental results showed that the hierarchical cross-correlation ARTMAP can achieve much higher performance on experimental data. The approximate recognition rates (calculated from eight experiments) achieved 83.8% on twelve Thai no-head fonts. This method has been applied to Thai vehicle license plate recognition as well (Duangphasuk and Thammano, 2006).

Other work on Thai character recognition, such as script identification is reported. Support vector machines (SVMs) have been applied in identifying printed Thai and

English scripts (Chanda *et al.*, 2007). They identify the script of the individual character group (word), combining the different character features obtained from structural shape, profile, component overlapping information, topological properties, the water reservoir concept, etc. However, the authors mentioned that the proposed technique does not work properly on degraded and broken documents.

2.4.2 Offline Handwritten Thai Character Recognition

The first attempts at handwritten Thai character recognition were in the mid 1980's, by Hiranvanichakorn *et al.* (1985b). They introduced a recognition method for handprinted characters based on local features, such as the concavity and convexity of contours. In contrast, Airphaiboon and Kondo (1996) used the head[3] to recognise handwritten Thai characters. The properties include the number and locations of end points and the loop structure of the processed character image. The rough classification stage is used to divide all characters into several subgroups. In the recognition process, a decision tree is manually designed to classify the characters in each category. Additional features based on small details in the character are also applied to the decision process for some character categories.

The work by Phokharatkul and Kimpan (1998) uses a cavity feature, an area of points bounded by stroke on at least three sides, and a neural network for recognising handprinted Thai characters. The recognition process is implemented using mathematical morphology to detect the cavity features of patterns, and a neural network classifier. They use six cavity feature types: east, west, north, south, centre and hole. The cavities are named according to the direction in which they open, that is, the side on which they are not bounded. In the classification stage, character images are divided by position into 3 subgroups: lower (2 classes), middle (80 classes) and upper (16 classes). Each subgroup is recognised separately by its classifiers. Databases for evaluation were collected from 40 different persons. The proposed method with this database achieved an actual recognition rate of 98.3%. Subsequent research was introduced in the following two years (Phokharatkul and Kimpan, 2002). This paper aimed to solve the rotated and scaling character recognition problem by use of Fourier descriptors and genetic neural networks. Fourier descriptors, which describe the closed contour of the character, are applied as features in training neural networks. The contours of the character image are extracted and separated between the outer contours and inner (loop) contours. These are used at the rough classification stage, and the outer contours are used at the fine classification stage. A combination of genetic algorithms and a back-propagation learning algorithm are then used to compute the optimal weights of the neural networks. They claimed that a recognition rate using this method achieves 99.1% for 1,200 examples of handwritten Thai words (a total of 13,500 characters) written by 60 persons.

[3]loop, a special characteristic of Thai characters.

The common disadvantage of the approaches (Hiranvanichakorn *et al.*, 1985b; Airphaiboon and Kondo, 1996; Phokharatkul and Kimpan, 1998, 2002) is that they are not applicable to an unconstrained writing style, where the heads in the characters are often not closed or are written as blobs. Moreover, due to the various writing styles, some characters may have an internal touching stroke that leads to misclassification by number of character heads.

Some attempts were made to solve this problem. Therramunkong *et al.* (2002) proposed an alternative method to extract features of character images using multi-directional island-based projection. A character image is normalised to a fixed size window. In their work, the window size is 36×36 pixels. The normalised image is scanned in four directions: vertical, horizontal, and two diagonal directions from top-left to bottom-right and top-right to bottom-left. Two statistical recognition approaches, the interpolated n-gram model and the HMMs, are employed in the classification process. The performance of the method was investigated using nearly 23,400 handprinted and handwritten characters, collected from 25 persons. They claimed that, in situations where local features are hard to detect, both n-gram and HMM approaches achieved up to 96-99% accuracy for closed tests and 84-90% for open tests. In the closed test, they use all the data for training and the same data for testing. On the other hand, in the open test, they performed an approximation to 3-fold cross validation, using 70% of the data for training and 30% for testing.

Other attempts were also based on HMMs (Nopsuwanchai and Povey, 2003; Nopsuwanchai *et al.*, 2006). They proposed a discriminative training method, maximum mutual information (MMI), to improve the performance of an HMM-based offline Thai handwriting recognition system. The feature extraction is based on their proposed block-based principal component analysis (PCA) and composite images, which are the concatenation of raw images, rotated and a polar transformed version of them. Similar to Therramunkong *et al.* (2002), the raw images to be processed into features are size-normalised to give images of a constant width and height (64×64 pixels in this case). Then the block-based PCA, which consists of applying PCA to a concatenation of small overlapping sections of vertical frames within an image, is employed. The database, collected from 20 native writers who were instructed to write characters in a specially prepared form, was applied for evaluation. Writers were instructed to write in an unconstrained style. The database was divided into disjoint training and testing sets of approximately equal size. They achieved a best result of 95.98% on this database. They also evaluate another database collected by the Thai government's science research organisation (NECTEC). This database for online Thai handwriting recognition research was collected from 63 native writers. Although it is a database for online Thai handwriting recognition research, the authors claim that they use only the static images of handwritten characters without temporal information. The result from the NECTEC database was 95.13%.

The methods introduced by Therramunkong *et al.* (2002), Nopsuwanchai and Povey (2003) and Nopsuwanchai *et al.* (2006) show that these features can handle an unconstrained writing style better than the previous works (Hiranvanichakorn *et al.*, 1985b; Airphaiboon and Kondo, 1996; Phokharatkul and Kimpan, 1998, 2002), which use a set of local features including loops, concavity, endpoints and lines to recognise character image. The HMMs need more data for training and many parameters have to be adjusted to optimise the model. Additionally, the HMMs are deemed most appropriate to cursive handwriting recognition because of their ability to model continuous signals. Nevertheless, offline handwriting is represented as a static image that does not convey the time-varying signals suitable for HMM modelling. To overcome this problem, transformation techniques that can convert two dimensional data into one dimensional sequential data are applied. For example, the sliding-window technique is commonly used to generate an observation sequence from the handwritten image.

The work by Methasate *et al.* (2005) combines both a global feature that represents character shape and local features that represent symbolic structure, to solve the similarity and variety problems of Thai characters. A pixel distribution feature is applied, as a global feature, to classify an image into 20 groups of similar characters. Confusion matrix analysis is used as a rough classification to find the cluster of character models based on their shape. Modified structural feature extraction is used to extract character structures. The structural features consist of loop, end point, junction point and curl. Applied to unconstrained written characters, the robustness of structural feature extraction improved. Several techniques were employed to detect three styles of loop: complete-, incomplete- and filled-loop. All features, global and local, are then used to classify the character within each group. A back-propagation neural network was applied at this step. The proposed method is evaluated on the NECTEC offline handwritten database. The training set contained 300 samples per class and 100 samples per class for the testing set. The proposed method yielded the best recognition rate (87.3%).

2.4.3 Online Handwritten Thai Character Recognition

In online handwriting recognition, characters are recognised while they are being written on an appropriate device, such as an electronic tablet or a combination of screen and digitiser that instantaneously displays what the user writes. Dynamic information captured during the writing process includes trajectories, duration and the order of strokes.

There have been only a few attempts at online Thai character recognition since the first report in 1985. Pioneering work in online handprinted Thai recognition is described in Hiranvanichakorn *et al.* (1985a), where the structural analysis of character strokes, such as sequence, type and location of arcs, together with the convexity and concavity of

contours, are extracted and used to classify the characters. Template matching is used in the classification process. An online handwriting recognition system that can recognise Thai, English, numerals and symbol characters is presented in Kortungsap *et al.* (1999) and Madarasmi and Lekhachaiworakul (2000). In that system, a vector sequence that represents the sequence of directions between each successive sample in a character stroke is used as the feature. It is then passed to three sets of neural networks as input. These three neural networks units are responsible for baseline and non-baseline Thai characters and English characters, respectively. Online Thai handwriting recognition has received more interest over the past few years, and many different approaches are employed.

The work of Pornpanomchai *et al.* (2001) determines the distinct features of Thai characters that are combined to construct compound features. These features are only useful for the recognition of handprinted and well-written characters. In Methasate and Sae-Tang (2002), the features extracted from each stroke segment of the online Thai characters are used as inputs to the HMM-based recogniser. Similar features with the use of neural networks classifiers, rather than HMMs, are presented in Sae-Tang and Methasate (2002). Bounnady *et al.* (2008) proposed a method using multiple representations and elastic matching. The multiple representations are constructed by clockwise and counter clockwise curve segments.

2.5 NECTEC Printed OCR System

NECTEC has been developing a Thai OCR system for two decades. The first engine used a multi-layer perceptron network and a back-propagation learning algorithm as a classifier (Tanprasert and Koanantakool, 1996). The basic concept of the back-propagation neural network is described in the following section. According to this publication, the recognition rate achieved on the NECTEC Thai and English character image database was approximately 90%. After that, the engine was continuously improved by several techniques. In 1997, a two-step neural network classification had been applied to improve the recognition rate (Tanprasert *et al.*, 1997). This achieved an accuracy of 97% in recognising fixed fonts on the NECTEC Thai and English character image database. For feature extraction, several techniques were explored (Tanprasert and Koanantakool, 1996; Tanprasert *et al.*, 1996). In the end, 8×8 matrix features and the aspect ratio of characters were chosen because of their simplicity and best performance.

Figure 2.3 illustrates the concept of the NECTEC OCR system. An input document is digitised by a scanner to produce a black and white image. Then the pre-processing, one of the most significant techniques in Thai OCR, is applied to improve image quality, such as noise cleaning, alignment and segmentation (Tanprasert *et al.*, 1996). Then an isolated character image is transformed to a standard image size and fed into the classification engine to recognise the character (Tanprasert and Koanantakool, 1996;

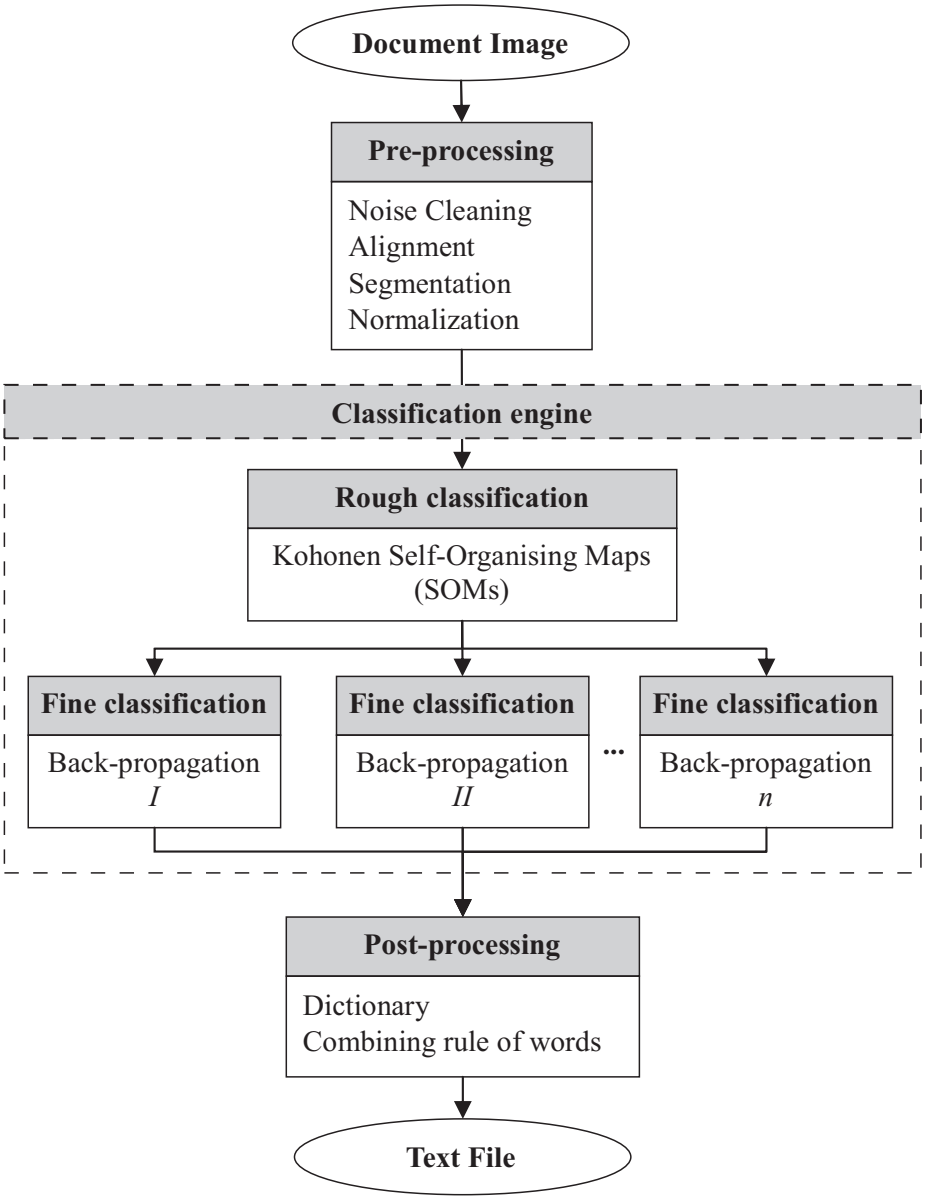


FIGURE 2.3: Diagram of the NECTEC OCR system.

Tanprasert *et al.*, 1997). Finally, text from the recognition engine is corrected by post-processing using a dictionary and grammatical rules for combining words (Meknavin *et al.*, 1998). Details of each process are described in the following sections.

2.5.1 Artificial Neural Network

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of the paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example.

An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process (Bishop, 1995). Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an 'expert' in the category of information it has been given to analyse. This expert can then be used to provide projections, given new situations of interest and answer 'what if' questions.

Neural network models have been proved to be very effective in solving difficult pattern recognition and optimisation problems (Bishop, 1995). It has been shown that the decision rule of a multilayer perceptron is consistent with that of a Bayes classifier. In practise one cannot build a Bayes classifier, since the class probability density functions are often not known. A neural network can work well, since it can learn from training samples so that its decision rule reflects the data distribution. Here, we are using a popular neural network architecture called back-propagation, which will be introduced in the following section.

Standard Back-propagation

The training of a network by back-propagation involves three stages: the feed forward of the input training pattern, the calculation and back propagation of the associated error, and the adjustment of the weights. After training, application of the network involves only the computations of the feed forward phase. Even if training is slow, a trained network can produce its output very rapidly. Numerous variations of back propagation have been developed to improve the speed of the training process. Although a single-layer network is severely limited in the mappings it can learn, a multilayer net (with one or more hidden layers) can learn any continuous mapping to an arbitrary accuracy. More than one hidden layer may be beneficial for some applications, but one hidden layer is sufficient.

A multilayer neural network with one layer of hidden units (Z units) is shown in Figure 2.4. The output units (Y units) and the hidden units may also have biases (as shown). The bias on a typical output unit Y_k is denoted by w_{0k}; the bias on a typical hidden unit Z_j is denoted v_{0j}. These bias terms act like weights on connections from units whose output is always 1. (These units are shown in Figure 2.4, but are not usually explicitly displayed.) Only the direction of information flow for the feedforward phase of the operation is shown. During the back-propagation phase of learning, signals are sent in the reverse direction.

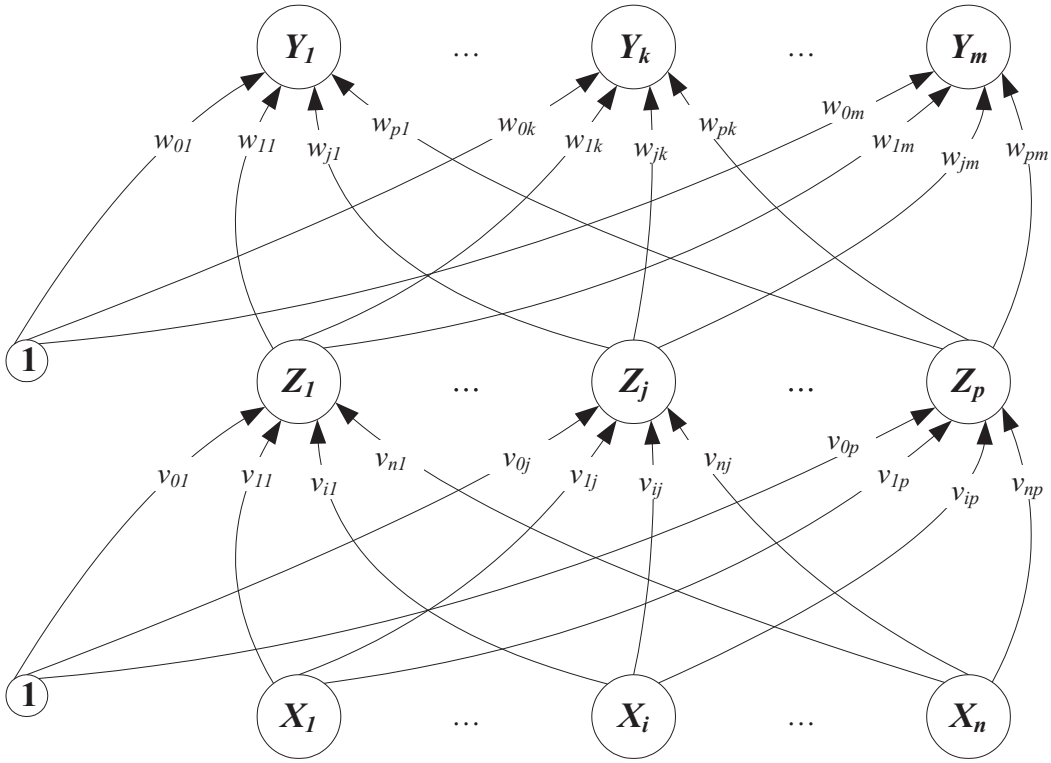


FIGURE 2.4: Back-propagation neural network with one hidden layer.

Activation function

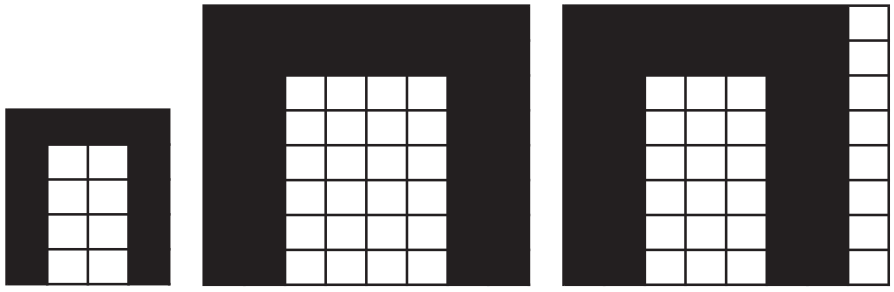
An activation function for a back-propagation net should have several important characteristics. It should be continuous, differentiable, and monotonically non-decreasing. Furthermore, for computational efficiency, it is desirable that its derivative be easy to compute. For the most commonly used activation functions, the value of the derivative (at a particular value of the independent variable) can be expressed in terms of the value of the function (at that value of the independent variable). Usually, the function is expected to saturate, i.e., approach finite maximum and minimum values asymptotically. One of the most typical activation functions is the binary sigmoid function, which has a range of $(0, 1)$ and is defined as:

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.1}$$

This function will be used in the standard back-propagation algorithm for training the network.

2.5.2 Pre-processing

The document image from a scanner is fed into pre-processing. Some pre-processing techniques are used to improve image quality. Noise cleaning is performed to erase some



(a) Original image. (b) Normalisation 1. (c) Normalisation 2.

FIGURE 2.5: An example of an original and two normalised images.

unwanted pixels (salt and pepper noise) from the image. One or two isolated pixels are deleted at this step. The alignment process is applied to adjust the direction of the page, if the input document is not passed upright through the scanner or some mistakes occur just before scanning. Line and character segmentation algorithms are applied to the input image at this period to separate each character. The block-colouring technique searches a group of connected pixels and cuts it into isolated characters. The details of alignment and line and character segmentation algorithms are described in Tanprasert *et al.* (1996). A set of isolated characters is obtained from this step.

Since back-propagation neural networks have to receive input of a fixed size, a normalisation process is required to change each isolated character into a standard image size. Two major problems should be considered at this stage: first, the suitable fixed size of input matrix for the neural networks; second, a method to transform the original character to the normalised one. Usually, there are two methods for mapping the original input to the fixed input matrix, as shown in Figure 2.5. One is to transform the input to be both full width and height in the new size matrix (Figure 2.5(b)). The other is to scale the input to fit the frame, while still preserving the original character aspect ratio (Figure 2.5(c)).

In Tanprasert and Koanantakool (1996), experiments on the two transformation techniques are described. The results show that the recognition rates of these two techniques were close to each other. The first method, therefore, was chosen, because it was easy and fast to implement.

A further experiment by Tanprasert and Koanantakool (1996) used many different sizes of matrix. The results indicate that an 8×8 input matrix was acceptable, because it achieved the same level of correct recognition with the smallest size of input matrix. This reduces time and space complexities in the recognition processes. Figure 2.6 shows the conversion from an original image into an 8×8 matrix feature.

Moreover, it is the view of Tanprasert *et al.* (1997) that the ratios between width and height of every Thai character are different. Some of them have a long tail, while others

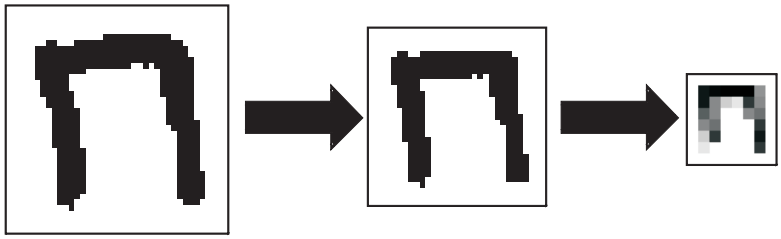


FIGURE 2.6: Example of converting from an original image to a 32×32 binary image and then to an 8×8 matrix feature.

are wide-bodied. Therefore, the aspect ratio (between width and height) may be a useful feature to classify each character. It was chosen as an additional feature for classification.

2.5.3 Classification Engine

The first NECTEC classifier used only the standard back-propagation neural networks (as mentioned in Section 2.5.1) introduced by Tanprasert and Koanantakool (1996), and can recognise only Thai characters. However, most Thai documents usually contain both Thai and English characters. Hence, the OCR engine should be capable of recognising both languages. The total number of characters for the OCR engine is more than 180, which is quite a large number for recognition by artificial neural networks. To reduce this problem, applying SOMs and a back-propagation algorithm to perform a two-step classification of all characters was developed in 1997. Experimental results by Tanprasert *et al.* (1997) show that the new classifier produces a significantly higher recognition rate over the existing classifier.

The technique aims at reducing the problem size for a back-propagation neural network by breaking the recognition process into rough and fine classifications, as shown in Figure 2.3. The performance is expected to be better if we can categorise all characters into groups of similar characters and then use a back-propagation neural network to learn to classify characters in each group.

For rough classification, SOMs, an unsupervised learning algorithm, were employed. It is a popular unsupervised model for categorising input data into small clusters. With this technique, the number of output clusters is specified, but it is quite difficult to determine the optimal value. If there are too few clusters, many characters will be packed into the same group, which will not effectively reduce the workload for the back-propagation neural network. However, with too many clusters, the accuracy of the clustering phase may be reduced, even though higher accuracy for each back-propagation neural network can be expected, due to fewer characters per cluster.

Tanprasert *et al.* (1997) stated that an optimal number of clusters should produce the highest final recognition rate (the maximum product of rough and fine classification accuracy). In determining the number of clusters, they examined the number of distinct

characters per cluster by experimenting on numbers of clusters. Then, the recognition rates of different numbers of clusters from the back-propagation neural network were computed in order to find an optimal solution. The number of clusters producing the best results was eight.

For fine classification, the back-propagation neural network was applied. According to Tanprasert and Koanantakool (1996), this technique has proved to be very promising in recognising the 78 Thai characters. Hence, when the characters are clustered, the complexity of the problem for each back-propagation neural network should be reduced and a higher recognition rate can be expected.

From experimental results by Tanprasert *et al.* (1997), each back-propagation neural network contains 64 input nodes and an aspect ratio feature, the 64 hidden neurons and the number of output nodes being equal to the number of distinct characters in each cluster. The average recognition rate in this technique was 97% on the NECTEC character image database, outperforming the 90% recognition rate of the previous technique on the same database (Tanprasert and Koanantakool, 1996). All details of these experimental results are shown in Tanprasert *et al.* (1997).

Although the results obtained by this engine achieved an acceptable recognition rate, it was evaluated on characters in a few fixed fonts and tested with the entire training data. There are two problems with using this naive approach: the final model will normally fit the training data specifically and the error rate estimate will be overly optimistic (lower than the true error rate). Hence, the NECTEC classifier should be examined with a more acceptable evaluation method. Details of a better evaluation method and its results are described in Chapter 3.

2.5.4 Post-processing

The objective of this step is to detect a series of mistaken characters and correct them. To solve the problem of OCR error correction, the first task is usually to detect error strings in the input sentence. For languages that have explicit word boundaries, such as English, in which each word is separated from the others by spaces, this task is comparatively simple. If the tokenised string is not found in the dictionary, it could be an error string or an unknown word. However, for languages that have no explicit word boundary, such as Thai, this task is much more complicated. First, the approximate boundaries of these dubious areas could be obtained by applying a word segmentation algorithm and finding word sequences with low probability. To generate the candidate correction words, a modified edit distance that reflects the characteristics of Thai OCR errors could be used. Finally, a part-of-speech (POS) trigram model and feature-based model using Winnow's algorithm are combined to determine the most probable corrections (Meknavin *et al.*, 1998).

2.6 Summary and Discussion

This chapter has provided background knowledge on OCR, as well as reviewing different areas and common components of character recognition, followed by some successful applications of OCR. A review of English, Chinese and Arabic OCR research is also provided. Prior to describing Thai OCR and introducing the NECTEC printed OCR system, the chapter provided an overview of the characteristics of Thai script, its sound system and writing style. This background knowledge is essential to a thorough understanding of our model for Thai character recognition. In the next chapter, the evaluation of the NECTEC printed OCR system, particularly the classification module, will be performed. Additionally, some problems with the NECTEC Thai OCR system are discussed.

Chapter 3

Evaluation of NECTEC Printed OCR

This chapter describes the evaluation of the NECTEC printed OCR. First, the database used in the evaluation is described in Section 3.2. An evaluation method is presented in Section 3.3. Then, an investigation into performance of the NECTEC classifier is presented in Section 3.4. Next, a comparison of the NECTEC classifier and a simple classifier based on the nearest neighbour algorithm is described in Section 3.5. Finally robustness of the classifiers is examined.

3.1 Introduction

In the previous chapter, research into OCR was reviewed, including English, Chinese and Arabic. A survey of Thai OCR was also reported. The NECTEC printed OCR system was described. Prior to describing evaluation of NECTEC printed OCR, the character images for training and testing will be shown and explained. Then the methods for evaluating printed OCR and the results will be shown. Finally, some problems with the system are also discussed.

NECTEC printed OCR consists of several components, as shown in Figure 2.3, all of which should be evaluated, though, in this chapter, we only focus on the classification engine. The evaluation is divided into two parts. First of all, the input of including aspect ratio of an original character image, a feature used in the recognition engine, is evaluated. Then, the effect of SOMs is estimated.

A comparison of the NECTEC printed OCR and nearest neighbour, a simple classification method, will be shown and discussed. First, the recognition performance on original data will be examined. Next, the robustness on distorted data will be reported. In the evaluation, two types of distorted data, rotated and noisy, were applied. In the case of

Microsoft	Apple
AngsanaUPC (AS)	AgfaTom (AT)
BrowalliaUPC (BW)	DB75Narai (75)
CordiaUPC (CD)	DB95ThaiText (95)
DilleniaUPC (DL)	DBFongNam (FN)
EucrosiaUPC (EC)	DBNarai (NR)
FreesiaUPC (FS)	DBSurawong (SW)
IrisUPC (IR)	DBThaiText (TT)
JasmineUPC (JM)	EACChuanPim (CP)
SV Jittra (JT)	EACEact (EA)
SV Kanokraykha (KK)	EACPemai (PM)
SV Kantima (KT)	PSL-Text (PS)
TS PeePee (PP)	

Table 3.1: Fonts in NECTEC Thai and English character image corpus.

noisy data, a random noise generator was used to create salt-and-pepper noise, which was added to the image.

3.2 NECTEC Printed Database

As with other pattern recognition applications, such as speech recognition, an efficient Thai optical character recognition system requires large amounts of data for training, validating and testing sets. NECTEC, as a national research centre, strives to help researchers save time preparing data and budgets. A Thai and English character image database was developed and published for academic use[1]. In addition, this corpus can be used as a standard for comparing the performance of recognition engines.

The database contains a set of training, validating and testing material for printed character recognition, including 76 Thai, 85 English and 23 connected characters. It consists of over a million character tokens in two groups of fonts: 12 fonts from Microsoft and 11 fonts from Apple. The font names are shown in Table 3.1. These fonts were chosen because of their popularity. Each character has four styles: normal, italic, bold and italic&bold and eight sizes: 8, 10, 12, 14, 16, 18, 20 and 22 point. Each character is digitised into a black and white bitmap with three scanning resolutions: 200, 300 and 400 DPI[2]. The 200 and 400 DPI characters are scanned twice, while 300 DPI characters are scanned five times because it is the standard for general OCR software (Thongprasirt *et al.*, 2001).

[1] `http://www.nectec.or.th/corpus/`

[2] Dots Per Inch (DPI) is a measure of printing, scanning or display resolution, in particular the number of individual dots or pixels that can be produced within a linear one-inch (2.54 cm) space.

There were four phases in the development of the database. First, all characters were prepared using Microsoft Word processing and printed out on a Hewlett-Packard laser printer (HP LaserJet 4050). Next, the pages of the computer printout were digitised with a Hewlett-Packard desktop scanner (HP ScanJet 5300). Some pre-processing such as noise removing and thresholding were applied to the digitised data. Touching and broken characters were edited manually. Then, the specific software developed by NECTEC extracted the character from the scanned form using a two-pass algorithm (Jain *et al.*, 1995). Finally, the image files were cut into isolated image characters and saved as black and white bitmap files.

3.3 Evaluation Methods and Significance Tests

Previous evaluation had only ever been done by training and testing on the entire data set. All the data were fed into the classifier for training, and the same data were subsequently tested on the same classifier. This is inappropriate evaluation, because it does not test for generalisation performance. The problem with using the entire training set to estimate performance is that it does not give an indication of how well the classification will do when it is asked to make new predictions for data it has not already seen. So the appropriate evaluation, n-fold cross-validation, is selected to verify the original classifier. In the following evaluation, we chose to do a 10-fold cross-validation because there is enough data to make each fold statistically significant.

For the 10-fold cross-validation the original data were partitioned into 10 subgroups, equally and randomly. A single fold is retained as the validation data for testing the model and the remaining folds are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 folds used exactly once as the validation data. The 10 results from the folds can then be averaged (or otherwise combined) to produce an estimate of mean performance and its variability.

After using 10-fold cross-validation, a significance test was performed. This is the method of testing the data to see if the differences between results are significant (Chatfield, 1995). In any such situation the experimenter has to weigh the evidence and, if possible, decide between two rival possibilities. The hypothesis that we seek to test is called the null hypothesis, and is denoted by H_0. Any other hypothesis is called the alternative hypothesis, and is denoted by H_1 (Sheskin, 2004). In our evaluation, a decision on whether to accept or reject the hypothesis was based on whether there was any difference between the classifier with and without the aspect ratio, or with and without SOMs. H_0 is known as the null hypothesis because it assumes there is no difference between the two classifiers. The second possibility in this situation is that the classifier with the aspect ratio, or with SOMs differs significantly from the classifier without them, and this is the alternative hypothesis.

Having decided on the null and alternative hypotheses, the next step is to calculate a test statistic (t_s) which will show up any departure from the null hypothesis. The t_s value will be compared with the t-test table value ($t_{\alpha,\ df}$) in order to reject or accept the null hypothesis (H_0). The level of significance (α) is the probability of a false rejection of the null hypothesis. The significance level at 0.05 (95% probability of making a correct statement) is usually acceptable for statistical work (Chatfield, 1995). If the t_s value exceeds the tabulated value, we say that the means are significantly different at that level of probability. The t-test table (table of student's t distribution) can be found in many statistics books (Chatfield, 1995; Sheskin, 2004).

When comparing two different methods, it often happens that experiments are carried out in pairs. The paired t-test, a statistical significance test that uses the t-distribution, is used. It can compare samples that are subjected to different conditions, provided that the samples in each pair are otherwise identical.

For computing the paired t-test, the difference between the folds is calculated for each pair, and the mean and standard error of these differences are calculated. Dividing the means by the standard error of the mean yields a test statistic (t_s) that is t-distributed with degrees of freedom (df) equal to one less than the number of pairs.

More details of the significance test, t-test table and paired t-test can be found in Chatfield (1995) and Sheskin (2004).

3.4 Experimental Evaluation

This section presents the evaluation results for the NECTEC OCR system. Training and evaluation of the NECTEC printed OCR engine were carried out using the Stuttgart Neural Network Simulator (SNNS) version 4.1 (Zell *et al.*, 1995). The network contains 64 hidden nodes and the number of input nodes depends on the evaluation (64 or 65 nodes). The number of output nodes is equal to the number of characters. In the case of back-propagation neural networks without applying the Kohonen self-organising maps (SOMs), the number of output nodes was 162. After applying the rough classification by SOMs, the training set was separated into 8 subgroups. The number of output nodes in each group was decreased by about 10 to 30% (a decrease of approximately 15 to 50 nodes). The training process was carried out until the Mean Squared Error (MSE) converged to less than 0.01, which corresponded to a correct classification rate of 99%, or when the number of epochs reached 1,000.

For the training and testing data, over 0.6 millions character tokens in 12 fonts from Microsoft, in different font sizes, styles and scanning resolutions on the NECTEC Thai and English character image database were used. The names of the fonts and details of the database are shown in Table 3.1 and Section 3.2 respectively. 10-fold cross-validation

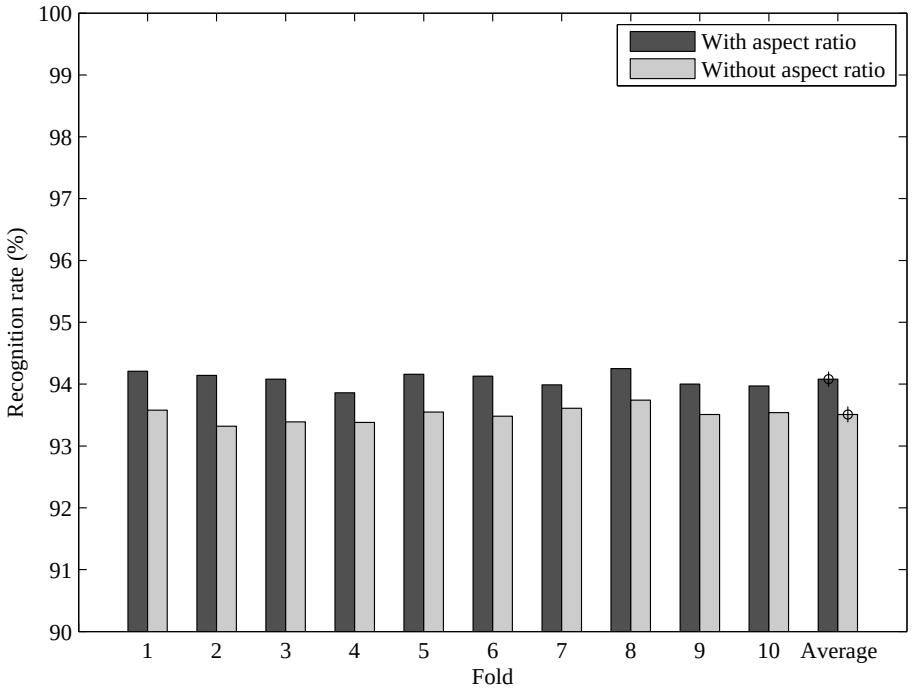


FIGURE 3.1: Recognition rates of each fold of the back-propagation neural networks when trained with the 8×8 matrix and 1 aspect ratio feature, and with only the 8×8 matrix. The last bar is the average recognition performance with the error bar.

was selected to evaluate the method and the paired t-test, a significance test, was applied to calculate the statistical significance.

3.4.1 Aspect Ratio of an Original Character

The objective of this evaluation was to determine whether or not the inclusion of the aspect ratio increased performance.

Figure 3.1 presents the accuracy of the NECTEC printed OCR engine. The x-axis shows the recognition rates of the 10-fold cross-validation in each fold and the last bar is the average recognition performance with the error bar. The y-axis indicates the percentage recognition rate. The dark grey bar represents the engine trained with the 8×8 matrix with the aspect ratio feature, and the light grey bar denotes the system trained with only the 8×8 matrix features. The chart shows that the recognition rates of the classifier with an aspect ratio feature exceed those of the classifier without an aspect ratio feature in all cases. Moreover, the average recognition rate of the classifier with an aspect ratio feature (94.1%) is higher than that of the classifier without an aspect ratio feature (93.5%), although there is only a slight increase.

A paired t-test, as described in Section 3.3, was used to measure the significance of the results. The calculated t ($t_s = 13.40$) is greater than the tabulated t ($t_{0.5,\ 9} = 2.26$).

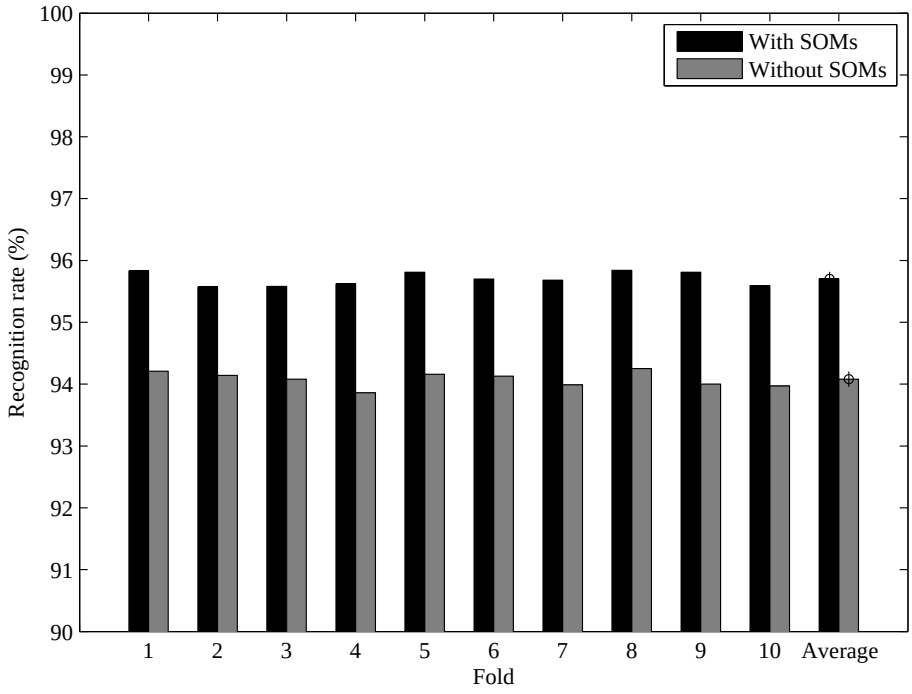


FIGURE 3.2: Recognition rates of each fold of the back-propagation neural networks with and without SOMs when trained with the 8×8 matrix and 1 aspect ratio feature. The last bar is the average recognition performance with the error bar.

Thus, the result is significant at the 5% level. We have fairly conclusive evidence that, although the difference in recognition rates between the neural network with and without an aspect ratio is quite small, it can be considered to be statistically significant.

3.4.2 Kohonen Self-Organising Maps

The purpose of this evaluation was to quantify the effect that SOMs play in the recognition process. Because of the large number of Thai and English characters, the two-step classifier was intended to reduce the complexity of the problem, and so produce a higher recognition rate (Tanprasert *et al.*, 1997).

The dark grey bar in Figure 3.2 represents the two-step classifying neural network, including SOMs for the rough classification with the back-propagation neural network for fine classification. The light grey bar represents only back-propagation neural networks. The graph indicates that the recognition rates (each fold) of the classifier with SOMs exceed those of the classifier without SOMs by a small margin. The average recognition rate of the two-step classifier (95.7%) is always slightly greater than that of the classifier without SOMs (94.1%), by approximately 1.6 percentage points.

The calculated t ($t_s = 45.38$) is greater than the tabulated t ($t_{0.5,\ 9} = 2.26$). Thus, the result is significant at the 5% level. We have fairly conclusive evidence that the

difference in recognition rates between the two-step classifier and the back-propagation neural network alone is statistically significant.

3.5 Comparison with Nearest Neighbour Classification

As described in Section 2.5 and Figure 2.3, the NECTEC OCR system, particularly the classification engine, is complicated. This leads to the question: "*Do we need all the complexity of the NECTEC classifier?*". To answer this, a very simple classifier was tested and compared with the NECTEC printed OCR system.

3.5.1 Nearest Neighbour Classification as a Baseline

The k-nearest-neighbour (k-NN) method is a fundamental and simple classification that is easy to understand. It has been used as a classifier in many works (Garain and Chaudhuri, 2003; Cha *et al.*, 2005; Lee and Coelho, 2005; Pati and Ramakrishnan, 2007; Zhang *et al.*, 2008). Hence, it was chosen as a baseline classifier and compared with the NECTEC OCR system. The principal concept is to classify an object by the majority vote of its neighbours in some space, e.g. feature and coordinate space. The object is assigned to the class most common among its k nearest neighbours. The value of k is a positive number, and it is helpful to choose k to be an odd number in order to reduce tied votes. If k is equal to 1, then the object is simply assigned to the class of its nearest neighbour.

In Figure 3.3, an example of k-NN is shown. There are two classes: triangles and squares. The circle represents the unknown sample. If k is equal to 1, its nearest neighbour comes from the triangle class. Therefore, the unknown object is labelled as the triangle class. With k value equal 3, two of its nearest neighbours come from the square class, so the unknown object is labelled as the square class.

The neighbours are taken from a set of objects for which the correct classification is known. This can be thought of as the training set for the method, although no explicit training step is required. The training phase consists only of storing the feature vectors and class labels of the training samples. In the actual identification phase, an unknown object is represented as a feature vector. Distances from the unknown object to all training objects are calculated, and the k closest samples are selected. A variety of distance measures may be used.

The Euclidean distance measure is the straight-line distance between two points. The distance between points $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ is defined as:

$$d(A, B) \quad = \quad \sum_{i=1}^{n} \sqrt{(a_i - b_i)^2} \tag{3.1}$$

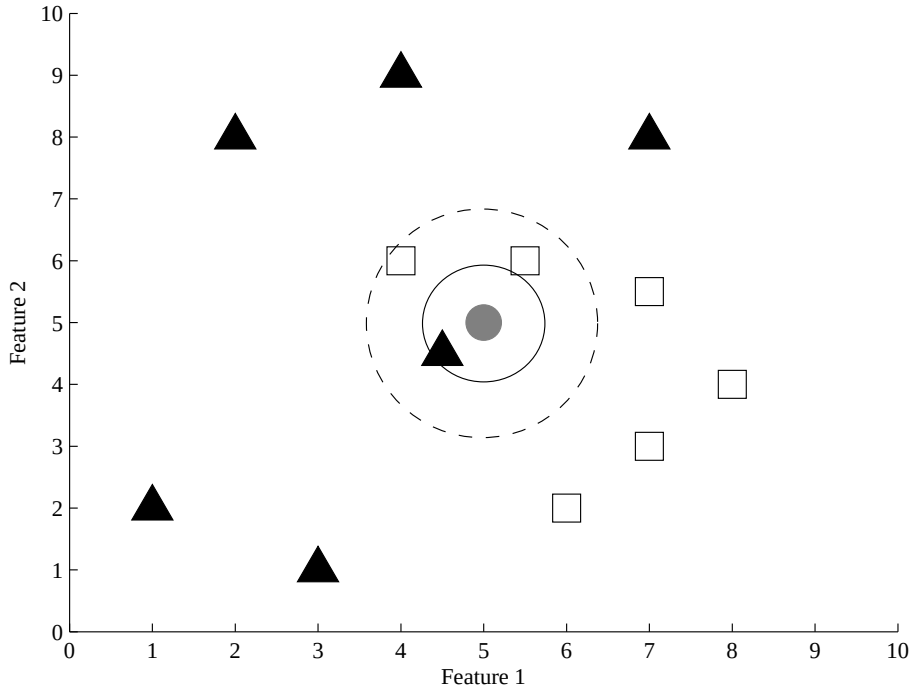


FIGURE 3.3: Principal concept of k-nearest-neighbour (k-NN) classifier.

The city block distance measure (also known as the Manhattan distance) is another commonly used metric. Instead of using a straight-line distance like the Euclidean distance, the path between two points is based on a 4-connected neighbourhood. Points whose edges touch are 1 unit apart, while points diagonally touching are 2 units apart. So the distance between points $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ is defined as:

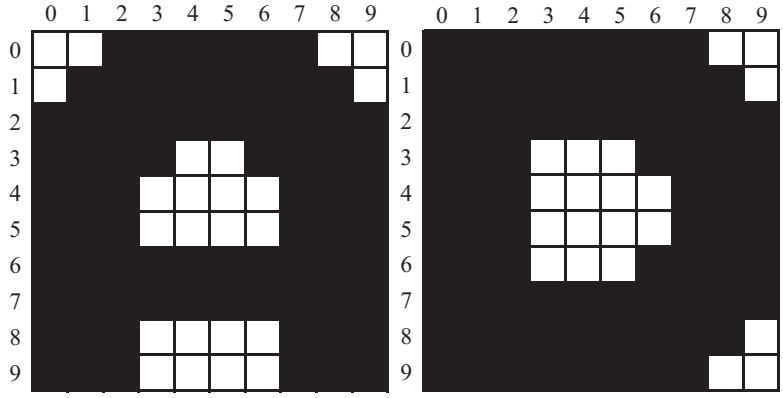
$$d(A, B) = \sum_{i=1}^{n} |a_i - b_i| \tag{3.2}$$

3.5.2 Other Distance Metrics

In addition to the well-known distance metrics described in the previous section, some more complicated distance measures were used. In the next sections, other distance measures, for instance exclusive-OR (XOR), classical, modified and greyscale Hausdorff distances, are described and could in principle be used instead.

Exclusive-OR distance

The exclusive-OR distance measure (XOR) is usually applied to binary data $\{0, 1\}$. For non-binary data, the Hamming distance can be applied instead of the exclusive-OR



(a) Examples of two binary images

	0	1	2	3	4	5	6	7	8	9
0	1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	1	1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	1	1	1	0	0	1
9	0	0	0	1	1	1	1	0	1	1

(b) Result of two binary images using exclusive-OR logical operation.

FIGURE 3.4: Examples and result of exclusive-OR logical operation.

distance. However, sometimes a thresholding technique can be used to create binary data. The distance between sets $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ is found by a logical exclusive-OR operation ($\oplus$). The resulting element is 1, if a_i or b_i , but not both, is non-zero. The distance can be defined as:

$$d(A, B) = \sum_{i=1}^{n} a_i \oplus b_i \tag{3.3}$$

An example of two binary images is presented in Figure 3.4(a) and the result of exclusive-OR distance computation is shown in Figure 3.4(b). So the distance between the two binary images, $d(A, B)$ is 18.

Classical Hausdorff Distance

Similar to the exclusive-OR distance, the Hausdorff distance measure (HD) is used to compute a degree of similarity between two binary images (Huttenlocher *et al.*, 1993; Rucklidge, 1996), but it is a non-linear operator. The concept of Hausdorff distance is to find the maximum distance of a set to the nearest point in another set. Suppose that

there are two sets of points representing a model and an image. The Hausdorff distance between two point sets is small exactly when every point in the model is close to some point in the image, and every point in the image is close to some point in the model.

The Hausdorff distance is actually composed of two asymmetric distances: the forward distance, the distance from the model to the image, and the reverse distance, the distance from the image to the model. The forward distance is small when every point in the model is close to some point in the image, but some points in the image may be far from any point in the model. The reverse distance is small when every point in the image is close to some point in the model. In other words, the forward distance indicates when the model looks like some subset of the image, but not necessarily vice versa, and similarly for the reverse distance. When both are small, then the image and model look like each other, and the Hausdorff distance is small as well. So the distance from set $A = \{a_1, a_2, \ldots, a_m\}$ to set $B = \{b_1, b_2, \ldots, b_n\}$ is defined as:

$$H(A,B) = \max\{h(A,B), h(B,A)\} \tag{3.4}$$

$$\text{where } h(A,B) = \max_{a \in A} d(a,B) \tag{3.5}$$

$$\text{and } h(B,A) = \max_{b \in B} d(b,A) \tag{3.6}$$

The function $h(A,B)$ is called the forward distance (or 'directed')from A to B, while the function $h(B,A)$ is called the reverse distance (or 'undirected'). The distances from point a to set B and from point b to set A are defined as:

$$d(a,B) = \min_{b \in B}\{|a_x - b_x| + |a_y - b_y|\} \text{ where } a_v = 1 \tag{3.7}$$

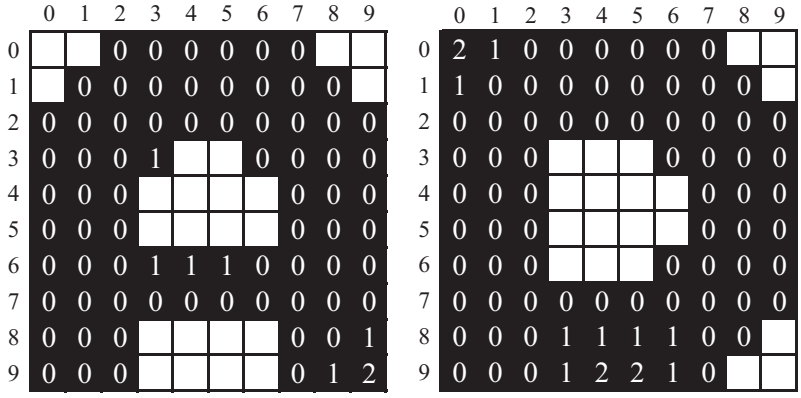
$$d(b,A) = \min_{a \in A}\{|a_x - b_x| + |a_y - b_y|\} \text{ where } b_v = 1 \tag{3.8}$$

where a_x and b_x are x-coordinates, while a_y and b_y are y-coordinates of the points a and b. The binary value of pixels a and b are represented by a_v and b_v respectively. The distance between two points $d(a,b)$ usually uses Euclidean distance. However the city block distance has been used in our evaluation, because it is relatively simple and reduces recognition time without decreasing accuracy.

In Figure 3.5, an example of classical Hausdorff distance is shown. Figure 3.5(a) indicates the 'directed' Hausdorff distance, while the 'undirected' Hausdorff distance is shown in Figure 3.5(b). From the example, the Hausdorff distance of image A and B, $H(A,B)$, calculated by the classical Hausdorff method, is 2.

Modified Hausdorff Distance

The modified Hausdorff distance (MHD) is one of several distance measures based on the Hausdorff distance. Because the classical Hausdorff value is set by the maximum distance among the two point sets, it is sensitive to outliers. The advantages of modified Hausdorff



(a) Distances from image A to B.

(b) Distances from image B to A.

FIGURE 3.5: Distances between image A and B using classical Hausdorff metric.

distance over classical Hausdorff distance are that its value increases monotonically as the amount of difference between the two sets of points increases, and it is robust to outlier points that might result from segmentation errors (Dubuisson and Jain, 1994). The distance from set $A = \{a_1, a_2, \ldots, a_m\}$ to set $B = \{b_1, b_2, \ldots, b_n\}$ uses the same as classical Hausdorff distance, but Equations 3.5 and 3.6 are changed to Equations 3.9 and 3.10 respectively.

$$h(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B) \tag{3.9}$$

$$h(B, A) = \frac{1}{N_b} \sum_{b \in B} d(b, A) \tag{3.10}$$

N_a and N_b represent the number of black pixels in image A and B respectively. Instead of using the maximum of forward and reverse Hausdorff distances, the modified Hausdorff distance uses their average. From Figure 3.5, the forward Hausdorff distance, $h(A, B)$, is 0.11 (8/76), while the reverse Hausdorff distance, $h(B, A)$, is 0.18 (14/80). So the distance between two binary images, $H(A, B)$ is 0.18.

Greyscale Hausdorff Distance

Zhao *et al.* (2005) proposed a new measure called the greyscale Hausdorff distance (GHD). Unlike the other methods that match two binary images, the proposed method can match greyscale images that have few pixel values. This method transforms a binary image into a grey image, and then the Hausdorff distance is computed for the grey image. Therefore, after the binary image is transformed, the intensity of the random noise is reduced and the object is matched more easily and accurately.

To transform the binary image into a greyscale image, a 3×3 window is used. However, a larger window can also be used. In the binary image, '1' denotes the black pixel which forms the foreground, and '0' denotes the white pixel which forms the background. In

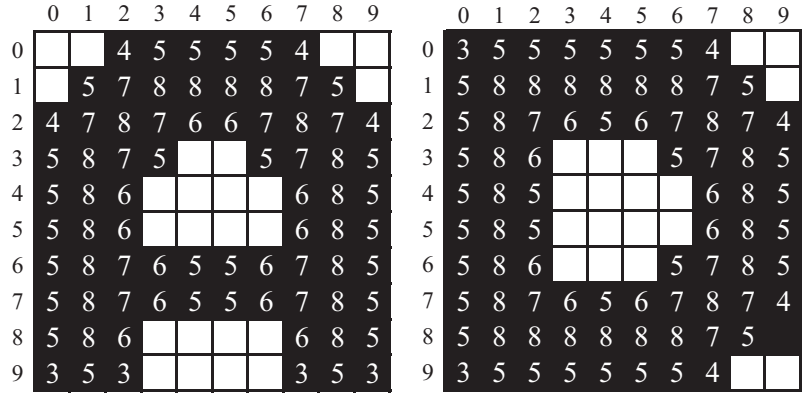


FIGURE 3.6: Transformation of binary into greyscale image by 3×3 window.

the window, if the central pixel is a black pixel and it has N black neighbour pixels, then this central pixel value is taken to be N; if the central pixel has $N-1$ black neighbour pixels, its value is taken to be $N-1$. An example is shown in Figure 3.6.

After the transformation, the distance between set $A = \{a_1, a_2, \ldots, a_m\}$ and set $B = \{b_1, b_2, \ldots, b_n\}$ uses the same as classical Hausdorff distance, but the distances from point a to set B and from point b to set A are defined as follows:

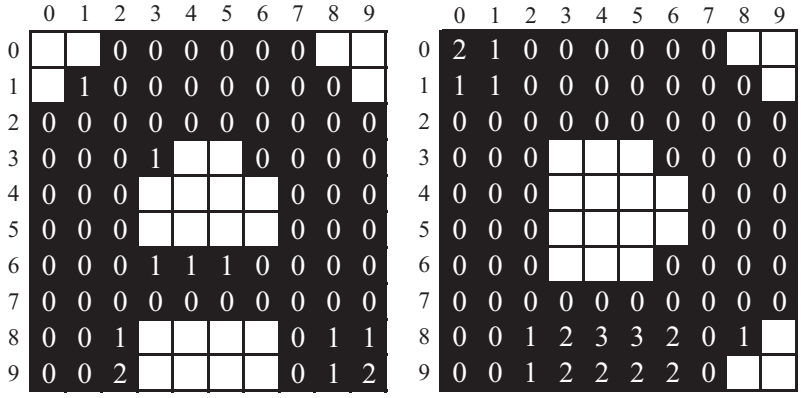
$$d(a, B) = \min \left[\min_{b_v = a_v} d(a, b), \min_{b_v = a_v - 1} d(a, b), \min_{b_v = a_v + 1} d(a, b) \right] \quad (3.11)$$

$$d(b, A) = \min \left[\min_{a_v = b_v} d(a, b), \min_{a_v = b_v - 1} d(a, b), \min_{a_v = b_v + 1} d(a, b) \right] \quad (3.12)$$

$$\text{where } d(a, b) = |a_x - b_x| + |a_y - b_y| \quad (3.13)$$

The distance between a in the point set A and b in the point set B is calculated where the value of a (a_v) and b (b_v) is equal and the values range between 1 and N ($1 \leq a_v \leq N$ and $1 \leq b_v \leq N$). Its two neighbour values $b_v - 1$ and $b_v + 1$ are also computed and then their minimum values are chosen. Finally $h(A, B)$, $h(B, A)$ and $H(A, B)$ are computed by Equations 3.5, 3.6 and 3.4 respectively. Figure 3.7 indicates the distance metric using greyscale Hausdorff distance. For the example, the greyscale Hausdorff distance between image A and B is 3.

Instead of the original greyscale Hausdorff distance as proposed by Zhao *et al.* (2005), a modified greyscale Hausdorff distance was employed in our evaluation. The maximum of forward distance, Equation 3.5 and reverse distance, Equation 3.6 are replaced with Equation 3.9 and 3.10 respectively. With the same example, the 'directed' distance of image A and B is 0.33, while the 'undirected' distance is 0.17. Hence the distance of image A and B is 0.33.



(a) Distances from A to B.

(b) Distances from B to A.

FIGURE 3.7: Distances between image A and B using greyscale Hausdorff metric.

3.5.3 Performance on Original Data

In this section, a comparison of the NECTEC printed OCR and the baseline classifier is shown. The evaluation is tested with the NECTEC printed character image dataset, as mentioned in Section 3.2. In the case of the k-NN classifier, we found the best value of k by varying k from 1 to 5 and tested it with a Euclidean distance metric. Our evaluation showed that the value of k equal to 1 gave the best result. Finally, a comparison of nearest neighbour classifiers using several distance metrics is presented and discussed.

Comparison of NECTEC OCR and Nearest Neighbour Classifier (XOR)

Experimental results with three different classifiers are presented in Figure 3.2. The classifiers are:

1. NECTEC classifier,
2. nearest neighbour classifier using exclusive-OR distance, and
3. standard back-propagation neural network classifier.

As seen in Table 3.2, the nearest neighbour classifier using exclusive-OR distance achieves a better recognition performance than the NECTEC classifier. This difference is small, but the calculated t (t_s) exceeds the tabulated t ($t_{0.5,9}$). So it can be considered to be statistically significant, at the 95% confidence level. However, the features used for training and testing were different. The NECTEC classifier used an 8×8 matrix (F08) and the aspect ratio of a character (AR), while the nearest neighbour classifier used a 32×32 binary image (B32). (Details of these features were given in Section 2.5.) It seems difficult to conclude that the nearest neighbour classifier using exclusive-OR distance is better than NECTEC classifier. It may be that it is the 32×32 binary image

Classifier	Recognition rate (%)	t_s
NECTEC	95.70±0.11	52.20
Nearest neighbour (exclusive-OR)	97.80±0.10	-
Back-propagation neural network	94.65±0.10	70.34

TABLE 3.2: Recognition rates and paired t-test ($t_{0.5,9} = 2.26$) of NECTEC, nearest neighbour classifier using exclusive-OR distance and back-propagation neural network on 10-fold cross-validation.

that accounts for the better recognition rate, so we should compare with the same input representation. As shown in Table 3.2, Back-prop is a standard back-propagation neural network trained and tested with 32×32 binary image, the same as with the nearest neighbour classifier, using exclusive-OR. The nearest neighbour classifier using exclusive-OR can achieve much higher accuracy than the Back-prop classifier and is statistically significant at the 95% confidence level.

As stated previously, the features used to train and test NECTEC and standard back-propagation neural networks are different. NECTEC uses an 8×8 feature matrix and aspect ratio, while standard back-propagation neural network is applied to a 32×32 binary image. A comparison of the NECTEC classifier and standard back-propagation neural network aims to study the performance of different features. The results show that the 8×8 feature matrix achieved a better recognition rate than the 32×32 binary image feature, in the case of using neural networks. There could be many reasons why the 8×8 feature matrix gives a better recognition rate over the 32×32 binary image feature. It may be due to the design of the neural network architecture. We tried to keep the original neural network architecture of the NECTEC OCR system. Hence the number of input nodes was only changed to 1,024 nodes, while the number of hidden nodes is still the same, 64 nodes. In the training process for the standard back-propagation neural network using the 32×32 binary image features, the mean square error (MSE) is approximately 0.03 to 0.04 at 2,000 training cycles and was still consistent at this level, but the MSE of the NECTEC classifier is lower than 0.01, as shown in Figure 3.8. It seems to indicate that the design of neural network architecture is not good enough. It is possible that, if the number of hidden nodes is increased, the MSE would decrease and achieve 0.01.

Nearest Neighbour Classifier using Several Distances

Table 3.3 shows a comparison of nearest neighbour classifiers using several distances, such as classical, modified and greyscale Hausdorff, as well as exclusive-OR. The figure shows the nearest neighbour, using modified Hausdorff distance can perform much better recognition performance. The calculated t (t_s) of the nearest neighbour classifier using modified Hausdorff distance, in comparison with the nearest neighbour classifiers using

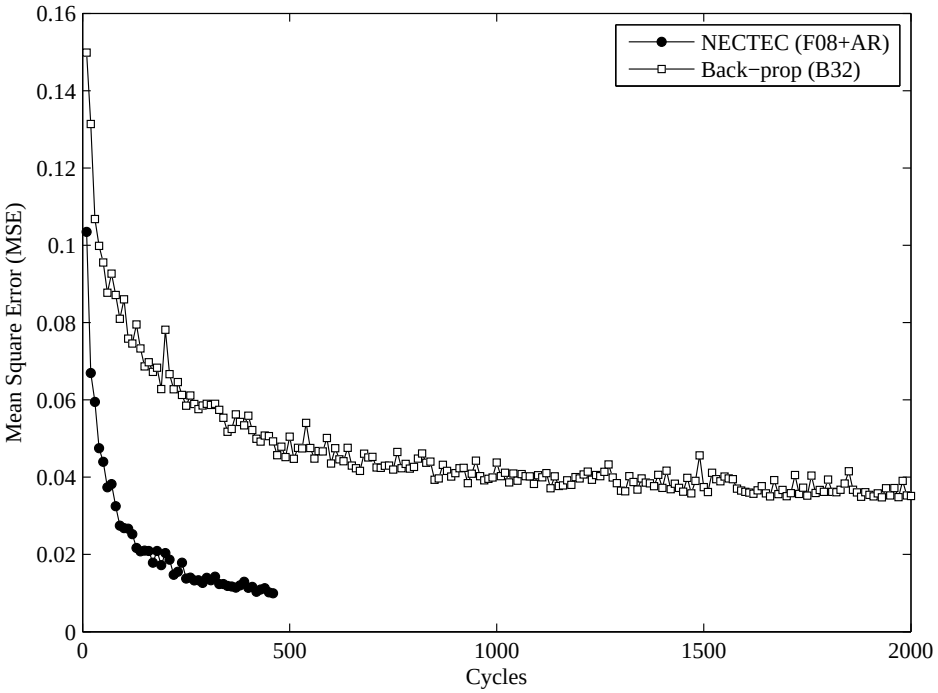


FIGURE 3.8: An example of the mean square error of the back-propagation neural network when trained with the 32×32 binary image features and the NECTEC classifier.

Distance metric	**Recognition rate (%)**	t_s
classical HD	96.30±0.12	46.75
modified HD	98.14±0.11	-
greyscale HD	97.99±0.12	5.84
exclusive-OR	97.80±0.10	8.21

TABLE 3.3: Recognition rates and paired t-test ($t_{0.5,9} = 2.26$) of nearest neighbour classifiers using several distance metrics on 10-fold cross-validation.

other distance metrics, is higher than the tabulated t ($t_{0.5,9}$). Thus, the difference is statistically significant at the 95% confidence level.

3.5.4 Misclassified Characters among the Nearest Neighbour Classifier

The most frequently misclassified characters among the nearest neighbour classifiers using the exclusive-OR, classical, modified and greyscale Hausdorff distances, are the relatively small and featureless characters. Most of them are special symbols, upper and lower vowels. Table 3.4 shows the 10 characters with the lowest recognition rate among the NECTEC and nearest neighbour classifiers using exclusive-OR, classical, modified and greyscale Hausdorff distances.

Classifier	Most frequently misclassified characters									
NECTEC	,	◌̊	.	/	j	◌ุ	*	’	◌ิ	◌ิ
1-NN (exclusive-OR)	,	-	’	.	'	◌ิ	|	◌ิ	◌ิ	◌
1-NN (classical Hausdorff)	,	◌ิ	◌	|	-	'	◌	.	ฑ	’
1-NN (modified Hausdorff)	,	-	’	'	.	|	◌	◌ิ	l	i
1-NN (greyscale Hausdorff)	,	-	’	'	.	|	◌ิ	◌ิ	◌ิ	l

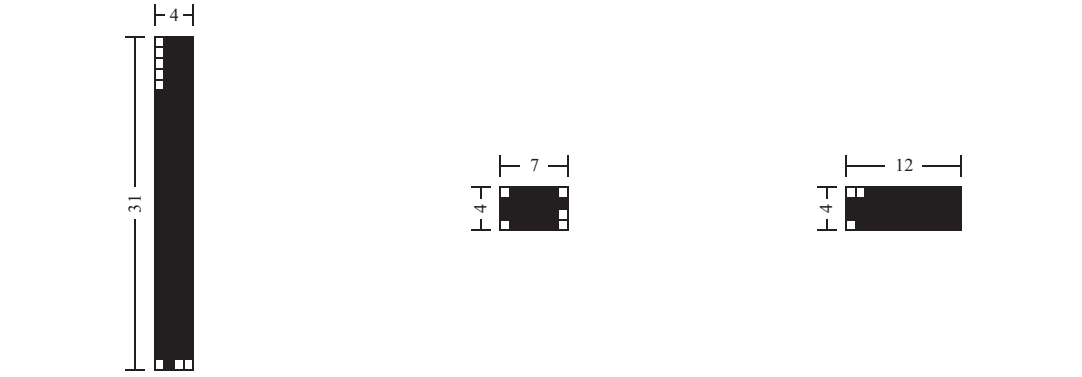
TABLE 3.4: The most frequently misclassified characters among the classifiers.

When these misclassified characters are converted into a 32×32 binary image and then into an 8×8 matrix, distinguishing features may be lost. Some characters are quite similar, such as ‘ ’ ’ (apostrophe), ‘ , ’ (comma), ‘ . ’ (full stop) and ‘ ◌̊ ’ (a Thai upper vowel). Although the original images of some characters are different such as ‘ - ’ (hyphen), ‘ | ’ (pipe) and ‘ ' ’ (a Thai tone marker), the 32×32 binary images of them are difficult to identify, because most of them are normalised into a black square.

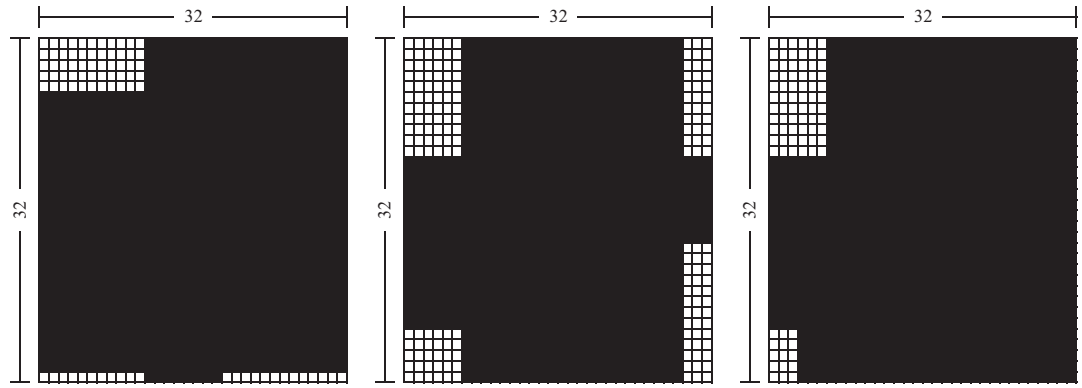
As shown earlier in Figure 2.2, a Thai sentence consists of four levels. Some characters (tone marks and above vowels) are written above other characters, while some are placed under the baseline. Hence, the position of a character compared with others, and the level of each character (in the case of Thai), can be a good description for classifying similar characters, for instance, the difference in the position and level between ‘ | ’ (pipe) and ‘ ' ’ (a Thai tone marker), ‘ . ’ (full stop) and ‘ ◌̊ ’ (a Thai upper vowel), and ‘ , ’ (comma) and ‘ ◌ุ ’ (a Thai lower vowel). These misclassified characters might be resolved.

Some misclassified characters, for instance, ‘ | ’ (pipe), ‘ . ’ (full stop), ‘ - ’ (hyphen) and ‘ , ’ (comma) cannot be resolved using this description, because their positions are at the same level (middle). Regardless of font they are very different characters and have totally different patterns in the scanned data. In fact, when normalised, they look the same, due to an artefact in pre-processing (size-normalisation). The normalisation technique disregards the size of an original character (width and height). Figure 3.9 shows an example of normalised characters in comparison with the original characters. To resolve this problem, the size of the original character should be considered. However, the width and height of the same character may be different when changing font size and scans in different resolutions. Hence its aspect ratio should be used rather than its size.

However, some misclassified characters cannot be dealt with using these techniques, such as ◌ิ, ◌ิ and ◌ิ. Their positions are at the same level (above the Thai consonant) and their sizes are approximately the same. In this case, linguistic post-processing, for example, spell checking and a part-of-speech (POS) n-gram model may be applied, which is beyond the scope of this research, since we only focus on the feature extraction and classification stages.



(a) An example of original images.



(b) An example of normalised images.

FIGURE 3.9: Problem of the normalisation technique which throws away the size of an original character. The example characters are ' | ' (pipe), ' . ' (full stop) and ' - ' (hyphen).

3.5.5 Robustness of Rotated and Noisy Data

The previous section presented, discussed and compared the NECTEC OCR engine and nearest neighbour classifiers. As described in Section 3.5.3, a simple classifier, the nearest neighbour classifier, yielded better accuracy than the much more complex NECTEC OCR engine on the original image characters. To compare their robustness, this section examines the performance of the classifiers using distorted data. In the evaluation, two types of distorted data, rotated and noisy, are utilised. In the case of noisy data, a random number generator was used to create salt-and-pepper noise, which was added into the image. The process of creating rotated and noisy data is described. The robustness and performance of the NECTEC OCR engine and the nearest neighbour classifier are presented and discussed.

Generation of Rotated and Noisy Data

This section describes the process of creating rotated and noisy rotated data. The rotated and noisy data are produced from the original NECTEC character image dataset.

(a) Original image. (b) 5° rotated image. (c) 10° rotated image.

FIGURE 3.10: An example of two degrees of rotated images.

First, the process of rotating the image will be introduced, followed by a description of salt-and-pepper noise.

In the case of rotating images, an original image was turned by several degrees (1 to 5° in steps of 1, 7 and 10°). It was performed by computing the inverse transformation for every destination pixel. Output pixels were computed using linear interpolation, then converted to the 32×32 binary image and 8×8 matrix features, respectively. An example of rotated images is shown in Figure 3.10.

Salt-and-pepper noise is a type of noise often seen in images. It represents itself as randomly occurring white and black pixels. The salt-and-pepper noise was introduced in an image by setting half of the corrupted pixels in the image (randomly selected pixels) to black and the other (randomly selected pixels) to white. For example, adding 10% salt-and-pepper noise into the image means 5% randomly selected pixels are set to black and the remaining 5% randomly selected pixels are set to white. For the random process, the Mersenne twister, a pseudorandom number generator was used (Matsumoto and Nishimura, 1998). To add salt-and-pepper noise to an original image, first, a number of corrupted pixels are computed from the size of an original image. Next, random x and y positions in an image are picked up. Then, half of them are changed to black, and the remainder set to white. Finally, a salt-and pepper noise image is fed into the normalisation process, and converted into the 32×32 binary image and 8×8 feature matrix respectively. Figure 3.11 displays an example of salt-and-pepper noise images.

(a) Original image. (b) 10% noise image. (c) 20% noise image.

FIGURE 3.11: An example of two levels of salt-and-pepper noise images.

Performance as a Function of Rotation

The performance of the classifiers on distorted data was investigated. The two conditions, rotation and additive noise, are covered. The first evaluation in this section focuses on a comparison of the classifiers on the original and rotated images, and the second, in the following section, concentrates on a comparison of the classifiers on the original and noisy images. The classifiers are trained with the original images and tested with the original and distorted image. As in the previous evaluation, 10-fold cross-validation is used to examine the recognition rates of the classifiers. For the nearest neighbour classifier, various distance metrics are used, such as exclusive-OR and the Hausdorff distance, with value of k equal to 1.

Figure 3.12 shows that the recognition rates of all classifiers decrease with increase in the degrees of rotation of the images as expected. The recognition rates from 1° and 2° rotated images are not dramatically different when compared with those of the original images. However, after 2° of rotation, the recognition rates start to decline rapidly. For example, the recognition rate of the nearest neighbour using classical Hausdorff distance drops over 13 percentage points when the rotation increases from 0° to 5° (from 96% to 83%). Similar, the NECTEC and standard back-propagation classifiers decrease from 96% to 82% and from 94% to 77%, respectively. Nevertheless, the nearest neighbours using exclusive-OR, modified and greyscale Hausdorff distances look similar, but they are more robust in rotation. Their recognition rates fell by approximately 5 percentage points when the images were rotated from 0° to 5° (from 98% to 92%, 98% to 95% and 98% to 95%, respectively). Among the tested classifiers the nearest neighbours using exclusive-OR, modified and greyscale Hausdorff distances yield good results for the original and rotated images.

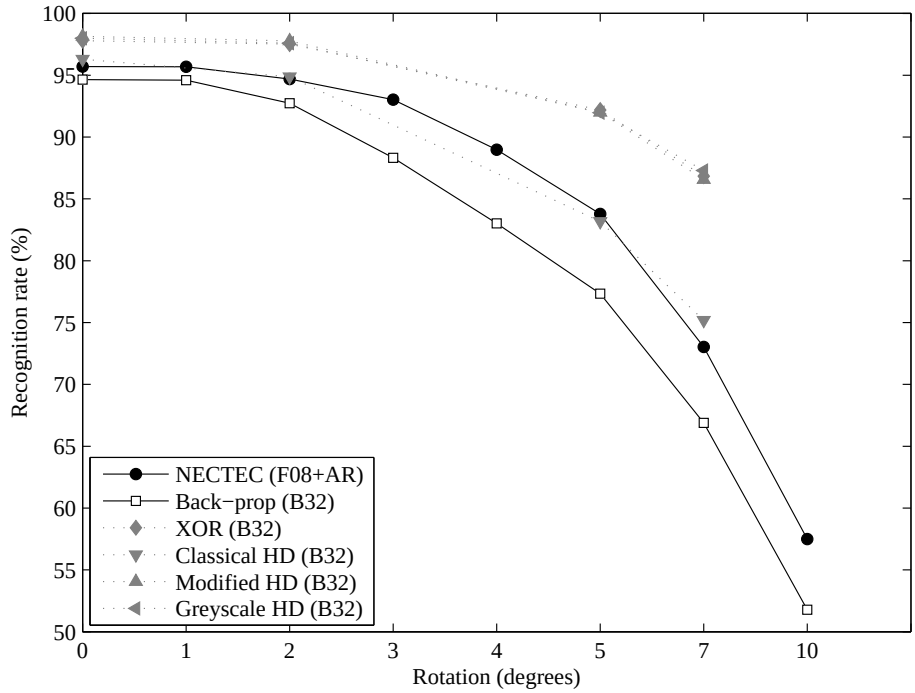


FIGURE 3.12: Recognition rates of the classifiers when tested with rotated images on 10-fold cross-validation.

Performance as a Function of Noise

As seen in Figure 3.13, the recognition rates of all classifiers decreased when salt-and-pepper noise was increased. For example, the NECTEC classifier yielded 96% accuracy for free-noise images, while testing with 10 and 20% noise gave recognition rates of 93% and 85%, respectively. By contrast, the recognition rates of the nearest neighbour classifier using exclusive-OR distance reduced very little when compared with those of the NECTEC classifier and nearest neighbour classifier using various Hausdorff distances. This classifier achieved accuracy of 97.65% in recognising the original images, and 97.56% and 97.24% with 10% and 20% noise, respectively.

In the case of nearest neighbour using classical Hausdorff distance, although the recognition rate at 0% noise was about 96% at 10% and 20% noise, the accuracy decreased rapidly with an average recognition rate of 26% and 13% at 10% and 20% noise, respectively.

From the experimental results, it is fair to conclude that the nearest neighbour using exclusive-OR distance is more tolerant of salt-and-pepper noise than the other classifiers examined. The addition of salt-and-paper noise slightly affected the performance of the nearest neighbour classifier using exclusive-OR, but had more impact on the performance of the nearest neighbour classifier using classical Hausdorff. As shown in Figure 3.14,

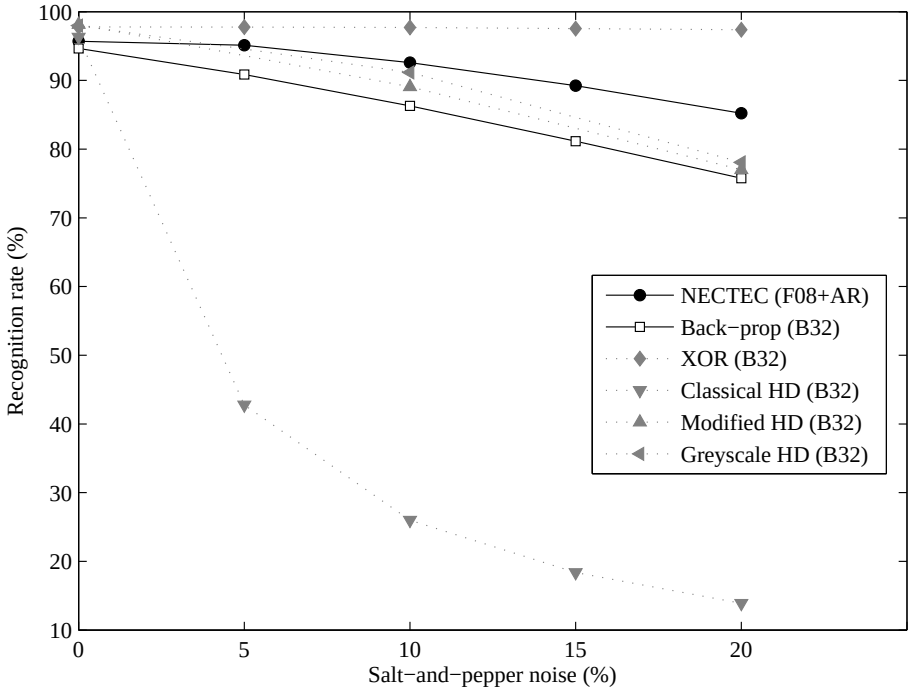


FIGURE 3.13: Recognition rates of the classifiers when tested with salt-and-pepper noise images on 10-fold cross-validation.

one pixel added to the example image can change the distance from 0 to 4, while the exclusive-OR distance only changes from 0 to 1.

Image A is an original image, while image B_2 is the original image with an added pixel (salt-and-pepper noise) in the top right corner. Image B_1 is another original image. If we would like to find the one that is similar to image A, the right answer should be image B_2, rather than B_1. The distances of image A to images B_1 and B_2 (calculated by exclusive-OR) are 4 and 1 respectively. In the case of classical Hausdorff distance, the distances between image A and B_1, and between image A and B_2 are the same value (4). From the distance values, if we use the nearest neighbour classifier, the answer is B_2 for the XOR distance and B_1 for the Hausdorff distance. The Hausdorff distance produced the wrong answer, due to only one pixel of noise being used. The computation of directed distance, $h(A, B_i)$, and undirected distance, $h(B_i, A)$, by seeking the maximum, is the problem.

To tackle the problem, modified Hausdorff distance is implemented. Instead of using the maximum of the forward distance and reverse distance, the averages are used. As seen in Figure 3.14, $H(A, B_1)$ and $H(A, B_2)$, using the modified Hausdorff distance are 0.77 and 0.40 respectively. This modification improves the recognition rate dramatically. Nevertheless, although the modified Hausdorff distance can improve the recognition rate dramatically compared with the classical Hausdorff distance, the XOR distance is still more robust for salt-and-pepper noise than the modified Hausdorff distance.

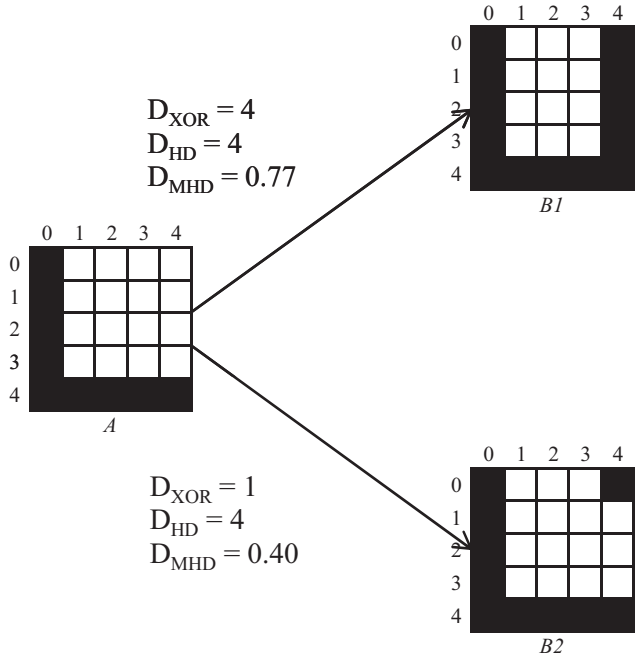


FIGURE 3.14: Problem of classical Hausdorff distance on noisy data.

Evaluation	**Recognition rate (%)**	
	With	Without
Aspect ratio	94.1±0.12	93.5±0.12
SOMs	95.7±0.11	94.1±0.12

TABLE 3.5: Summary of the evaluation results of the NECTEC printed OCR system.

3.6 Summary and Discussion

An evaluation of the NECTEC printed OCR system, focusing on its classification performance, was reported. The recognition performance was examined and some limitations of the system were found. The input of including aspect ratio (the ratio between the width and the height of a character) as a feature was evaluated. The results show that addition of the aspect ratio statistically improved the recognition performance by about 0.5 percentage points. An evaluation of Kohonen self-organising maps (SOMs), a rough classifier for NECTEC printed OCR, was performed. The comparison between the system with and without SOMs shows that the system consisting of SOMs as a rough classifier statistically produced a higher recognition rate by about 1.5 percentage points. The evaluation results are summarised in Table 3.5.

A comparison of NECTEC and simple classifiers was performed in Section 3.5.3. A very straightforward classifier, the nearest neighbour, was selected as baseline. Several distance metrics, such as exclusive-OR and Hausdorff distances, were used to evaluate performance and explore the problems of different distance metrics. Our evaluation shows that the nearest neighbour classifier using Hausdorff distance yielded a significant

Classifier	**Recognition rate (%)**
NECTEC	95.7±0.11
Back-propagation neural network	94.7±0.10
Nearest neighbour (exclusive-OR)	97.8±0.10
Nearest neighbour (classical HD)	96.3±0.12
Nearest neighbour (modified HD)	98.1±0.11
Nearest neighbour (greyscale HD)	98.0±0.12

TABLE 3.6: Summary of the results of the NECTEC and nearest neighbour classifiers.

improvement in recognition rate over the NECTEC classifier. An approximate correct recognition rate of 98% was achieved on the NECTEC printed character image dataset. Table 3.6 summarises the results of the NECTEC and nearest neighbour classifiers.

In Section 3.5.5, the robustness of the classifiers was tested. Two evaluations were conducted: the performance of the classifier on rotated and noisy data. The nearest neighbour classifier using exclusive-OR gave better results for recognition rate over other classifiers. Although the nearest neighbour classifier performs with good accuracy, some limitations still exist. For example, the most frequently misclassified characters are relatively small and lacking significant structures. However, the problems can be solved using a combination of several techniques, as reported in Section 3.5.4. For example, the position and level of a Thai character, the separation between English and Thai characters, the size-normalisation technique and aspect ratio between the width and height of original characters, and spell-checking together with a part-of-speech (POS) n-gram model, can be used. Another limitation is that the nearest neighbour classifier consumes vast computational resources, at the expense of longer recognition times and huge disk space.

Knowledge of Thai machine printed OCR will be applied to offline handwritten Thai character recognition, which tackles a similar, but more difficult, problem (static images). Section 2.2 described several groups of Thai characters that have similar shapes and only very small differences in details, such as loops and curls. These details are well preserved in printed Thai characters. However, they are often neglected in several handwriting styles. Therefore, in the next chapter, handwritten, instead of printed Thai character databases, are used to evaluate the classifiers.

Chapter 4

Handwritten Thai Character Recognition and Evaluations

This chapter gives an introduction to handwritten Thai character recognition (Section 4.1). The collection procedure for three different databases of handwritten Thai characters and their description are presented in Section 4.2. Training and evaluation methods are described in Section 4.3. An introduction to the features and architectures of classifiers is presented in Section 4.4. The nearest neighbour classifier (Section 4.4.1) and a back-propagation neural network (Section 4.4.2) that performed reasonably well on printed Thai character recognition (as shown in Chapter 3) need to be put in place, followed by HMMs in Section 4.4.3. An investigation of the classifiers and their performance is presented and discussed in Section 4.5. Summary and discussion of the chapter are addressed in Section 4.6.

4.1 Introduction

Handwriting is a natural form of human communication that has existed for centuries. It is involved in many of our day-to-day activities, such as note taking, form filling and letter addressing. During the past two decades, there have been increasing demands for applications to process automatically the content of these handwritten documents, as a supplement to the tasks performed by humans. In connection with that, handwriting offers an attractive and efficient method to interact with a computer, as witnessed from the recent advent of many hand-held devices that accept handwritten inputs, such as a tablet personal computer. These devices also require such applications to recognise handwritten entries. As a result, research in handwritten recognition becomes increasingly important.

While the term 'character recognition' is generally concerned with the problem of recognising any form of characters, 'handwriting recognition' specifically aims at the

more difficult problem of recognising handwritten characters. The ultimate goal of handwriting recognition research is to develop systems that can read any text with the same recognition accuracy as humans, but at a faster rate. Handwriting recognition is a difficult task, due to its large problem dimension, which is composed of several sub-problems and the generally ambiguous nature of handwriting itself. There have been a large number of handwriting recognition research projects trying to tackle different aspects of the problem. The performance of handwriting recognition systems has improved dramatically over the past decade, especially in some specific tasks, such as handwritten address reading and the recognition of numbers on bank cheques. However, the recognition of the general handwritten word and sentence is still poorly done by machine and not comparable to that achieved by humans.

A good handwriting recognition system should have the ability to cope with variations in writing styles, while being capable of distinguishing similar yet different characters, both are equally important for the recogniser to achieve this facility. This thesis focuses mainly on the classification stage in order to improve the performance of the overall system. Appropriate features for the proposed classifiers are also taken into account. Offline handwritten work, rather than online, is the focus. In online handwritten character recognition, the writer can adapt to the recognition system to improve performance, although this may not be ideal. By contrast, no such adaptation is possible with offline character recognition. Moreover, in contrast to the printed character, handwritten characters vary widely from person to person (inter-person variation), and from the same person at different times (intra-person variation).

For handwritten Thai, several groups of Thai characters have similar shapes, for instance, ก, ถ, ฦ and ฦ, and vary only in small details such as loops and curls, for example, ค, ด and ต. These details are well preserved in printed Thai characters. However, they are often neglected in handwriting. Thai handwriting styles can be broadly categorised into two groups: constrained and unconstrained. The constrained or handprinted style is what children learn at schools from first grade. It is intended to be analogous to printed Thai characters. Each character is written neatly, showing the distinct features of the characters clearly. On the contrary, with unconstrained handwritten styles, characters are written less carefully and important details may be omitted. Loops are not fully closed or are written as blobs. Curls are likewise eliminated or attenuated. Figure 4.1 shows samples of constrained and unconstrained Thai handwriting styles. When writing at high speed, such as in note-taking, the additional difficulty of touching characters may arise. Touching can be divided into two types: the internal touching stroke in each character (intra-touching) and touching among characters (inter-touching). Moreover in the case of inter-touching, Thai characters can occur both horizontally and vertically. Therefore, handwritten Thai is also a more challenging topic for automatic handwriting recognition.

(a)

(b)

FIGURE 4.1: Examples of a handwritten Thai phrase in different writing styles: (a) constrained style of writing, (b) unconstrained style of writing.

In this chapter, knowledge of machine printed Thai OCR, gained in the work described in earlier chapters, will be applied to handwritten Thai character recognition. Hence, we will focus on the recognition of unconstrained-style handwritten Thai characters, without inter-touching characters being involved, or in other words, offline isolated Thai handwriting recognition. The research also deals with variations in handwriting style from multiple writers and internal touching strokes within each character. Form filling and formal letter writing are examples of this writing style. This study focuses mainly on the classification stage of handwriting recognition, with the assumption that the data have been pre-processed and segmented into isolated characters. However, it is inevitable that the feature extraction process has to be taken into account when designing the recogniser. Hence, effective feature extraction methods for Thai handwritten recognition have also been studied. A comparison with previous work will be carried out.

4.2 Thai Character Databases

This section introduces three published databases of Thai handwriting and discusses the various advantages and disadvantages of each. First, the database called 'ThaiCAM' is presented. Then the two databases collected by NECTEC are described. They are an online and offline handwritten character databases. In this thesis, they will be named 'NECTEC-ON' and 'NECTEC-OFF', respectively. An overview of the databases is given in Table 4.1. The following evaluations in this research are conducted on them. Their specifications and collection methods are described as follows.

4.2.1 ThaiCAM Offline Handwritten Character Database

Nopsuwanchai and Povey (2003), Nopsuwanchai (2005) and Nopsuwanchai *et al.* (2006), state that the ThaiCAM database was produced in 2001 at the Computer Laboratory,

Details	ThaiCAM	NECTEC-ON	NECTEC-OFF
Database type	Offline	Online	Offline
Number of classes	77	156	79
Number of writers	20	63	71
Samples per class	5 – 9	3	2
Resolution (DPI)	300	n/a	200
Width of stroke	Varied	3 pixels	Varied
Size on average (pixels)	1,572	2,852	713

TABLE 4.1: Comparison of three handwritten databases (information collated from Nopsuwanchai and Povey (2003) and Sae-Tang and Methasate (2004)). Note that the width of stroke in ThaiCAM and NECTEC-OFF are varied according to the written tool, while the width of stroke in NECTEC-ON is fixed at three pixels, because it is simulated from the temporal information generated by the digital tablet.

University of Cambridge. It was collected from 20 Thai writers who were studying at universities in England, France and Germany at that time. They wrote isolated characters in a specially prepared form in an unconstrained writing style. The form contains an example of printed Thai characters and an empty area for handwritten samples to be filled in. Figure 4.2(a) shows a part of the form after the pre-processing stage. Writers were instructed to write in their own style, resulting in a wide variety of styles, as seen in Figure 4.2(b). Each person contributed at least 5–9 handwritten samples for each class. There were approximately 130 tokens for each class. The forms were then scanned on a desktop scanner at 300 dots per inch resolution and stored as 8-bit greyscale images. Pre-processing techniques for noise removal and thresholding were applied to the scanned forms. The character images were extracted from the boxes, using a two-pass algorithm (Jain *et al.*, 1995), in which a label was assigned to each pixel in the first pass, with label equivalence based on eight-neighbour connectivity. Equivalent classes were determined, and a second pass updated each pixel in a connected component with a label unique to that component. Strokes intersecting pre-printed lines on the form were adjusted manually. Finally, a 5-pixel white space was added around the characters. The average number of pixels (weight by height) of the character images was 1,572.

4.2.2 NECTEC Online Handwriting Database

Before studying at the University of Southampton in 2007, I worked at the National Electronics and Computer Technology Center (NECTEC), the Thai government's science and technology research organisation, and produced the Thai handwritten character corpus (Sae-Tang and Methasate, 2004). It consists of two databases, an online and an offline database. The NECTEC-ON database will be described in this section and NECTEC-OFF in the following.

(a)

(b)

FIGURE 4.2: Examples from the ThaiCAM database: (a) A part of the form for collecting handwritten samples from a writer, (b) A different realisation of a single character from three different writers.

Data from NECTEC-ON was divided into three sets. The first, a Thai character set, consisted of consonants, vowels, tones and Thai digits. The second set was English characters, including uppercase and lowercase characters and digits. The last set consisted of special symbols, such as full stops, commas and question marks. It was collected from 63 native writers, 28 males and 35 females. The ages of the participants were between 20 and 29. Each person wrote three handwritten samples for each class. Hence, there were 189 tokens for each class. An electronic tablet (WACOM intous 6×8) was used with a specific software developed by NECTEC. The software displays characters in the left window, sequentially. Writers operate the tablet with an electronic pen, and the written strokes are shown in the right window. The software captures a signal $(x, y$ coordinates) from the tablet for collecting data. It simulates the written strokes from the temporal information generated by the tablet. The width of the simulated strokes is fixed at three pixels. Data from the tablets are composed of two parts, a bi-level image of 150×200 pixels and temporal information. Because my interest is in offline handwritten recognition, I use only the static images of handwritten characters without the temporal information. The size of the character images on average is 2,852 pixels. For this thesis, some white spaces (pixels) were added to these images in the same manner as in the ThaiCAM database.

4.2.3 NECTEC Offline Handwritten Character Database

In the same year, NECTEC also released the Thai offline handwritten character database named 'NECTEC-OFF'. It covers three sets: isolated handwritten characters, handwritten words and handwritten sentence sets. The isolated handwritten character set includes Thai consonants, vowels, tones and Arabic numerals. Province names[1] and numeral words were included in the handwritten word set. The last set is a sentence set, including 'legal amounts'[2] and Thai general articles. In this thesis I focus only on the isolated character set. A sample of collected handwritten data was specifically designed, as shown in Figure 4.3. It contains several empty boxes for handwritten samples to be filled in. It was collected from 71 native writers. The ages of participants were between 20 and 29. Some were in the same group as NECTEC-ON. They were instructed to write in an unconstrained style and within the box. They were also recommended to use a pen, because it is a widely-used implement and line patterns are clearer. However the participants were allowed to use various pen sizes and colours for a more complete range of data. These resulted in a wide variety of styles. Each person wrote two handwritten samples for each class. Hence, there were 142 tokens for each class. Next, the form was digitised on an HP ScanJet Pro 6300 desktop scanner at 200 dots per inch resolution and stored as 8-bit greyscale images. Noise removing and thresholding were then applied to the scanned forms and the data extracted from the boxes, using the same algorithm as explained in Section 4.2.1. The average size of the character images was 713 pixels. Finally, for this research, some white space pixels were added to the images, in the same manner as in the ThaiCAM database.

4.3 Training Method

As described in Section 4.2, three Thai handwritten databases were used in the evaluation, ThaiCAM (Section 4.2.1), NECTEC-ON (Section 4.2.2) and NECTEC-OFF (Section 4.2.3). These databases are separated into training and testing sets of approximately equal size. This approach was chosen in order that our results can be used in comparison with the Nopsuwanchai (2005) results. Moreover, there is inadequate data to make 10 folds statistically significant. For ThaiCAM, the first three handwritten samples for each particular character from all participants were chosen to be the training set, leaving the rest to be the testing set. For NECTEC-ON, three samples per class were gathered from the writers. The first two samples were the training set, while the third sample was a testing set. For NECTEC-OFF, each volunteer wrote two samples per class. So the first token is a training set and the second is a testing set. However, the databases were collected from different organisations. As seen in Table 4.1, the

[1] Similar to a city name in the UK.

[2] a textual format or the spelling out of a number, for example, "100" written as "one hundred". It is commonly used on cheques. Below the "Pay To" line is a blank line. This line is used to write out the amount of the cheque in long form (in word form).

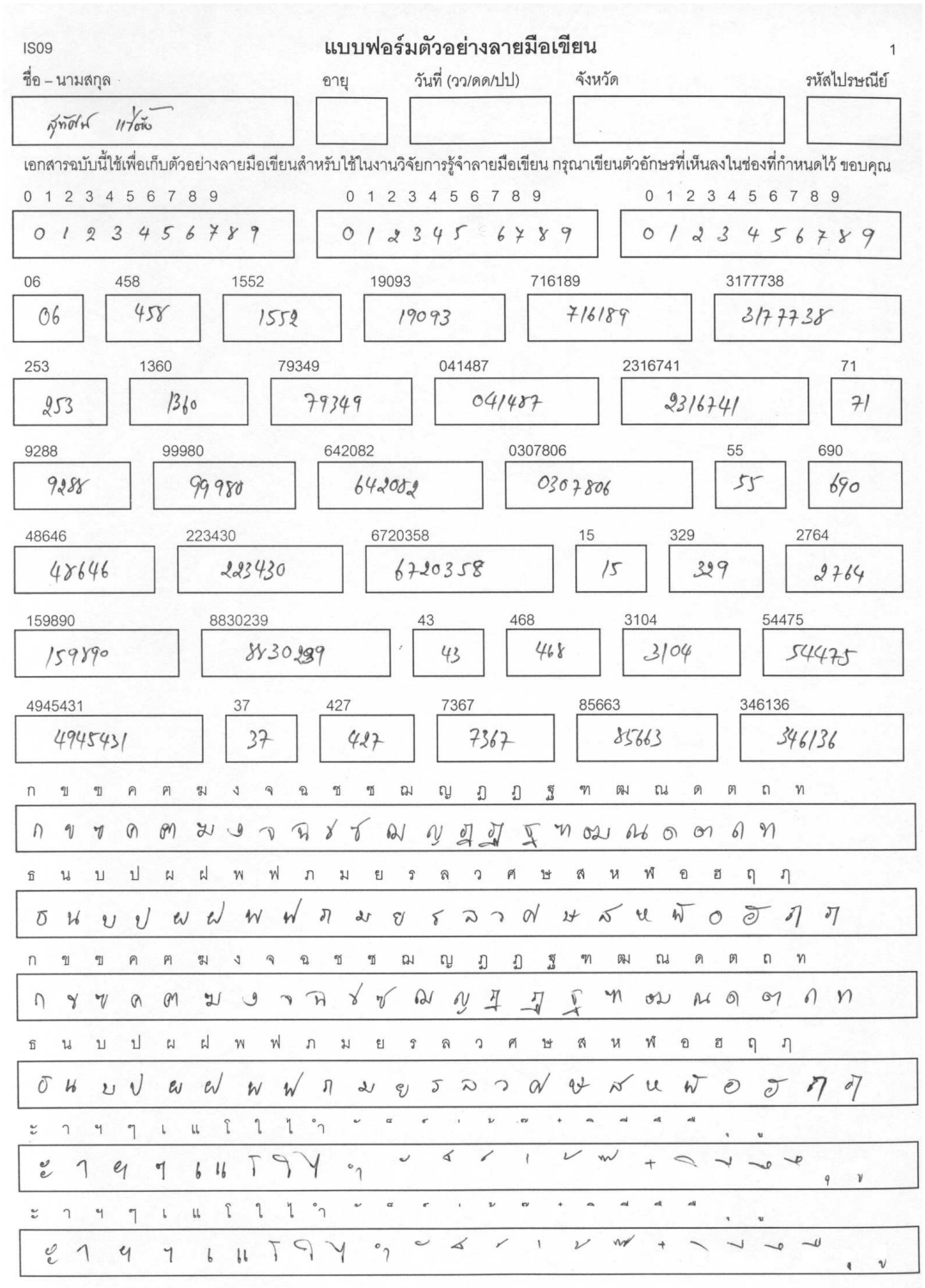


FIGURE 4.3: Example of the form for collecting handwritten samples of the NECTEC-OFF database.

Database	Character categories	Number of classes	Number of tokens		
			Training	Testing	Total
ThaiCAM	Middle Level	54	3,240	3,626	6,866
		10	600	569	1,169
	Above/Below	13	780	760	1,540
	Total	77	4,620	4,955	9,575
NECTEC-ON	Middle Level	54	6,804	3,402	10,206
		10	1,260	630	1,890
	Above/Below	13	1,638	819	2,457
	Total	77	9,702	4,851	14,553
NECTEC-OFF	Middle Level	54	3,834	3,834	7,668
	Above/Below	13	923	923	1,846
	Total	67	4,757	4,757	9,514

TABLE 4.2: Details of the Thai handwritten databases used in the evaluations. Note that the middle level characters consist of 54 characters (consonant and middle vowels), and 10 Thai digits (except NECTEC-OFF database).

number of classes in the databases is not equivalent. ThaiCAM includes only Thai consonants, vowels, tone marks and Thai digits, while NECTEC-ON consists of Thai and English characters. NECTEC-OFF collected only Thai consonants, vowels, tone marks and Arabic numerals. Thai digits are excluded from the NECTEC-OFF database, but we used some of them (Thai consonants, vowels, tone marks and Thai digits) for the evaluation. Thai is a multi-level language. Some characters (tone marks and above vowels) are written above other characters, while some are placed under the middle level, as seen in Figure 2.1 and 2.2. Hence, they can be classified by level into two groups, middle level and above/below level. The numbers of tokens in the three databases are summarised in Table 4.2. The evaluations described in the following sections were carried out for this training method.

4.4 Features and Recognition Modules

As stated previously, Thai characters pose a unique challenge for automatic handwritten recognition, on the grounds that there are several groups of Thai characters that have similar shapes and vary only in small details, such as loops and curls. The majority of previous studies rely on the presence of these details as important clues to indicate the identity of the character (Choruengwiwat *et al.*, 1998; Phokharatkul and Kimpan, 1998, 2002; Pomchaikajomsak and Thammano, 2003; Phokharatkul *et al.*, 2005, 2006). However, in an unconstrained-writing style, loops are not fully closed or are written as blobs; curls are likewise elided or attenuated. Our aim is to solve this problem by not relying on the precise information of loops and curls, and to make use of

efficiently computed global features with a trainable classifier, instead of searching for specially defined local structural features. Some previous research taking on this approach exists, such as Therramunkong *et al.* (2002), Nopsuwanchai and Povey (2003) and Nopsuwanchai *et al.* (2006).

We now describe the three classifiers used in the evaluation:

1. Nearest neighbour algorithm with exclusive-OR distance (XOR) (Section 4.4.1),
2. Feed-forward neural network with back-propagation learning algorithm (Section 4.4.2), and
3. Hidden Markov models with maximum likelihood estimation (Section 4.4.3), respectively.

The classifiers used different feature vectors. The following section describes their principal concepts and features in detail.

4.4.1 Nearest Neighbour Algorithm

As described in Section 3.5.1, the nearest neighbour algorithm is a non-parametric classification technique that classifies an unknown pattern to the class of its nearest neighbour in the reference data. It is simple and widely-used in pattern recognition and machine learning (Garain and Chaudhuri, 2003; Cha *et al.*, 2005; Lee and Coelho, 2005; Pati and Ramakrishnan, 2007; Zhang *et al.*, 2008). An evaluation of the nearest neighbour classifier with many distance metrics was performed and it was found that the nearest neighbour classifier with exclusive-OR distance (XOR) achieves very good performance on NECTEC printed Thai and English databases. It yields the best classification rate on original as well as on rotated and noisy data, so it will also be applied to the handwritten character recognition problem, as a baseline classifier. The principal concept can be found in Section 3.5.1 and 3.5.2. For the feature, an original character image, with a 5-pixel white space added around, was size-normalised to give a bi-level image (black and white) of 64×64 pixels. This was used as a template for characters in the training set.

4.4.2 Back-propagation Neural Networks

Section 2.5 presented the back-propagation neural network as a main classifier in NECTEC printed OCR. An evaluation of NECTEC printed OCR was presented in Section 3.4, especially the classification engine, and it was found that the back-propagation neural network is an essential classifier for the NECTEC printed OCR system. It is also used for comparison among classifiers. The features used with the

back-propagation neural network are the same as in the NECTEC OCR system, but the Kohonen self-organising map was not employed for rough classification. An original character image was size-normalised into a 32×32 binary image and then converted by counting black pixels in the 4×4 grid, to give an integer (n) representation. Finally, each value n, $0 \leq n \leq 16$, is divided by 16 to give floating point values. An example of converting from an original image to an 8×8 matrix feature was shown in Section 2.6. The 64 floating point values and the aspect ratio (the ratio between width and height of the original character image) were formed into a NECTEC feature set. All character samples were converted into a NECTEC feature set, which was subsequently used in the training and recognition processes. Training and evaluation of back-propagation neural networks was carried out using the Stuttgart Neural Network Simulator (SNNS), version 4.1 (Zell *et al.*, 1995), with additional implementations to prepare the feature vector sequences from the character images.

4.4.3 Hidden Markov Models

HMMs are a powerful statistical method for characterising data samples of a discrete time series (Jelinek, 1998). Data samples can be continuously or discretely distributed, and can be either scalar or vector. HMMs have become the most popular method for modelling human speech, and are used successfully in automatic speech recognition, speech synthesis, statistical language modelling and other areas of artificial intelligence and pattern recognition problems over decades (Young *et al.*, 2009). Recently, HMMs have also been used in computer vision applications, such as object tracking and character recognition. A number of reports regarding HMMs in character recognition have been presented and have proved their success (Methasate and Sae-Tang, 2002; Khorsheed, 2003; Nopsuwanchai and Clocksin, 2003; Pechwitz and Margner, 2003; Margner *et al.*, 2005; El-Hajj *et al.*, 2005).

Hidden Markov models are an extension of the 'Markov chain'. Instead of each state corresponding to a deterministically observable event, the hidden Markov model features a non-deterministic process that generates output observation symbols in any given state. The observation becomes a probabilistic function of the state. In this way, hidden Markov models can be regarded as a double-embedded stochastic process with an underlying stochastic process (the state sequence).

A hidden Markov model is essentially a Markov chain, where the output observation is a random variable X generated according to an output probabilistic function associated with each state. There is no longer a one-to-one mapping between the observation sequence and the state sequence. For a given observation sequence, the state sequence is not directly observable, hence the name 'hidden Markov models'. Formally, a hidden Markov model is defined by:

- An output observation, $O = \{o_1, o_2, ..., o_M\}$. The observation symbols correspond to the physical output of the system being modelled.
- A set of states representing the state space, $\Omega = \{1, 2, ..., N\}$. Here s_t denotes the state at time t.
- A transition probability matrix, $A = \{a_{ij}\}$, where a_{ij} is the probability of taking a transition from state i to state j.

$$a_{ij} = P(s_t = j | s_{t-1} = i) \text{ where } 1 \leq i, j \leq N \tag{4.1}$$

- An output probability matrix, $B = \{b_i(k)\}$, with $b_i(k)$ the probability of emitting symbol o_k when state i is entered. Let $X = X_1, X_2, ..., X_t, ...$ be the observed output of the HMM. The state sequence $S = s_1, s_2, ..., s_t, ...$ is not observed and $b_i(k)$ can be rewritten as follows:

$$b_i(k) = P(X_i = o_k | s_t = i) \text{ where } 1 \leq i \leq N \tag{4.2}$$

- An initial state distribution, $\pi = \{\pi_i\}$.

$$\pi_i = P(s_0 = i) \text{ where } 1 \leq i \leq N \tag{4.3}$$

A complete specification of an HMM thus includes two constant-size parameters N and M, representing the total number of states and the size of observation, the observation O and three probability matrices: A, B and π. The complete HMM can be denoted by:

$$\Phi = (A, B, \pi) \tag{4.4}$$

To use HMMs for pattern recognition, the evaluation problem needs to be solved, which will provide a method to determine how well a given HMM matches a given observation sequence. The likelihood $P(X|\Phi)$ can be used to calculate the a 'posteriori' probability $P(\Phi|X)$ using Bayes' rule, and the HMM with the highest probability can be determined as the pattern for the best observation sequence. Solving the decoding problem will make it possible to find the best matching state sequence given an observation sequence (i.e., the hidden state sequence). This is essential to automatic speech recognition. If the learning problem can be solved, the model parameters Φ can be automatically estimated from the training data (Rabiner, 1989).

HMM Topology

An effective way to explain the HMM is to view it as a finite state machine which generates an output segment O_t at each time step t while being at state s_t and changes

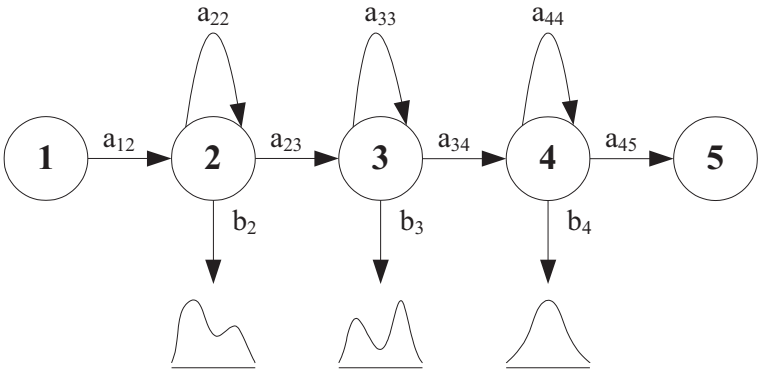


FIGURE 4.4: An HMM having left-right topology as often used in speech and handwritten recognition.

to the next state s_{t+1} once every time step $t+1$. This process continues until the model reaches the maximum time step T. As a result, a sequence of output segments and the observation sequence O^T is produced, where $O^T = \{o_1, o_2, ..., o_t, ..., o_T\}$. Every state in the HMM can be reached from every other state of the model, including itself, thereby resulting a fully-connected (or ergodic) model topology. However, the left-right model topology is normally used since it is more capable of modelling a time-varying signal. Its important property is that, as the time step increases, the state can only be moved to either the same state or the state that has an increasing index, i.e., proceeding from left to right. The N-state left-right model is forced to start at some start state and end in some end state at the time step T. The left-right model topology is widely used in HMMs for speech and handwritten recognition, and so it is in this work. Additional constraints are usually imposed to limit large changes of the state index occurring in each transition. In our models, no transitions of more than one consecutive state are allowed.

Figure 4.4 illustrates an example of the 5-state left-right HMM common to many speech and handwriting recognition systems. It also demonstrates two important properties of HMM. First, the transition from one state to another state in an HMM is determined by a probabilistic transition function. This function is described by an $N \times N$ matrix of discrete probabilities a_{ij}, known as the state transition probability matrix A. Second, as opposed to the Markov model, it is shown that every state in the HMM is associated with a probability density function for the production of particular observations. In practice, non-emitting states that do not generate outputs are often placed in the model as the start and end states, as shown in Figure 4.4. These states assume no increments of the time step t while entering the state and are useful for the connection of several HMMs. A $1 \times N$ matrix B contains a collection of all states' probability density functions, b_j.

The number of internal states of an HMM can vary according to the problem. For speech recognition, three to five states are commonly used for representing a 'phoneme'. If the HMM represents a word, a significantly larger number of internal states is

required. Depending on the pronunciation and duration of the word, this can be 15 to 25 states. More complex transitions between states than the simple topology illustrated in Figure 4.4 are also possible. If skipping of states is allowed, the model becomes more flexible, but also harder to train properly. For this work, however, HMMs are used to represent a whole character image containing more data than a phoneme. A small HMM topology (three to five states) seems to be insufficient. More internal states can be applied, the details will be discussed in the next section.

Features for Hidden Markov Models

The hidden Markov models and some feature extraction techniques used in Nopsuwanchai and Povey (2003), Nopsuwanchai (2005) and Nopsuwanchai *et al.* (2006) are used. A left-to-right topology is used, in which each state has a transition to itself and the next state, as shown in Figure 4.4. The features used with HMMs consist of three feature sets, 64×64 binary, polar transformed and clockwise 90-degree images. The 64×64 binary image is the original character image size-normalised into 64×64 pixels. The polar transformed and clockwise 90-degree images are created using the original image and then the aspect ratios and sizes of the resulting images are size-normalised to give bi-level images of 64×64 pixels. The polar transformation is widely used in computer vision research and, recently, in character recognition. It is a mapping from points in the image $f(x, y)$ (Cartesian coordinates) to points in the polar image $g(r, \theta)$ (polar coordinates). By defining an origin $O = (o_x, o_y)$, which is given by the centroid $(o_x = \bar{x}, o_y = \bar{y})$ of the image, the mapping is described by:

$$r = \frac{\sqrt{(x - o_x)^2 + (y - o_y)^2}}{d}, \tag{4.5}$$

$$\theta = \arctan\left(\frac{y - o_y}{x - o_x}\right), \tag{4.6}$$

where d is defined as the maximum distance between O and any pixel in f. The maps are then normalised to the size of 64×64 pixels. Examples of the 64×64 image together with its polar transform and clockwise 90-degree rotated versions are illustrated in Figures 4.5(a), 4.5(b) and 4.5(c) respectively. These three images are concatenated together to form a new image called a 'composite image', as depicted in Figure 4.5(d). All character samples are converted into composite images, which are later used in the training and recognition processes.

When the HMM-based recogniser is used, it is necessary to convert the static character image into a sequence of observations, as required by the HMMs. The conversion process may be carried out in two ways: the analytical and the sliding-window approaches. The analytical approach relies on topological features that describe the structural properties of the characters, for example, a sequence of edges and loops from the skeleton graph of a character or word (Bunke *et al.*, 1995; El-Yacoubi *et al.*, 1999; Khorsheed, 2003; Koerich

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

FIGURE 4.5: Examples of transformed character images: (a) 64×64 image, (b) polar transformed image, (c) clockwise 90-degree rotated image and (d) composite image.

et al., 2003). This approach is deemed inapplicable to our recognition tasks, because many Thai characters have a relatively similar shape which can only be distinguished by small details. Moreover, some important details in the characters may be diminished during the skeletonisation process.

On the other hand, the sliding-window approach is a more general method, which is widely used in the literature, for both cursive (Pechwitz and Margner, 2003; El-Hajj *et al.*, 2005) and isolated character recognition (Nopsuwanchai and Povey, 2003). In this instance, a narrow window is applied over the character image. The portion of the image covered by the window, called a frame, is captured, while it is moving from left to right, sometimes from top to bottom or in diagonal directions. A sequence of image frames is obtained as a result. An advantage of this approach is that it allows various types of features to be extracted from the image frame, not restricted to only structural features. The left-to-right sliding direction is also appropriate for HMMs with a left-to-right topology.

In this research, we utilise the sliding-window approach. The window, which has a width of w pixels and a height of character images H, is moved from left to right with a step size of o_h pixels. Given a character image of size $W \times H$, every two successive frame overlaps by $w - o_h$ pixels, and a sequence of $\frac{W-w}{o_h} + 1$ frames is generated. The optimal values of the window width and step size are task- and data-dependent, and thus can only be determined by evaluation. In preliminary experiments of Nopsuwanchai and Clocksin (2003), setting w and o_h to 4 and 1, respectively, gives reasonable recognition results. The size of the character images from the Thai handwritten database in Section 4.2 is 64×64 pixels, combined with the polar transformed version and 90-degree rotation of itself ($W = 192, H = 64$). As a consequence, the character images are transformed into a sequence of 189 frames (size 4×64 pixels), each of which is delivered to a subsequent feature extraction stage. Figure 4.6 shows a process of generating frames from the character image using the sliding-window technique.

In the training and evaluation of HMMs, the additional implementations to create the feature vector sequences from the original images are performed, and then the HTK toolkit version 3.4, a toolkit primarily used for speech recognition (Young *et al.*, 2009), is used.

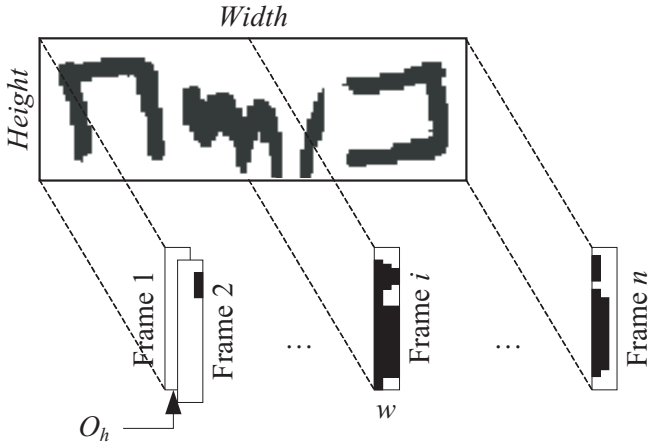


FIGURE 4.6: The sliding-window technique used to generate a sequence of frames from the character image.

Recognition rate (%)		
ThaiCAM	**NECTEC-ON**	**NECTEC-OFF**
83.4	70.2	76.7

TABLE 4.3: Recognition results of nearest neighbour algorithm with exclusive-OR distance on the testing set for three different databases.

4.5 Recognition Performance of Three Algorithms

The recognition results for the nearest neighbour algorithm with exclusive-OR distance on the testing set for three different databases are presented in Table 4.3. It performed best on the ThaiCAM database and worst on the NECTEC-ON database. However, the results on the handwritten databases, in comparison with the results on the printed database, are lower due to a wide variety of handwritten styles. As previously stated, the recognition results of the nearest neighbour algorithm will be a baseline for the next evaluation.

An evaluation of the back-propagation neural network was performed using NECTEC features, as stated in Section 4.4.2. Images were converted into an 8×8 matrix, with the aspect ratio between width and height of an original character as an additional input to the neural network. In the evaluation, the number of hidden nodes was varied, to find the optimal model configuration. The networks were trained by the back-propagation learning algorithm until the mean squared error (MSE) converged to less than 0.01, which corresponds to a correct classification rate of 99%, or until the learning process executed 10,000 epochs. The average recognition accuracy for both training and testing sets was measured. Typical recognition rates for the training set and MSE on ThaiCAM are shown in Table 4.4. In the case of 16 and 32 hidden nodes, the training results show that the MSE cannot converge to less than 0.01. That indicates that a small number of hidden nodes are inadequate to handle the problem. This leads to a low recognition rate

Hidden nodes	Training epochs	MSE	Recognition rate (%)
16	10,000	0.182098	87.9
32	10,000	0.0281332	97.2
64	1,050	0.00985704	99.0
128	340	0.00981138	99.1
256	240	0.00943488	99.2
512	1,680	0.00588377	99.5

TABLE 4.4: An example of training epochs, MSE and recognition rates of the back-propagation neural network on the training set for ThaiCAM database and various number of hidden nodes.

Hidden nodes	Recognition rate (%)		
	ThaiCAM	NECTEC-ON	NECTEC-OFF
16	64.0	69.4	58.1
32	73.5	74.2	68.8
64	80.0	79.7	70.7
128	83.6	83.1	74.3
256	84.6	84.4	76.6
512	84.9	84.8	77.1

TABLE 4.5: Recognition results for the back-propagation neural network on the testing set for three different databases and various number of hidden nodes.

in the testing set as well. The recognition results for the testing sets for three different databases and a various number of hidden nodes are illustrated in Table 4.5.

An increase in recognition rates is obtained with respect to an increase in the number of hidden nodes. It is found that an increase in the number of hidden nodes gives rise to a faster convergence in the training procedure when the network has less than 256 hidden nodes. All three databases show improvement in recognition accuracy in the same manner. The results show that the back-propagation neural network with 512 hidden nodes is the most effective, yielding 86.9, 86.3 and 77.1% accuracy on ThaiCAM, NECTEC-ON and NECTEC-OFF, respectively. This indicates that the network with 512 hidden nodes seems to be the best network for the NECTEC feature and serves as a good representation of our databases.

A series of evaluation of recognition performance of HMMs was conducted. In the evaluation, the sample data was converted into a sequence of 189 feature vectors as input to the HMMs (as stated previously in Section 4.4.3). The number of HMM states was varied to find the optimal model configuration. The optimisation of the HMMs is carried out by maximum likelihood (ML) training using the Baum-Welch algorithm. Training is executed for 100 iterations. The average recognition accuracy of both training

States	**Recognition rate (%)**
40	88.7
50	90.2
60	90.4
70	91.1
80	91.1
90	91.1

TABLE 4.6: Recognition results of hidden Markov models on the ThaiCAM database for various number of HMM states.

and testing sets is measured. Table 4.6 summarises the recognition performance of the testing sets for various numbers of HMM states on the ThaiCAM database.

Among the various numbers of HMM states, the results show that an improvement in recognition rate is achieved with respect to an improvement in the number of HMM states, when the HMMs have less than 70 states. However, there is no significant difference in recognition accuracy when the number of states exceeds 70. This demonstrates that the model with 70 states tentatively provides best recognition performance and serves as a good representation of the ThaiCAM database, which appears as a sequence of 189 frames. Having stated that the HMM with 70 states yields the best results, the evaluation of two different databases using the models trained on ThaiCAM database reported in the following are based on these models. The results show that a recognition rate of 88.8 and 80.4% on unseen samples are obtained on NECTEC-ON and NECTEC-OFF databases, respectively.

As the results show (Tables 4.3, 4.5 and 4.6), the back-propagation neural network and hidden Markov model are more effective than the nearest neighbour classifier and give higher recognition rates. Among the three different classifiers, HMMs with 70 states, significantly outperform the others, while the best recognition accuracy (91.1%) is on the unseen samples of the ThaiCAM database. On the other databases, NECTEC-ON and NECTEC-OFF, the recognition rates for the HMMs with 70 states are also performed in the same manner. A comparison of the three classifiers for three different databases is illustrated in Figure 4.7.

4.6 Summary and Discussion

This chapter provides details of the databases, ThaiCAM, NECTEC-ON and NECTEC-OFF, and the training methods used in this research. Features and classifiers are presented. The basic feature of 64×64 bi-level images (black and white) and nearest neighbour classifiers with exclusive-OR distance that achieves very good performance on the NECTEC printed Thai and English database (as mentioned in Chapter 3) are applied

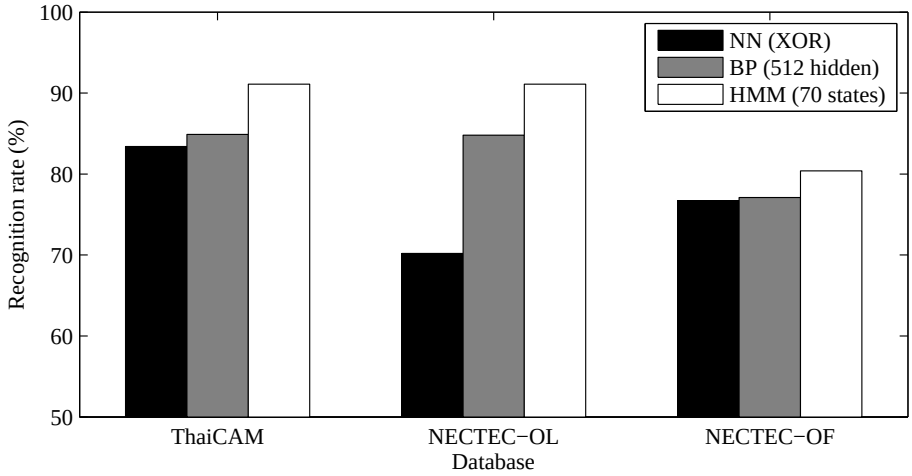


FIGURE 4.7: The best recognition results of three different classifiers on three different databases, summarised from Table 4.3, 4.5 and 4.6.

to the handwritten databases as a baseline. The results show that when applied to the handwritten database, the nearest neighbour classifier was unable to achieve a good performance. The back-propagation neural network and hidden Markov models are more effective and have higher recognition rates. The back-propagation neural network with an 8×8 matrix feature yielded slightly better results in recognition rate for ThaiCAM and NECTEC-OFF, but fell sharply for NECTEC-ON. The HMMs yielded the best results in recognition rate for all databases.

The evaluations on a varied number of hidden nodes for back-propagation neural networks show that an increase in recognition rates is obtained with respect to an increase in the number of hidden nodes. The results illustrate that the back-propagation neural network with 512 hidden nodes has the best accuracy on our datasets. In the same manner as the HMMs, an improvement in recognition rates is obtained with respect to an improvement in the number of HMM states. For the evaluation, the best number of states giving a good representation of the ThaiCAM database is 70.

The evaluation in this chapter was done independently on three different databases. In the next chapter, the recognition performance of three classifiers on cross databases is evaluated and discussed. An adjustment to stroke width in the databases will be performed. A combination of three different databases for training the classifiers will be examined.

Chapter 5

Improving Generalisation for Handwritten Thai OCR

The main objective of this chapter is to evaluate the classifiers on cross databases. Section 5.2 shows and discusses the recognition results of three different classifiers on cross databases. Section 5.3 studies the difference between pseudo handwritten data[1] (NECTEC-ON) and real handwritten data (ThaiCAM and NECTEC-OF) and makes tests to improve recognition accuracy. An improvement in recognition rates by merging and training with three databases is reported in Section 5.4. Finally, a summary and discussion of the chapter are addressed in Section 5.5.

5.1 Introduction

In the previous chapter, the three different classifiers, using their different feature vectors, achieved a good recognition rate in 'within' database. The evaluation of the performance of three classifiers on cross databases is performed in this chapter. To improve the recognition performance of the classifiers, an investigation of online handwritten data (NECTEC-ON) and offline handwritten data (ThaiCAM and NECTEC-OF) is carried out, especially in the case of the simulation of stroke width. A straightforward method to improve the generalisation of the classifiers by merging and training three databases is performed.

5.2 Cross Database Comparison

The previous evaluations (Section 4.5) were carried out independently on three different databases, which achieved a good classification rate. This section evaluates the

[1] Written strokes were simulated from the temporal information generated by an electronic tablet.

(a) Nearest neighbour algorithm using exclusive-OR distance.

Training set	Recognition rates (%)		
	ThaiCAM	**NECTEC-ON**	**NECTEC-OFF**
ThaiCAM	**86.2**	44.9	54.5
NECTEC-ON	70.0	**72.6**	54.0
NECTEC-OFF	65.0	28.5	**76.8**

(b) Back-propagation neural network with 512 hidden nodes.

Training set	Recognition rates (%)		
	ThaiCAM	**NECTEC-ON**	**NECTEC-OFF**
ThaiCAM	**87.1**	58.3	57.4
NECTEC-ON	77.6	**88.2**	56.6
NECTEC-OFF	69.6	42.1	**76.3**

(c) Hidden Markov models with 70 states.

Training set	Recognition rates (%)		
	ThaiCAM	**NECTEC-ON**	**NECTEC-OFF**
ThaiCAM	**90.7**	74.3	71.3
NECTEC-ON	82.1	**88.9**	60.9
NECTEC-OFF	80.3	55.4	**79.9**

TABLE 5.1: Recognition results of three different classifiers on cross databases.

recognition performance of the three classifiers on cross databases. As stated previously (Section 4.2), the three databases are different in many aspects. For example, ThaiCAM was collected from a relatively small number of writers, but who wrote many samples per character, while NECTEC-OFF was gathered from a large number of volunteers who wrote only two samples per character. Moreover the digitisation is different in resolution (DPI). NECTEC-ON is the online handwritten database. The character image is simulated from temporal information.

In this section, An evaluation of three classifiers when tested on the other databases was performed. Note that we used only the middle level of Thai characters for evaluation. The nearest neighbour algorithm using exclusive-OR distance, back-propagation neural network with 512 hidden nodes and HMMs with 70 states were used as well. The recognition results for the three different classifiers are summarised in Table 5.1.

The results in the diagonal in these tables are trained and evaluated on the same database. All results from the three classifiers show that the recognition accuracy of testing on different databases has dramatically decreased, especially when trained on the offline database and tested on the online database; it seems because of the pseudo data on the online database. The investigation of pseudo and real handwritten data will be discussed in the next section.

5.3 Pseudo and Real Handwritten Data

As described in Section 4.2 and Table 4.1, the databases are different in many aspects, such as collection methods. ThaiCAM and NECTEC-OFF are offline handwritten databases. They were collected from participants, who wrote down the samples on a prepared form. It was then digitised and saved as an image. On the other hand, NECTEC-ON is an online handwritten database. The 'offline' images were simulated from temporal information generated by a tablet. Hence, the stroke width of characters is different. For an offline database, the stroke width varies according to the writing implement, while it is arbitrarily fixed for the online database. In the case of NECTEC-ON, the width is fixed at three pixels. Figure 5.1 shows examples of the mean character images from three databases (only the middle level of Thai characters). These images were generated from a size-normalised (64×64 binary) image. We can see that the stroke width in NECTEC-ON is thinner compared to that of ThaiCAM and NECTEC-OFF.

The results from Table 5.1 show that going online for training and offline for testing, and vice versa yields a relatively low recognition rate. As seen in Figure 5.1, the stroke width in NECTEC-ON seems thinner than the others. Hence, a regenerated image of NECTEC-ON by increasing the stroke width will be created to compare with the original. It could be recreated from the temporal information, however, the following evaluation applied a basic image processing operation called erosion and dilation to adjust the stroke width rather than recreating from the temporal information. An adjusted image is quite similar to a regenerated image, but easy and fast to implement. The erosion of the set A by set B is denoted by $A \ominus B$ and is defined by

$$A \ominus B \quad = \quad \{x : B_x \subset A\}, \tag{5.1}$$

where x denotes a point in set A or B, and $\subset$ denotes the subset relation. A and B are called the input image and the structuring element (template). In this work, the structuring element B is represented by a 3×3 diamond matrix. $A \ominus B$ consists of all points x for which the translation of B by x fits inside of A. Figure 5.2 shows an example of a character before and after using the erosion operation. The dilation of set A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B \quad = \quad (A^c \ominus \breve{B})^c, \tag{5.2}$$

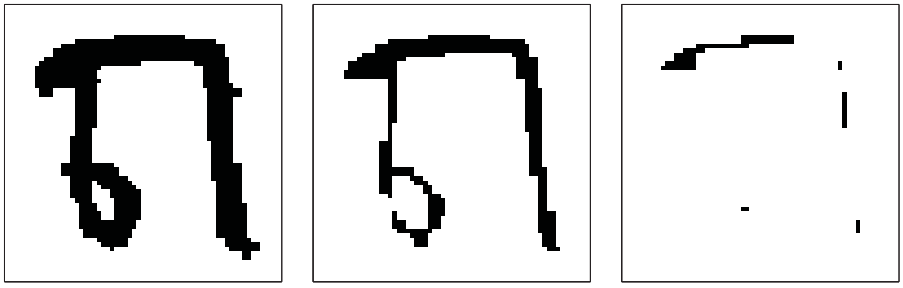
where A^c denotes the complement of A. To dilate A by B, B is rotated around the origin to obtain $\breve{B}$, A^c is eroded by B, and then the complement of the erosion is taken. Details can be found in many image processing books, such as Dougherty and Lotufo (2003). Figure 5.3 shows an example of a character before and after using the dilation operation.

(a)

(b)

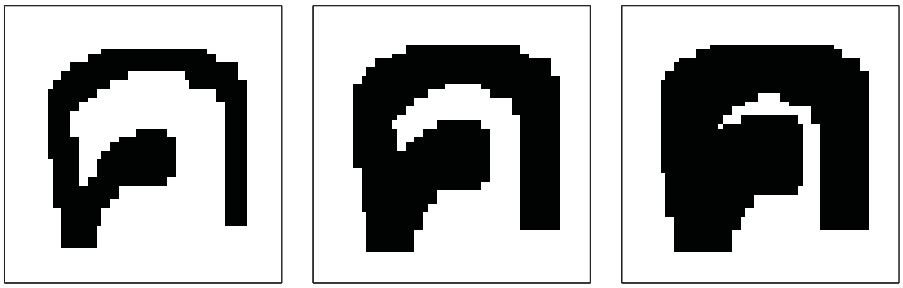
(c)

FIGURE 5.1: Examples of the mean character images of the databases. They are created from the 64×64 binary images. Only the middle level of Thai characters is shown. (a) ThaiCAM, (b) NECTEC-ON and (c) NECTEC-OFF.



(a) (b) (c)

FIGURE 5.2: Example of 64×64 binary image from ThaiCAM when the erosion operation was applied: (a) original, (b) ThaiCAM (Erosion), (c) ThaiCAM (Erosion×2).



(a) (b) (c)

FIGURE 5.3: Effect of dilation on a 64×64 binary image from NECTEC-ON that is misclassified when applied dilation operation two times. (a) original, (b) NECTEC-ON (Dilation), (c) NECTEC-ON (Dilation×2)

The back-propagation neural network (512 hidden nodes) and HMMs (70 states) were used in the evaluation. We used the middle level of Thai characters to evaluate these classifiers. Two evaluations were performed to improve the recognition performance of the classifiers. First, the dilation operation was applied to the NECTEC-ON database to increase the stroke width of a character image. This new NECTEC-ON database was used to evaluate the classifiers. Second, the erosion operation was used on the ThaiCAM database to decrease the stroke width of a character image and to evaluate the classifiers.

The correct recognition rates of the back-propagation neural network and HMMs which were tested on NECTEC-ON and NECTEC-ON dilation datasets are shown in Table 5.2. The NECTEC-ON (Dilation) and NECTEC-ON (Dilation×2) are the NECTEC-ON applied to the binary dilation operation one and two times respectively. The results show that using the dilation operation to increase the stroke width of the character image can improve the recognition rates. However, when applying the dilation operation two times, the recognition rates fell somewhat. Figure 5.4 shows examples of the main character images after applying the dilation operation. We can see that the stroke width in NECTEC-ON (Dilation) is quite similar in thickness to ThaiCAM and NECTEC-OFF, while the NECTEC-ON (Dilation×2) is too thick. This results in the lack of some significant features, as seen in Figure 5.3. The ambiguous character image in Figure 5.3(c) can be ค (/k^h/) or ด (/d/).

(a)

Testing Database	Recognition rate on testing set (%)	
	Neural Network (512 hidden)	**HMMs (70 states)**
NECTEC-ON	58.3	74.3
NECTEC-ON (Dilation)	63.9	79.4
NECTEC-ON (Dilation×2)	58.8	73.8

(b)

Testing Database	Recognition rate on testing set (%)	
	Neural Network (512 hidden)	**HMMs (70 states)**
NECTEC-ON	42.1	55.4
NECTEC-ON (Dilation)	45.2	67.4
NECTEC-ON (Dilation×2)	41.2	65.8

TABLE 5.2: Recognition rates of back-propagation neural network and HMMs classifiers tested on NECTEC-ON and NECTEC-ON dilation: (a) trained on ThaiCAM training set, (b) trained on NECTEC-OFF training set.

FIGURE 5.4: Examples of the mean character images of NECTEC-ON database after applying dilation operation. They are created from the 64×64 binary images. Only the middle level of Thai characters is shown.

Testing Database	**Recognition rate on testing set (%)**	
	Neural Network (512 hidden)	**HMMs (70 states)**
ThaiCAM	77.6	82.1
ThaiCAM (Erosion)	57.0	74.9
ThaiCAM (Erosion×2)	7.0	7.2

TABLE 5.3: Recognition rates of back-propagation neural network and HMMs classifiers trained on NECTEC-ON training set and tested on ThaiCAM erosion.

Figure 5.3 shows the correct recognition rates of the classifiers that were evaluated with the ThaiCAM and ThaiCAM erosion datasets. ThaiCAM (Erosion) and ThaiCAM (Erosion×2) are ThaiCAM with the binary erosion operation applied one or two times, respectively. The erosion operation has a disastrous effect on recognition performance. The reducing of stroke width results in a lack of many significant features as well. Figure 5.2 shows a disastrous effect of erosion operation.

5.4 Combination of Three Databases for Training

In this section, the training sets of three databases were merged together and used to train the classifiers. There are 13,878 tokens for training and 10,862 tokens for testing. The aim of the evaluation was to compare recognition results between training with a single database and training with a combination. The back-propagation neural network (512 hidden nodes) and HMMs (70 states) were used in the evaluation. The settings of the training parameters were similar to the previous evaluation, but only the middle level of Thai characters was evaluated.

Table 5.4 presents the average classification rate of the classifiers. As expected, the classifiers trained with the combined database achieved a recognition performance better than the classifiers trained with a single database. On average, the results showed that an improvement of 18 percentage points in recognition rate was obtained for the back-propagation neural network. In the case of HMMs, an average improvement of 7.6 percentage points over the single database training was obtained. We infer from these results that a generalisation for handwritten Thai character recognition can be improved by training with various written styles.

As in the previous evaluation (Section 5.3), using a dilation operation to increase stroke width in the NECTEC-ON dataset should markedly improve recognition rates. It is possible to use the adjusted NECTEC-ON dataset instead of the original and should give an improvement in the recognition rate. This is suggested for future research.

Training Database	Recognition rate on testing set (%)	
	Neural Network (512 hidden)	**HMMs (70 states)**
Combined database	86.0	83.6
ThaiCAM	67.6	78.8
NECTEC-ON	74.1	77.3
NECTEC-OFF	62.7	71.9
On average	68.0	76.0

TABLE 5.4: Recognition rates of back-propagation neural network and HMMs classifiers trained on the combination of three databases (ThaiCAM, NECTEC-ON and NECTEC-OFF) compared with the average recognition rates of the classifiers trained on each database.

5.5 Summary and Discussion

This chapter aimed to evaluate the classifiers on cross databases. Because of the differences in many aspects of the databases, as stated earlier, the results show that when testing on other databases, the recognition rates drop sharply. All three classifiers display the same behaviour. From our study, there is a big difference in stroke width between the online (NECTEC-ON) and offline (ThaiCAM and NECTEC-OFF) databases. So a regenerated image of the online database was created. An improvement in the classification rates can be obtained by adjusting the stroke width in the NECTEC-ON database. We applied a basic image processing method called dilation operation to increase the stroke width. An increase of approximately 5.6 percentage points in the recognition rate was obtained from the back-propagation neural network and HMMs trained on ThaiCAM and evaluated with NECTEC-ON (Dilation). Moreover, an improvement is noticeable in Table 5.1(b). An approximate 12 percentage points was yielded when using HMMs trained on NECTEC-OFF and evaluated with NECTEC-ON (Dilation). We still obtain an improvement by a combination of three databases into a training set. An average of 7.6 percentage points was obtained over single database training using HMMs, while we obtained an increase of 18 percentage points in the recognition rate for the back-propagation neural network.

In the next chapter, another method to improve the generalisation ability, the boosting algorithm, will be tested. A successful boosting algorithm in many research domains, such as face detection and character recognition, is called AdaBoost. This algorithm will be introduced, followed by an evaluation on our handwritten databases.

Chapter 6

Can AdaBoost Improve Generalisation for Thai OCR?

In the previous chapter, a significant improvement in recognition rates on cross databases was obtained by adjusting the stroke width in the NECTEC-ON database and by merging three databases into a unified training set. In this chapter, we will describe the application of a machine learning algorithm called AdaBoost to our handwritten problem. It is a method for improving the accuracy of a learning algorithms (Freund and Schapire, 1996, 1997, 1999).

Many problems have been successfully solved using the AdaBoost and its variants (Freund and Schapire, 1996), including character recognition, text filtering, image retrieval and medical diagnosis, etc. AdaBoost has been applied to rather weak learning algorithms (with low capacity), such as decision trees (Freund and Schapire, 1996; Drucker, 1996). Moreover, to the best of our knowledge, it has been applied to strong learning algorithms, such as artificial neural networks (Schwenk and Bengio, 1997a,b, 2000) and hidden Markov models (Gunter and Bunke, 2002), as well. These studies showed rather intriguing generalisation properties, such as a continued decrease in generalisation error after training error reaches zero.

In this chapter, the AdaBoost algorithms, which aim to solve the handwritten Thai problem, are presented. The general concept of AdaBoost is introduced in Section 6.1, followed by a training method for AdaBoost (Section 6.2). The evaluation on our handwritten databases was performed and the results reported in Section 6.3. An investigation of the results was performed using a controllable dataset in Section 6.4, followed by the evaluation on the adjustment of the size of the training set in Section 6.5 and the number of dimensional feature vectors in Section 6.6. Finally, a summary and discussion of the chapter are given in Section 6.7.

6.1 General Concept of AdaBoost

AdaBoost, which stands for Adaptive Boosting, is one of the most influential ensemble methods. It is an instantiation of the general boosting method. It has been applied with great success to several benchmark machine learning problems using mainly decision trees as base classifiers (Freund and Schapire, 1996). For the character recognition problem, it has also been applied with many classifiers, such as the nearest neighbour classifier (Freund and Schapire, 1996). Schwenk and Bengio (2000) used the AdaBoost with the back-propagation neural networks. Gunter and Bunke (2002) applied it with the hidden Markov models. Fu *et al.* (2008) employed it to the descriptive model based multiclass classifiers (Modified Quadratic Discriminant Function, MQDF). AdaBoost was first introduced by Freund and Schapire (1996) and originally designed for binary classification problems, and has been successfully used in face detection (face or non-face). Afterwards, it was broadly applied to other domain research, such as character recognition for multiclass classification cases. AdaBoost and its variants have been applied to diverse domains with great success, owing to its solid theoretical foundation, accurate prediction and great simplicity. Practically, the AdaBoost algorithm has many advantages. It is fast, simple and easy to program. There is only one parameter to choose, the number of iterations (T). It requires no prior knowledge about weak classifiers, and so can be flexibly combined with any method to improve performance. Moreover, the important benefit of AdaBoost is that, instead of trying to design a classification algorithm that is accurate over the entire space, we can instead focus on finding a set of weak classification algorithms that only need to be better than random.

In going from binary to multiclass classification, many extensions have been developed. In general, there are two ways to extend AdaBoost from binary to multiclass classification problems. Most boosting algorithms have been restricted to reducing the multiclass classification problems to multiple binary classification problems (Drucker, 1996; Schwenk and Bengio, 1997a,b; Sun *et al.*, 2007; Jun and Ghosh, 2009). However, some newly developed algorithms directly extend the AdaBoost algorithm to multiclass problems, without reducing it to multiple binary problems (Hao and Luo, 2006; Fu *et al.*, 2008; Zhu *et al.*, 2009). AdaBoost and its extensions have been applied in many papers in many research fields and in many datasets. As seen in Freund and Schapire (1996), Hao and Luo (2006), Sun *et al.* (2007), Jun and Ghosh (2009) and Zhu *et al.* (2009), the results of evaluation show that AdaBoost gives significant performance improvement.

The following sections describe the AdaBoost algorithm for two-class and for multiclass problems. In Section 6.1.1, the classical AdaBoost is introduced, followed by the multiclass AdaBoost in Section 6.1.2.

6.1.1 Classical AdaBoost

Classical AdaBoost is originally designed for binary classification problems. It constructs a composite classifier by sequentially training classifiers, while putting more and more emphasis on certain patterns. It normally has been applied to low capacity classifiers (called 'weak classifier') for creating a set of high capacity classifiers (called 'strong classifier'). AdaBoost only requires the performance of each weak classifier to be better than random guessing. The basic concept is very simple. In each round of iteration, it increases misclassified samples' weights and reduces right classified samples' weights so that the subsequent weak classifier could place more emphasis on those misclassified samples. The classical AdaBoost algorithm is presented in Algorithm 1 (Freund and Schapire, 1996). The algorithm takes as input a training sample set $\{(x_i, y_i), i = 1, \ldots, N\}$ where x_i belongs to a feature vector, y_i is the label of x_i and N is the number of training samples respectively. In the binary classification case, $y_i \in Y = \{-1, +1\}$. The output from the classifier (h) on training example i is denoted $h(x_i), h \in \{-1, +1\}$. T is the number of iterations and t is the iteration index. D_t^i is the weight of iteration t for sample i. The final output is denoted H.

Algorithm 1 Classical AdaBoost Algorithm. (Freund and Schapire, 1996)

1: {Training procedure}
2: $D_1^i = \frac{1}{N}$ for all i
3: **for** $t = 1$ to T **do**
4: Train a classifier using distribution (D_t).
5: $\epsilon_t = \sum_{i=1}^{N} D_t^i$ where $h_t(x_i) \neq y_i$
6: $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
7: $z_t = \sum_{i=1}^{N} D_{t+1}^i$
8: **if** $h_t(x_i) = y_i$ **then**
9: $D_{t+1}^i = \frac{D_t^i . e^{-\alpha_t}}{z_t}$
10: **else**
11: $D_{t+1}^i = \frac{D_t^i . e^{\alpha_t}}{z_t}$
12: **end if**
13: **end for**
14: {Testing procedure}
15: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . h_t(x)\right)$

Now we will explain the algorithm in detail. Initially, all weights of the first distribution (D_1) are set equally, as shown in Line 2. Next, the weighted value of misclassified samples for each iteration (ϵ_t) is summarised (Line 5). Then, in Line 6, the generalised confidence (α_t) is calculated. Next, the normalisation factor is calculated (Line 7). Then, the new weight (D_{t+1}) is updated. The weight of correctly classified

examples ($h_t(x_i) = y_i$) are decreased (Line 9), while the weight of incorrectly classified examples ($h_t(x_i) \neq y_i$) are increased (Line 11) so that the classifier is forced to focus on the hard examples in the training set in the next iteration. Finally, the procedure is repeated T iterations to compute a series of generalised confidence (α).

After training T iterations, AdaBoost generates a sequence of hypotheses (h_t) and combines them with confident weights (α_t) to evaluate the performance of the training data. For testing an unknown sample, the generalised confidence (α) is summarised (Line 15) and the sign ($+$ or $-$) is identified the class to which it belong.

6.1.2 Multiclass AdaBoost

As mentioned in Section 6.1.1, classical AdaBoost is not designed for multiclass classification problems. Many real world learning problems, however, are multiclass with more than two possible classes. To handle them, many AdaBoost extensions have been introduced. AdaBoost.M1 is the most straightforward generalisation of AdaBoost algorithm. It directly extends the classical AdaBoost algorithm to the multiclass case without reducing it to multiple two-class problems. Detail of the algorithm is given in Algorithm 2.

Algorithm 2 Multiclass AdaBoost.M1 Algorithm. (Freund and Schapire, 1999)

1: {Training procedure}
2: $D_1^i = \frac{1}{N}$
3: **for** $t = 1$ to T **do**
4: Train a classifier using distribution (D_t).
5: $\epsilon_t = \sum_{i=1}^{N} D_t^i$ where $h_t(x_i) \neq c_i$
6: $\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
7: $z_t = \sum_{i=1}^{N} D_{t+1}^i$
8: **if** $h_t(x_i) = c_i$ **then**
9: $D_{t+1}^i = \frac{D_t^i . \alpha_t}{z_t}$
10: **else**
11: $D_{t+1}^i = \frac{D_t^i}{z_t}$
12: **end if**
13: **end for**
14: {Testing procedure}
15: $H(x) = \arg\max_{c \in C} \sum_{t=1}^{T} \alpha_t . \delta(h_t(x) = c)$ where $\delta(h_t(x) = c) = \begin{cases} 1 & \text{if } h_t(x) = c \\ 0 & \text{otherwise} \end{cases}$

In the multiclass classification problem, $c_i \in C = \{1, \ldots, K\}$ where K is the number of classes. The condition for updating the sample weight is similar to the classical AdaBoost. It increases misclassified weights and reduces correct classified sample

weights so that the subsequent iterations could give more emphasis on those misclassified samples. For testing an unknown sample, the equation in Line 15 is applied. The answer is assigned to the class represented by the index of maximum generalised confidence (α).

However, AdaBoost.M1 has its limitations. AdaBoost.M1 is adequate when the weak classifier is strong enough to achieve reasonably high accuracy, even on the hard distributions created by AdaBoost. However, this method fails if the weak classifier cannot achieve at least 50% accuracy when run on these hard distributions (Freund and Schapire, 1999).

The second version of AdaBoost aims to improve the recognition performance by extending the communication between the boosting algorithm and the classifier. The boosting algorithm AdaBoost.M2 is based on this idea (Freund and Schapire, 1999). Detail of the algorithm is shown in Algorithm 3.

Algorithm 3 Multiclass AdaBoost.M2 Algorithm. (Freund and Schapire, 1999)

1: {Training procedure}
2: Let $B = \{(i, c) : i \in \{1, \ldots, N\}, c \neq c_i\}$
3: $D_1^{(i,c)} = \frac{1}{|B|}$ for $(i, c) \in B$
4: **for** $t = 1$ to T **do**
5: Train a classifier using mislabel distribution (D_t).
6: $\epsilon_t = \frac{1}{2} \sum_{(i,c)\in B} D_t^{(i,c)}.[1 - h_t(x_i, c_i) + h_t(x_i, c)]$
7: $\alpha_t = \frac{\epsilon_t}{1-\epsilon_t}$
8: $D_{t+1}^{(i,c)} = \frac{D_t^{(i,c)}.\alpha_t^{\frac{1}{2}.(1+h_t(x_i,c_i)-h_t(x_i,c))}}{z_t}$
9: **end for**
10: {Testing procedure}
11: $H(x) = \arg\max_{c\in C} \sum_{t=1}^{T} (\log \frac{1}{\alpha_t}).h_t^{(x,c)}$

From the algorithm, a mislabel is a pair (i, c) where i is the index of a training sample and c is an incorrect label associated with sample i. Let B be the set of all mislabels as shown in Line 2. A mislabel distribution is a distribution defined over set B of all mislabels. To apply this algorithm a classifier needs to be able to generate more expressive hypotheses rather than identifying a single label, instead of choosing a set of 'plausible'[1] labels. This may often be easier than choosing just one label. For instance, in Thai OCR, it may be hard to tell if a particular image is a ก or a ถ, but it is easy to eliminate all of the other possibilities. In this case, rather than choosing between ก and ถ, the hypothesis may output the set {ก, ถ} indicating that both labels are plausible. When the classifier indicates a 'degree of plausibility', each hypothesis gives a set of vector outputs $[0, 1]^k$, where the components with values close to 1 or 0 correspond to those labels considered to be plausible or implausible, respectively. When the classifier has

[1]The authors use the term 'plausible' rather than 'probable' to emphasise that these numbers should not be interpreted as the probability of a given label which needs to sum to one.

more expressive power, it also places a more complex requirement on the performance of the classifier. Rather than using the usual prediction error, the classifier does well with respect to a more sophisticated error measure called 'pseudo-loss' (as shown in Line 6 of Algorithm 3). Unlike ordinary error, which is computed with respect to a distribution over examples, pseudo-loss is computed with respect to a distribution over the set of all pairs of examples and incorrect labels. By manipulating this distribution, the boosting algorithm can focus the weak learner not only on hard-to-classify examples, but more specifically, on the incorrect labels that are hardest to discriminate.

Several more sophisticated methods have been developed, for example, AdaBoost.MH, AdaBoost.OC and AdaBoost.ECC (Fu *et al.*, 2008). However, the main procedure is still same as classical AdaBoost as shown in Algorithm 1, only some equations are changed. In this study, we will use AdaBoost.M1 and AdaBoost.M2 due to their simplicity.

6.2 Training Methods for the AdaBoost Algorithm

As seen in Algorithms 1, 2 and 3, each iteration (T) of the AdaBoost requires the distribution (D_t) for training a classifier. There are generally two methods for training a classifier using distribution (D_t) (Schwenk and Bengio, 2000). Some classifiers can use the weights directly to adjust the parameters of the classifier, such as neural networks and hidden Markov Models. Other classifiers, however, cannot make direct use of the weights, such as the nearest neighbour algorithm. In the latter case, a resampling technique can be applied. We can generate a new set of training samples distributed according to weights. The samples with larger weight values have more chance of being chosen in the new set, even multiple times, while the less favoured ones might get lost.

The resampling technique is rather simple and can apply to all of our three classifiers. Moreover, the neural networks (SNNS version 4.1) and hidden Markov models (HTK toolkit version 3.4) packaging seems unable to be customised for applying the weights directly. Hence the following evaluation will be based on this technique. The classifier is trained with a fixed training set obtained by resampling from the original training set. N samples from the original training set were selected, each time with a probability ($P_t(i)$) of picking sample i. After training a classifier with the new training set based on the weight distribution (D_t), a new weight distribution (D_{t+1}) is recalculated based on the results from the validation set. Next, a new training set is resampled using the new weight distribution (D_{t+1}) and then used to train the classifier and so on (Schwenk and Bengio, 2000).

From Table 4.2, the original training set was resampled to a new fixed training set based on weight distribution (D_t). 75% of the new fixed training set was obtained from the original training set. It applied to training the classifier. Validation is applied to the

whole training set (100%), while the whole testing set is used to evaluate the recognition performance of the classifiers.

6.3 Results for Thai OCR Using AdaBoost

The following sections present the AdaBoost results using the training method described in Section 6.2. First, the results for each database will be indicated and discussed, followed by the results on cross databases. In the evaluation the number of iterations (T) for training the AdaBoost algorithm is fixed at 50. To consider the different implementations of the AdaBoost algorithms, a multiclass AdaBoost.M2 is selected for the evaluation. Detail of the algorithm can be found in Section 6.1.2 (Algorithm 3). We applied it to the back-propagation neural network (512 hidden nodes). There is good reason for doing that, because the back-propagation neural network very easily gives us an output directly applicable to the AdaBoost.M2 algorithm. The score from an output node is always in the range from 0 to 1, the same as the requirement for the AdaBoost.M2 algorithm. For the other two classifiers we cannot obtain an output score in the range 0 to 1 immediately. We need to utilise some techniques (such as maybe range normalisation) to convert it. The settings for training parameters are similar to previous evaluation, but only the middle level of Thai characters is evaluated.

After that, a significance test is performed. In the following evaluation, we used McNemar's test. It is a non-parametric method used to compare two population samples that are related or correlated to each other. This test is also used when we analyse a study where subjects are tested before and after time periods. In this case, a decision is made as to whether to accept or reject the hypothesis that there is no difference between the classifier, with and without AdaBoost.

McNemar's test evaluates changes in related or paired binomial attributes, and whether changes in one direction are significantly greater than those in the opposite direction. To compute McNemar's, the following formula is used:

$$X^2 = \frac{(B-C)^2}{B+C} \tag{6.1}$$

where B denotes the number of correct answers in the classifier without AdaBoost converting to a wrong answer in the classifier when applying AdaBoost and C denotes the number of wrong answers in the classifier without AdaBoost converting to a correct answer in the classifier when applying AdaBoost. The calculated value is then compared to the critical value from the chi-square statistic table with one degree of freedom (df). More details of the significance test, chi-square statistic table and McNemar's test can be found in Chatfield (1995) and Sheskin (2004).

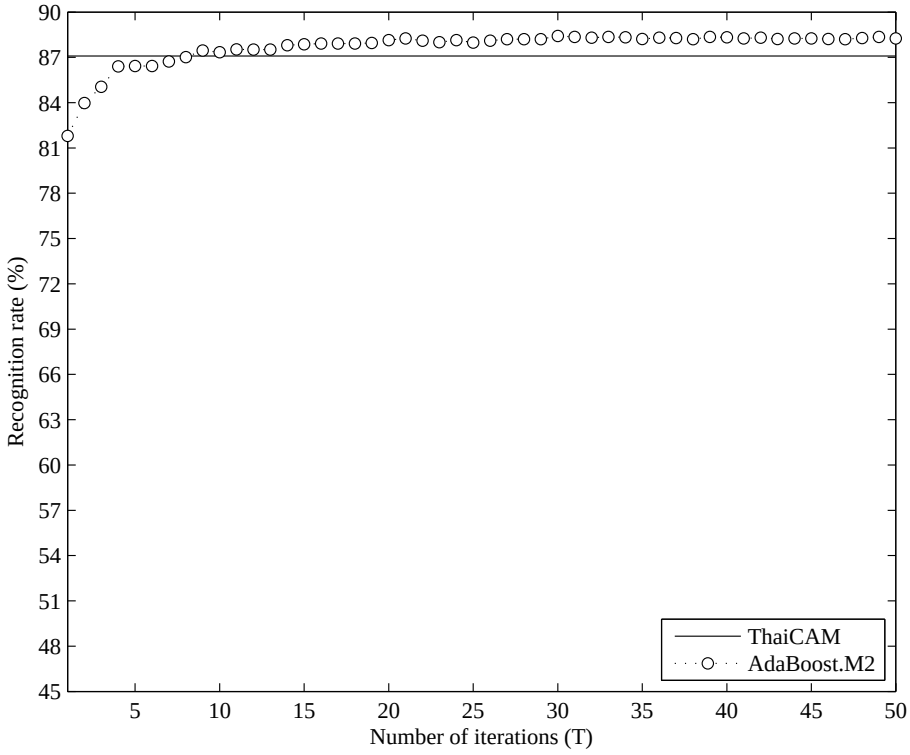


FIGURE 6.1: A comparison of the recognition rate of the neural network and AdaBoost.M2 on ThaiCAM database.

6.3.1 Results within Databases

In Figure 6.1, the reported recognition performance is on ThaiCAM database. From the baseline neural network without applying AdaBoost, AdaBoost slightly improves the recognition rate from 87.1 to 88.3% (the recognition rate at 50 iterations). As shown in the figure, from 1 to 5 iterations, the recognition rates when applying AdaBoost rise rapidly from 81.8 to 86.4%. After the 5th iteration, we obtain similar results to the original for recognition rate. Until around the 10th iteration, the results when applying AdaBoost are slightly better than those without applying AdaBoost. For the significance test, the critical value at the 5% level is 3.84 (from the chi-square table), and because the calculated value of 6.21 exceeds this value, there is a significant difference in the recognition rate between the baseline neural network and the method applying AdaBoost.

6.3.2 Results on Cross-databases

Figure 6.2 presents the average classification rates for the back-propagation neural network trained in ThaiCAM database. The recognition rates in this figure are an average recognition rate of the other two databases. The average recognition rate

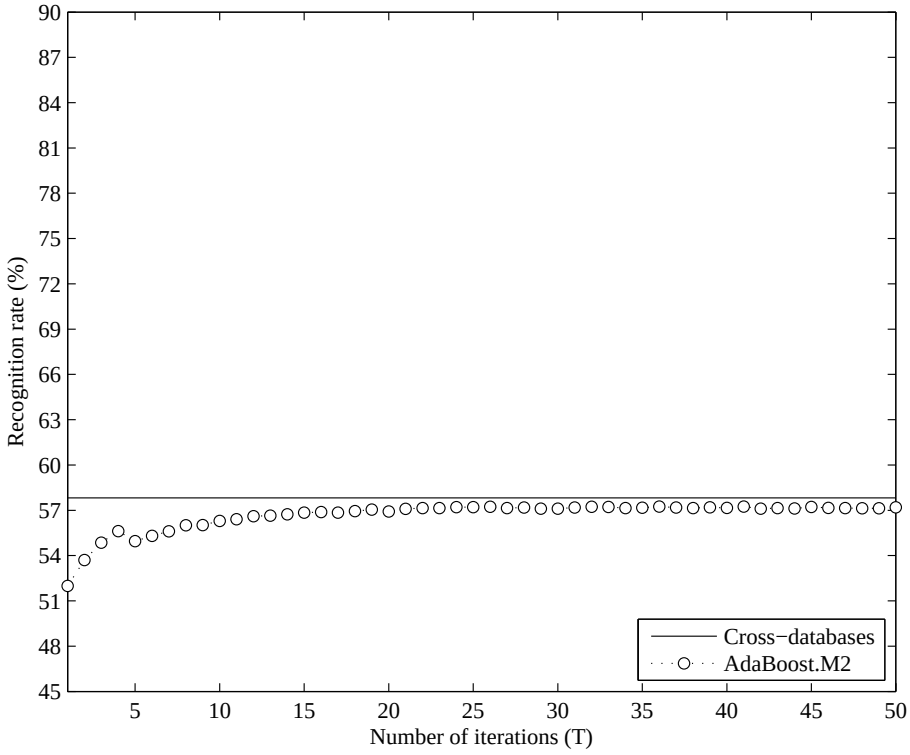


FIGURE 6.2: A comparison of the average recognition rate of the neural network and AdaBoost.M2 on cross-databases (NECTEC-ON and NECTEC-OFF). The classifier trained with ThaiCAM database.

obtained 57.8% without applying AdaBoost and 57.2% when applying AdaBoost at 50 iterations. For training with ThaiCAM, it seems that AdaBoost algorithms cannot improve the average recognition rates for cross-databases. The recognition rates for AdaBoost are always worse that those of the original. However, when using a significance test, the calculated value of 2.42 does not exceeds the critical value (3.84 at the 5% level). This indicates that, although there is a slight difference in the recognition rate between the classifier with and without AdaBoost, it cannot be considered to be statistically significant.

6.3.3 Discussion of Results

Table 6.1 summarises the recognition rates of the back-propagation neural network with and without applying the AdaBoost.M2 algorithm on ThaiCAM database and cross-databases. We obtained a slight improvement (sometimes a small decrease) in recognition rate over the original classifiers (without applying the AdaBoost algorithm). For the 'within database', the AdaBoost algorithm improved the recognition rate by about 1.2 percentage points (Figure 6.1). In the case of the cross-database, the AdaBoost algorithm could not improve the recognition rate over the back-propagation neural

Database	Recognition rate (%)	
	With	Without
ThaiCAM	88.3	87.1
Cross-databases	57.2	57.8

TABLE 6.1: Summary of the recognition rates of the back-propagation neural network with and without applying the AdaBoost.M2 algorithm. The recognition rates of the back-propagation neural network with applying the AdaBoost.M2 were obtained at 50 iterations.

network (Figure 6.2). It gave a lower, but not statistically significant recognition rate, of about 0.6 percentage points. In this case, it is possible that we applied the AdaBoost algorithm with our strong classifiers rather than applying it to a weak classifier. However, from our reviews, some research also applied the AdaBoost to a strong classifier and a slight improvement was obtained as well. For example, Schwenk and Bengio (2000) applied the AdaBoost algorithm to the back-propagation neural networks on an online handwritten digital dataset. An improvement in the recognition rate of approximately 1.2 percentage points was reported. Fu *et al.* (2008) applied the AdaBoost algorithm to the Modified Quadratic Discriminant Function (MQDF) on a Chinese handwritten database. Applying the AdaBoost algorithm improved the recognition rate by about 0.2 percentage points.

To understand the results from our evaluations, a synthesised dataset, which can be used to control a level of data difficulty, will be created and tested in the following section.

6.4 Results on the Synthesised Data

In Section 6.3, an attempt to improve the overall recognition performance among three different databases by the AdaBoost algorithm seemed to be unsuccessful. We obtained a slight improvement (sometimes a small decrease) in the recognition rate over the original classifiers (without applying the AdaBoost algorithm). Hence, the following section will try to understand the results from the previous section. In the previous evaluation, two parameters required to be fixed: the number of iterations and the training samples. The number of iterations was fixed at 50, while the size of the random samples was fixed at 75% of the training set. These two parameters will be evaluated in this section.

To understand the results of the AdaBoost algorithm from the previous section, a controllable dataset is required. Because we are dealing with a multiclass problem and that is easy to understand, a synthesised dataset with only three classes was created, and applied to the nearest neighbour classifier using Euclidean distance. Each sample is represented by a sequence of two-dimensional feature vectors (x, y). The multivariate log-normal pseudo random number generator was used to randomly synthesise 20

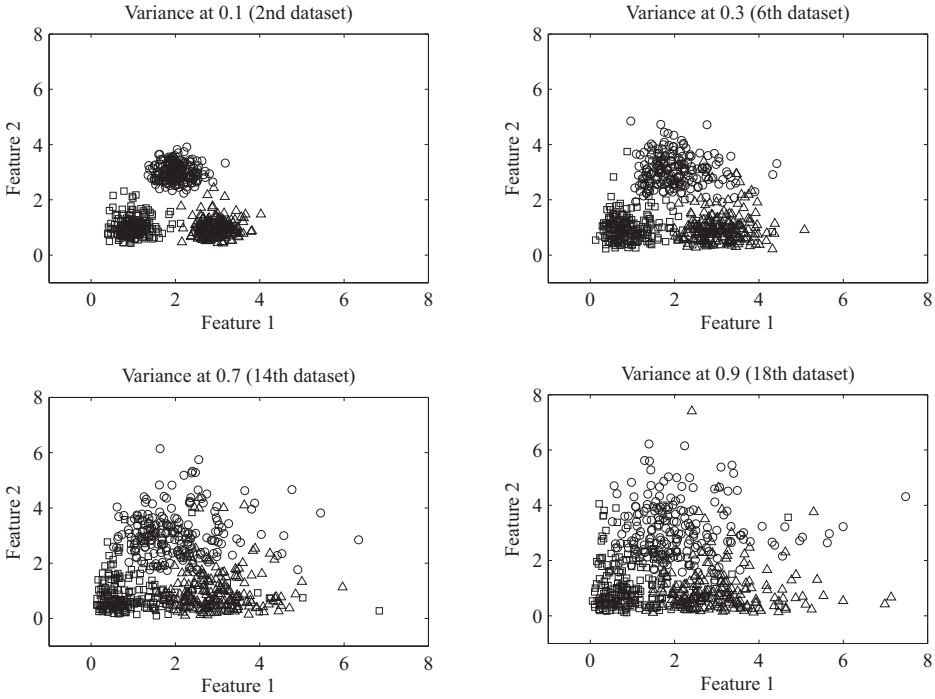


FIGURE 6.3: Examples of the synthesised dataset.

datasets with different variances. Each dataset consists of 10 sub-datasets with same variances.

First, a mean (μ) of each class was defined. In the evaluation, it was fixed at $(1,1),(2,3)$ and $(3,1)$ for the first, second and third class respectively. Other samples of each class are random, using a variance starting from 0.05 for the 1st dataset and then increasing by 0.05 until equal to 1 for the 20th dataset. The small variance means the well-separated data, while the larger variance means the harder data. Examples of the datasets are shown in Figure 6.3. The number of samples per class is 200. They were equally separated into two subsets, training and testing. Hence, there are 300 tokens for training and testing. The results of the following evaluations are repeated 10 times, with each of 10 sub-datasets. The 10 results from the sub-datasets can be averaged to produce and estimate of mean performance and its variability.

Figure 6.4 shows the recognition rates of the nearest neighbour classifier from 0.05 to 1.0 variance with an error bar. In the small variance the classifier yields a perfect recognition rate. This number decreased gradually to 84.6% at the 0.5 variance and continued to decrease to 76.9% at the 1.0 variance. It can be clearly seen that the recognition rates declined with respect to an increase of the variance.

In the first evaluation, a small number of iterations ($T = 20$) for training the AdaBoost algorithm was fixed, to investigate the recognition performance of each dataset. Figure 6.5 shows a comparison of the recognition performance of the nearest neighbour classifier using Euclidean distance, and AdaBoost.M1 on the 2nd, 6th, 14th

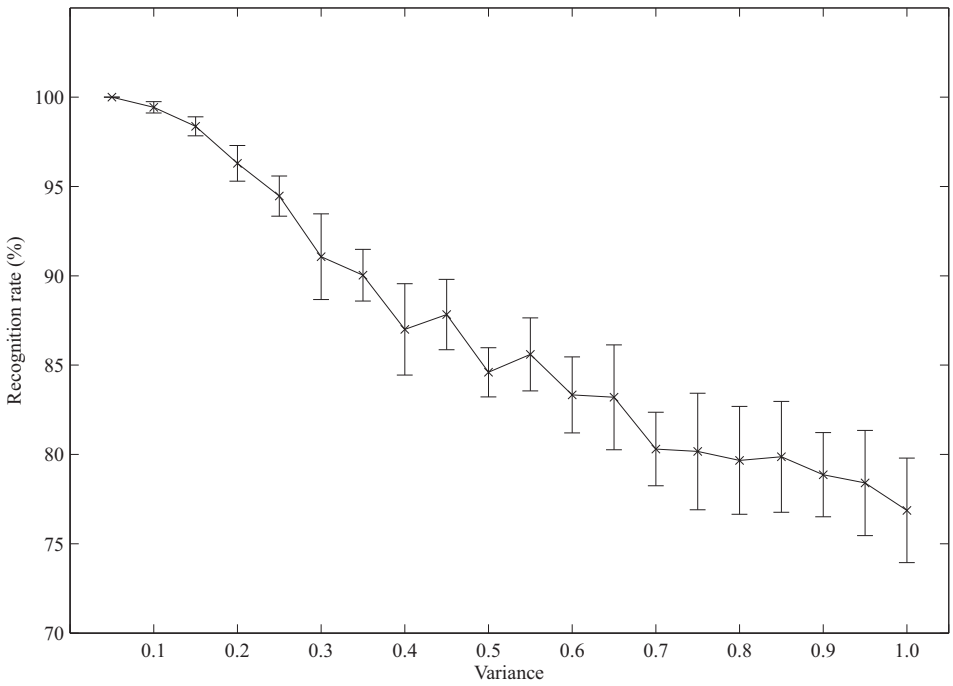


FIGURE 6.4: Recognition rates for the baseline classifier and various number of variances from 0.05 to 1.0 with the error bar.

and 18th datasets. AdaBoost.M1 is an extension of the classical AdaBoost designed for binary classification problems. It is the most straightforward generalisation and can be applied to the nearest neighbour directly without any modification. Detail of the algorithm can be found in Section 6.1.2 (Algorithm 2). For datasets that were well separated, such as the 2nd dataset, both classifiers yielded a perfect recognition rate with a very small error bar. However, in the case of the 6th dataset which contained some ambiguous tokens, AdaBoost seemed not to achieve a better recognition rate than the original classifier. With more ambiguous tokens, as in the 14th and 18th datasets, both achieved a lower recognition rate, and AdaBoost still seemed not to give better accuracy than the original classifier.

Regarding the results in Figure 6.5, it seemed that the AdaBoost algorithm could not improve the performance of the classifier. From our reviews, some papers have applied the AdaBoost algorithm with more than 100 iterations. For example, Hao and Luo (2006) and Athitsos and Sclaroff (2005) used 1,000 and 5,000 boosting iterations in their evaluations. Hence, the next evaluation will increase the number of iterations.

In the following evaluation, a synthesised dataset with 0.5 variance (10th dataset) was used because it is neither hard nor easy for testing. The number of iterations was fixed at 5,000. Figure 6.6 shows the recognition results with the error bar. The recognition rate was improved at the beginning and continually declined. A paired t-test was used to measure the significance of the results. It shows that the difference in recognition rates between the baseline and AdaBoost is statistically significant at the 5% level. In this

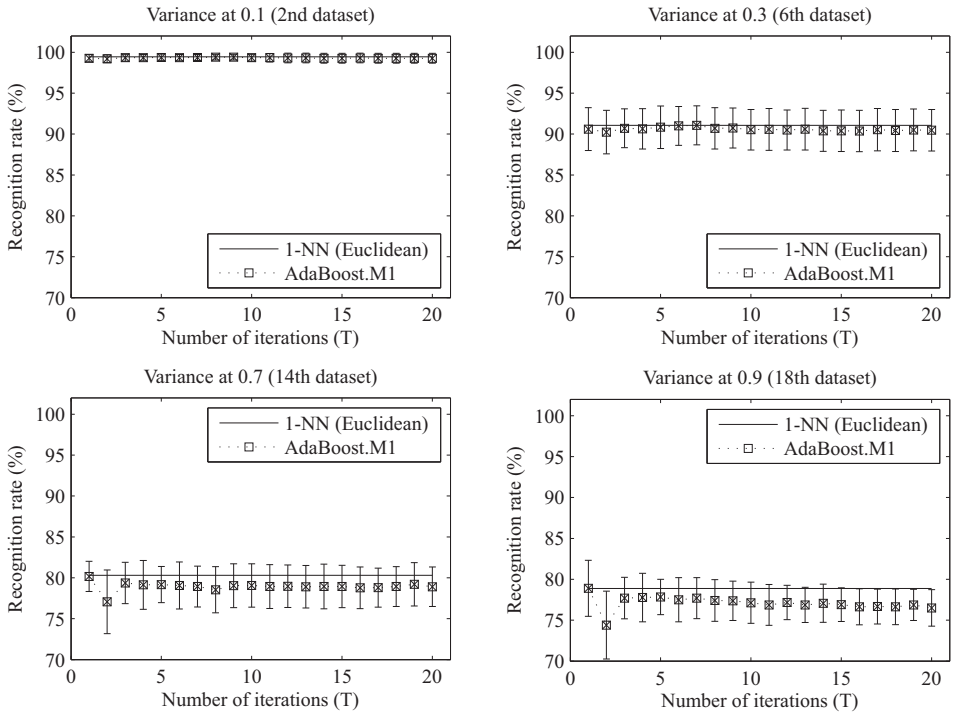


FIGURE 6.5: Comparison of the recognition rates of the nearest neighbour classifier using Euclidean distance and AdaBoost.M1 on some synthesised datasets (2nd, 6th, 14th and 18th). The AdaBoost.M1 is the combination of the nearest neighbour classifiers from the 1st to 20th iteration.

case, increasing the number of iterations seemed to achieve a worse result for recognition rate.

6.5 Adjusting Size of Training Set

In the previous evaluation, only a slight improvement (sometimes a small decrease) in the recognition rate was obtained when applying the AdaBoost algorithm. In our understanding, the AdaBoost algorithm should produce a better improvement in recognition rates rather than a slight improvement. There is implicit evidence in the literature that it may well apply to a weak classifier rather than a strong classifier, as in our evaluation. So in the next evaluation, we attempted to weaken our classifier.

In the training process for AdaBoost, each classifier was trained with a fixed training set obtained by re-sampling with replacement ones from the original training set. In this case, we randomly selected 75% of the training set to be a new training set for the nearest neighbour classifier. Details can be found in Section 6.2.

From Figure 6.7, in the case of using 75% of the training set, when applying more than one nearest neighbour classifier, the recognition rate decreased. It can be increased at the 3rd iteration, but was always at a lower level than the original classifier. It seems to be that the AdaBoost algorithm could not improve any recognition rate. This may

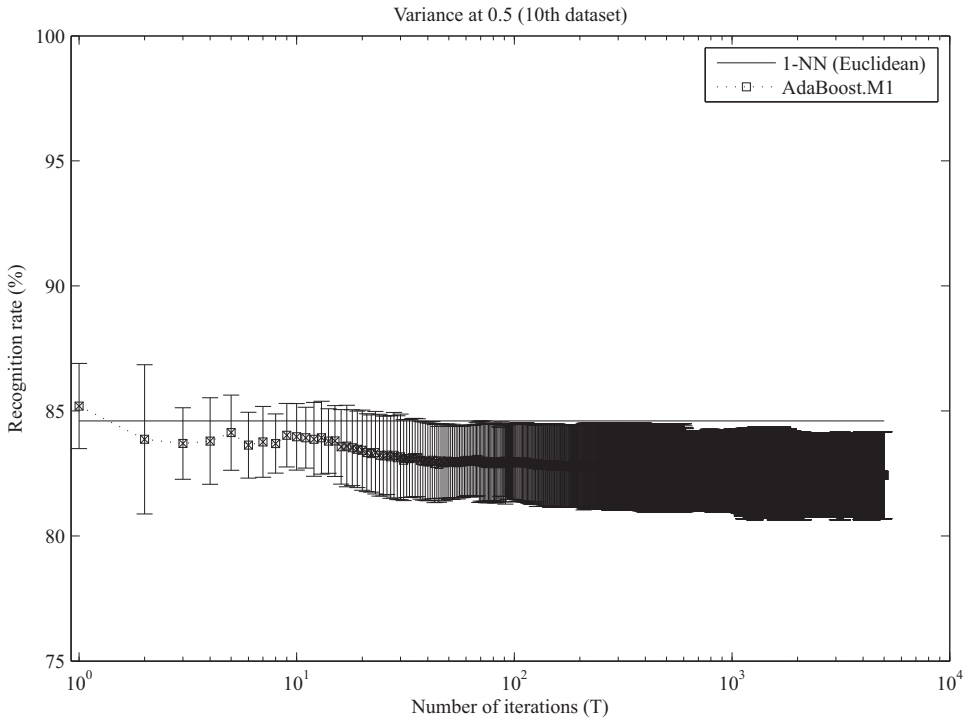


FIGURE 6.6: Comparison of the recognition rates of the nearest neighbour classifier using Euclidean distance and AdaBoost.M1 on the 10th dataset.

be because the nearest neighbour classifier performed a high recognition rate the first time it was run. The result shows that only the nearest neighbour classifier without AdaBoost is adequate for identification. It seems not to be the weak classifier in this case. To confirm this hypothesis, we weakened the classifier by reducing the number of training data in the training set. We reduced the number of training data from 75% to 10% and 5% respectively.

In the case of using the 10% training set, the AdaBoost algorithm achieved a better recognition rate at the beginning. The recognition rate gradually increased and remained stable after the 8th iteration. In the case of using a 5% training set, although the AdaBoost algorithm yielded a worse recognition rate at the beginning, it could achieve a better recognition rate after the 3rd iteration. After that, the recognition rate gradually increased and remained stable at the 8th iteration. The stable results after the 18th iteration are used to measure the significance. The calculated t (6.23, 4.70 and 9.51 for the 75, 10 and 5% training data against the baseline classifier) is greater than the tabulated t ($t_{0.5,\ 9} = 2.26$). Thus, the results are significant at the 5% level.

The recognition performance for AdaBoost.M1 at the 18th iteration and various sizes of the training set from 5 to 95% is illustrated in Figure 6.8. In the 5 and 10% training set the AdaBoost.M1 yields a significant improvement in recognition rates over the baseline classifier. After the 20% training set the recognition performance of AdaBoost.M1 is lower than the baseline classifier.

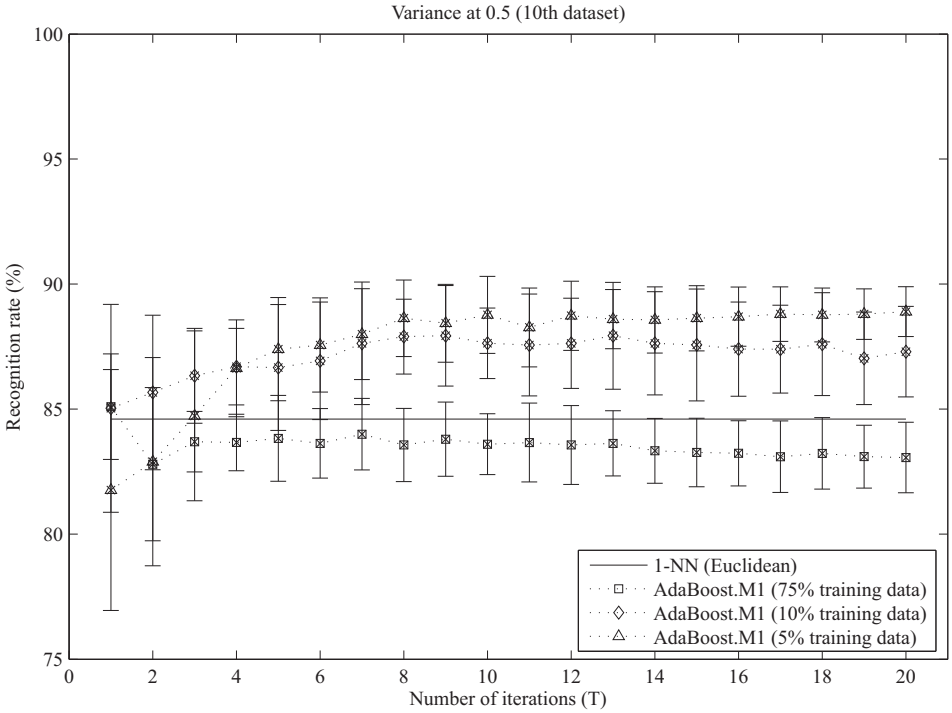


FIGURE 6.7: Comparison of the nearest neighbour classifier and AdaBoost algorithm (M1) using a different size of the training set (75, 10 and 5%).

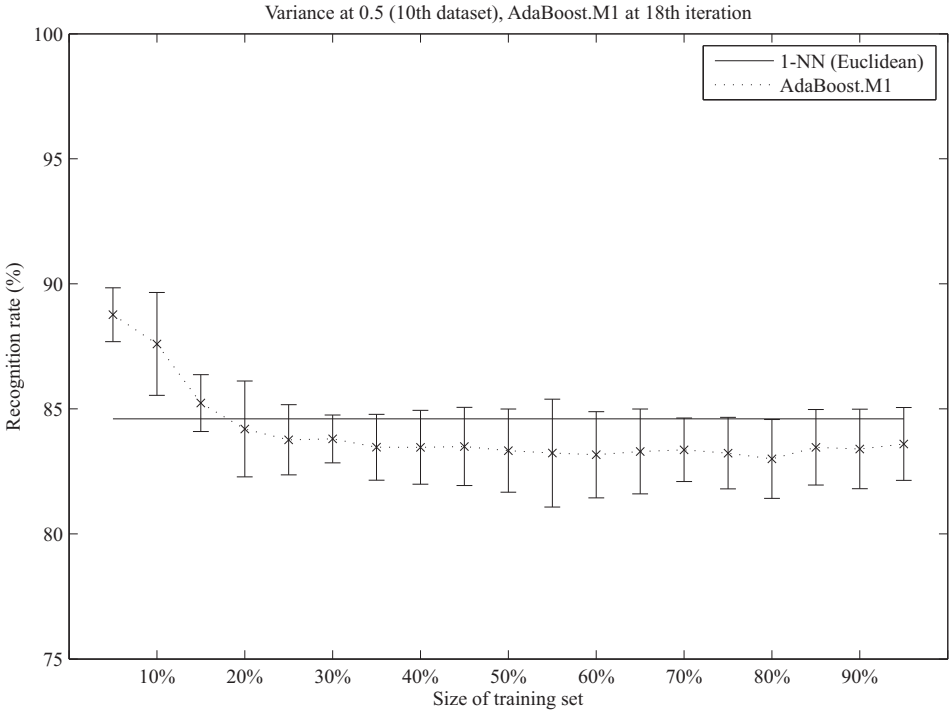


FIGURE 6.8: Comparison of the nearest neighbour classifier and AdaBoost algorithm (M1) at 18th iteration and various sizes of the training set.

The results from this evaluation seem to conclude that the AdaBoost algorithm can perform effectively using a weak classifier. In the case of the classifiers being too strong, the performance of the AdaBoost algorithm may be ineffective. However, the synthesised data differs from the handwritten data in many aspects, such as the number of dimensional feature vectors and variances for the classes. In the next evaluation, increasing the number of dimensional feature vectors will be tested to get an error estimate.

6.6 Increase of Dimensional feature vectors

The results from Section 6.5 seem to conclude that when we weakened the classifier by reducing the number of training data, we can get better recognition rates. The synthesised and handwritten data are rather different in many aspects, such as the number of dimensional feature vectors and variances for the classes. Hence in the following evaluation we will evaluate the various numbers of dimensional feature vectors and discuss the results. The parameters and dataset for training are similar to the previous evaluation.

Figure 6.9 shows the results for various dimensional feature vectors from 2 to 2000. We obtained an improvement in recognition rate only with a small dimensional feature vectors (2, 5, 10 and 50). With a larger dimension (100, 500, 1000 and 2000), it seems not to yield better accuracy over the baseline classifier. The creation of a weaken classifier by reducing the number of training data seems to be ineffective when testing on a large dimensional feature vectors.

In the following evaluation, a varying number of training tokens was examined. We choose 10- and 100-dimensional feature vectors for evaluation. The results for various training tokens from 300 to 4800 are shown in Figure 6.10. For 10-dimensional feature vectors, an increase of training tokens seems to be ineffective for the recognition performance. The recognition rate always exceeds the baseline classifier with statistical significance at 5% level. For 1000-dimensional feature vectors, the AdaBoost.M1 with a large number of training tokens yielded a higher improvement in recognition rate and exceeded the baseline classifier with a statistical significance at 5% level.

From the evaluations, the results showed that the algorithm seems to be effective for a small dimensional case. For a large dimension, an improvement in recognition rate is obtained with respect to an improvement in the number of training tokens. To achieve better recognition performance for a large dimensional feature vectors, the AdaBoost requires more training tokens. However, the results are based on the nearest neighbour classifier. An evaluation of other classifiers might yield a different result. The results from the synthesised data cannot directly explain the results from the handwritten data

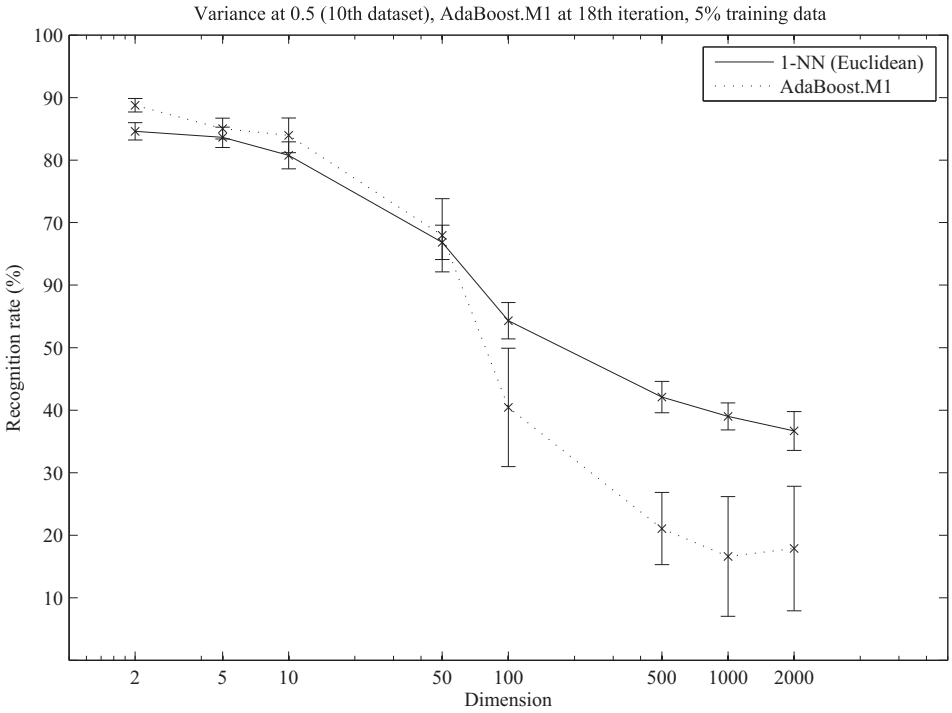


FIGURE 6.9: Comparison of the nearest neighbour classifier and AdaBoost algorithm (M1) at 18th iteration for various dimensional feature vectors.

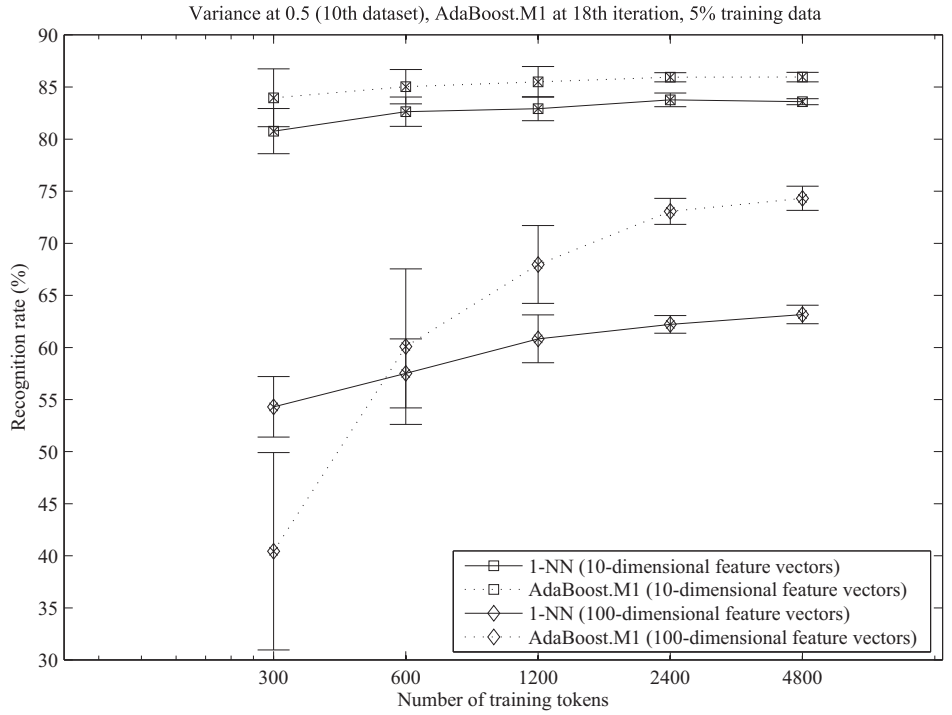


FIGURE 6.10: Comparison of the nearest neighbour classifier and AdaBoost algorithm (M1) at 18th iteration for various training tokens.

because they differ in many aspect. Furthermore, the optimal solution to a simple two-dimensional case is known i.e., linear separators between the classes. This is such a simple case, it bears little resemblance to the real character recognition situation. In other words, if the optimal classifier is known and very simple, there is little or no reason to expect AdaBoost to help to find it.

6.7 Summary and Discussion

This chapter aimed to improve the performance of the classifiers on the handwritten Thai OCR using AdaBoost algorithm. Although, as many papers mentioned, an improvement in the classification rate was obtained from their experiments, in our testing with the strong classifier (back-propagation neural network) on the handwritten Thai datasets, only a slight improvement (sometimes a small decrease) in recognition rate was obtained, as reported in Section 6.3. In the case of the 'within database', the AdaBoost algorithm has demonstrated a slight improvement in the recognition performance (1.2 percentage points) over the original classifiers without applying AdaBoost algorithm. Moreover, when testing on the cross-database, the AdaBoost could not improve the recognition rate over the original classifier. To understand these results, AdaBoost was applied on a controllable dataset. To simplify the problem, we generated a set of synthesised data which had only three classes. Each random sample was represented by a sequence of two-dimensional feature vectors (x, y). The results show that the AdaBoost algorithm still seems not to give better accuracy than the original classifier. We increased the number of iterations to 5,000. The recognition rate improved at the beginning and continually declined towards the end. For the evaluation on an adjusted training set, we obtained an improvement in the recognition rate when we weakened the classifier by reducing the amount of the training size from 75% to 10% and 5%. This enabled the AdaBoost algorithm to achieve better recognition. However, the evaluations on a large dimensional feature vectors, as reported in Section 6.6, showed that reducing the number of training data to weaken the classifier seems to perform effectively on a small, rather than a large dimensional feature vectors.

The results on the synthesised data were obtained by the nearest neighbour classifier applied to AdaBoost.M1. Although the results are based on a different classifier and version of AdaBoost algorithm, we believe that a rather similar result should be obtained by the back-propagation neural network applied to AdaBoost.M2. That is because both of the nearest neighbour classifier and neural network are generally a linear classifier. They rely purely on being able to setup a linear function to identify the class to which it belong. Furthermore, we are looking at the performance of AdaBoost in general. Its principle does not change. According to Freund and Schapire (1996), we found out that if AdaBoost.M1 and .M2 are applied to a simple problem, they would possibly give the same performance. Nonetheless, if applying to a real world problem like a

handwritten Thai, it might be different. Hence, further evaluations of this technique on a real database are required. They are suggested for future research.

Chapter 7

Conclusions and Future Work

The final chapter of this thesis discusses the conclusions that can be drawn from the work presented in the preceding chapters (Section 7.1), and proposes future work for the continuation of the research (Section 7.2).

7.1 Summary of Work

Character recognition techniques associate a symbolic identity with a character image. Character recognition is commonly referred to as optical character recognition (OCR), as it deals with the recognition of optically processed characters. In a typical OCR system, input characters are digitised by an optical scanner. The resulting document image is fed into pre-processing, to improve the quality of the character image. Each character is then segmented and extracted to distinguishing features. After classification, the identified characters are grouped to reconstruct the original symbol strings, and context may then be applied to detect and correct errors by language models.

An essential background to character recognition was presented in Chapter 2. A survey of English OCR was presented, followed by OCR research for other languages, for example, Chinese and Arabic. A basic knowledge of the Thai language and Thai character recognition research was presented. Thai OCR from the National Electronics and Computer Technology Center (NECTEC) was described. This background information is an essential requirement to solving efficiently the problems of Thai character recognition.

An evaluation of the NECTEC Thai OCR system, particularly in its classification process, was reported in Chapter 3. Performance was examined and some limitations to the system were found. In evaluation of the aspect ratio, the ratio was examined between width and character height; Kohonen self-organising maps (SOMs), a rough classifier in NECTEC Thai OCR, were examined. From the evaluation, the accuracy of the system

combined with the aspect ratio slightly improved to about 0.5 percentage points. In the case of SOMs, the comparison between the system, with and without SOMs showed that the system consisting of SOMs as a rough classifier produced a higher recognition rate by about 1.5 percentage points. The complexity of the NECTEC classifier led to a comparison between NECTEC and simple classifiers. A very straightforward classifier, nearest neighbour, was selected as a baseline. Several distance metrics, such as exclusive-OR and Hausdorff distances, were used to evaluate performance and explore the problems of different distance metrics. The evaluation showed that the nearest neighbour classifier, using Hausdorff distance, yielded a significant improvement in recognition rate over the NECTEC classifier. The approximate correct recognition rate achieved 98% on the NECTEC printed character image dataset.

The robustness of the classifiers was tested. Two evaluations were conducted, one for the performance of the classifier on rotated data and one for noisy data. The nearest neighbour classifier, using exclusive-OR, gave better recognition rate results over other classifiers. However, although the nearest neighbour classifier performs with good accuracy, some limitations still exist. For example, the most frequently misclassified characters were of small font size character and lacked significant structure.

Knowledge of Thai printed OCR was applied to offline, handwritten Thai character recognition, which tackles a similar, but more difficult problem. In Chapter 4, three databases were described for the research. 'ThaiCAM' is an offline, handwritten character database, produced in 2001 by the Computer Laboratory, at the University of Cambridge. 'NECTEC-ON' and 'NECTEC-OFF' are online and offline databases provided by NECTEC. The following evaluations were conducted on them. Three different classifiers, nearest neighbour, the back-propagation neural network and hidden Markov models (HMMs), were evaluated. The nearest neighbour classifier using exclusive-OR distance with a basic feature of 64×64 bi-level image (black and white), which achieved very good performance on a NECTEC printed database, was used as baseline. Recognition performance achieved 83.4, 70.2 and 76.7% on ThaiCAM, NECTEC-ON and NECTEC-OFF, respectively. The back-propagation neural network, with an 8×8 matrix feature and HMMs with 'composite image' feature, gave a more effective and higher recognition rate. In the case of the back-propagation neural network, a varying number of hidden nodes, from 16 to 512, were implemented. An improvement in recognition rate was obtained with respect to can increase in the number of hidden nodes. The best results obtained from 512 hidden nodes were 84.9, 84.8 and 77.1% on ThaiCAM, NECTEC-ON and NECTEC-OFF, respectively. Similarly, a various number of HMM states from 40 to 90 was examined. Among a various number of HMM states, the results showed that an improvement in recognition rates was obtained with respect to an improvement in the number of HMM states, when the HMMs numbered less than 70. A recognition rate of 91.1% was obtained on ThaiCAM.

However, when evaluated on different databases, the recognition rates dramatically decreased, especially when training was carried out on the offline database and when tested on the online database. For example, a recognition accuracy of 87.1% was obtained from the back-propagation neural network when training was carried out and tested on ThaiCAM; but recognition rates of only 58.3 and 57.4% were achieved when evaluated on NECTEC-ON and NECTEC-OFF, respectively. This indicates that the generalisation ability of the classifiers is not good enough.

In Chapter 5, an improvement in the overall recognition performance among the three different databases was obtained to improve the generalisation ability of the classifiers. A difference among the databases in the case of stroke width was investigated. From the evaluation, the stroke width in NECTEC-ON was thinner compared to that of ThaiCAM and NECTEC-OFF. An adjustment in stroke width, using a basic image processing operation called erosion and dilation, was applied in the evaluation. A NECTEC-ON image was applied, using a dilation operation to increase stroke width. For the back-propagation neural network, the best result showed an improvement in the recognition rate from 58.3 to 63.9%, or equivalent to a 13.4% relative error reduction. For HMMs, the best achievement improved the recognition accuracy from 55.4 to 67.4%. This is equivalent to a 26.9% relative error reduction. However, when the dilation operation was applied twice, the stroke width of the character was too thick. This results in missing some significant features and leads directly to a decrease in recognition performance. Similarly, applying the erosion operation to decrease stroke width in ThaiCAM had a disastrous effect on recognition performance. The reduction in stroke width results in many significant features missing as well.

A combination of the three databases for training in the classifier was carried out. The training sets from three databases were merged and used for training in the classifiers. The recognition rate, when applied to the new training set, achieved approximately 86.0 and 83.6% for the back-propagation neural network and HMMs, whereas average recognition rates of 68.0 and 76.0% (for the back-propagation neural network and HMMs) were obtained after training with a single database training set. These showed a significantly better performance, equivalent to a 56.3 and 31.7% relative error reduction.

In Chapter 6, another method to improve the generalisation ability of a classifier was considered, the use of a boosting algorithm. AdaBoost, a machine learning algorithm for improving the accuracy of a classifier, was applied to our classifier. In our evaluation, the AdaBoost.M2 gave the best recognition rate. However, the results showed a slight improvement in recognition rates over the original classifier (without applying the AdaBoost algorithm). The best result obtained by AdaBoost.M2 (trained on ThaiCAM) gave a recognition rate of 88.3%, which is equivalent to a 9.3% relative error reduction from the original classifier (87.1%). The cross-database did not obtain an improvement on AdaBoost.M2. The recognition rate for AdaBoost.M2 was slightly

lower than the original classifier (approximately 0.6%). Using McNemar's test, it could not be considered to be statistically significant.

Using the AdaBoost algorithm to improve the recognition rate seems to be ineffective. A slight improvement in recognition rates can be obtained. To understand these results, the AdaBoost algorithm was evaluated on a synthesised dataset. The results showed that its recognition performance still seemed not to give better accuracy than the original classifiers. The evaluation on an increasing number of iterations showed an improvement in recognition rate at the beginning, continually declining towards the end. Reduction in the amount of the training size from 75 to 10 and 5% achieved an improvement in recognition rate on a small vector, but cannot be applied on a large dimensional feature vectors. This result, however, was evaluated on synthesised data. Further studies to evaluate the technique on real databases, e.g. ThaiCAM, NECTEC-ON and NECTEC-OFF, are also important. However, these are beyond the scope of this thesis and are left for future work.

7.2 Suggested Future Work

This thesis has been concerned with the study of Thai character recognition, including printed and handwritten script. We attempted to overcome the problem by simple but effective methods. Significant results and improvement in correct classification rates were obtained. However, there is still much scope for further investigation, which includes:

- For machine printed Thai character recognition:
  - As the results showed in Section 3.5.4, the most frequently misclassified character (nearest neighbour classifier) using exclusive-OR are small font size and featureless characters. Most of them are special symbols, upper and lower vowel. When they are size-normalised into a 32×32 binary image and then into an 8×8 feature matrix, most of them are converted into black squares, which are very difficult to identify by the classifier. A Thai sentence consisting of four levels, as illustrated in Figure 2.2, can be classified by the position of characters in comparison to others, and the level of each character can be a good feature for the classification of similar characters. Furthermore, linguistic post-processing, for example, spell-checking and the part-of-speech (POS) n-gram model may be applied to resolve the problem.
  - Although the nearest neighbour classifier performs with good accuracy over the NECTEC classifier, some limitations still exist. It consumes vast computational resources and is expensive in terms of longer recognition and huge disk space. Further studies to reduce the computational resources are important.

- For offline handwritten Thai character recognition:
  - In Section 5.3, a form of image processing, the dilation operation, was applied to an online database (NECTEC-ON) to increase stroke width. Although an image adjusted by this technique is quite similar to an image recreated from temporal information, further studies should be evaluated on images recreated from temporal information.
  - In Section 5.4, a combination of three databases for training the classifiers was carried out on the original databases. From the results in Section 5.3, an improvement in recognition rate was obtained from an increase in stroke width on the NECTEC-ON database. A combination of three databases for training the classifiers should be performed on the adjusted rather than the original database. It is likely to produce a relative improvement in correct classification rates.
  - The techniques used in this research were applied to isolated handwritten Thai character recognition. It requires a handwritten text to be segmented into individual Thai characters. When writing is done quickly and carelessly, correct segmentation tends to be difficult to achieve, because contiguously written characters may connect to each other, yet without any change in each characters appearance. This is a difficult problem, because touching Thai characters may occur in both horizontal and vertical positions, since some characters can be written above or below other characters. Two different approaches can be applied: 'segmentation-based' and 'segmentation-free'. The segmentation-based approach attempts to segment the handwritten words into characters. This approach converts the problem of cursive handwriting recognition into an isolated character recognition problem, to which our techniques are applicable. Nevertheless, it requires a very powerful segmentation procedure in order to achieve good performance. Whereas the segmentation-free approach obviously does not require any segmentation procedure, we need a classification method that enables the modelling of continuous signals. A method of tackling this problem is either to concentrate on improving the segmentation procedure or to utilise a segmentation-free approach. The HMMs are capable of modelling continuous signals applied to a segmentation-free approach. However, this requires further explorations of techniques to model vertical touching characters by HMMs.
  - The results in Section 6.5 obtained an improvement in recognition rates when we weakened the classifier by reducing the number of the training size from 75% to 10% and 5% respectively. This result was based on a synthesised dataset with only three classes. Each sample was represented by a sequence of two-dimensional feature vectors (x, y). However, when we tested on a large dimensions, we cannot obtain an improvement in recognition rates. Hence,

further investigations on this technique on real data (which have more classes and more dimensional feature vectors) are required.

– A benchmark database of Thai characters is required, on which different researchers can report results. Until now, more research on handwritten Thai character recognition is evaluated by proposed methods from the collected data. To offset this, the three databases used in this work (ThaiCAM, NECTEC-ON and NECTEC-OFF) will be published and made freely available to other researchers. With the availability of a standard database for research and evaluation, we hope that others will make further progress on the challenge of the pattern recognition problem.

Bibliography

Airphaiboon, S. and Kondo, S. (1996). Recognition of handprinted Thai characters using loop structures. *IEICE Transactions on Information and Systems*, **E79-D**(9), 1296–1304.

Amin, A. (1997). Offline Arabic character recognition - a survey. In *4th International Conference on Document Analysis and Recognition*, volume 2, pages 596–599, Ulm, Germany.

Arica, N. and Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **31**(2), 216–233.

Athitsos, V. and Sclaroff, S. (2005). Boosting nearest neighbor classifiers for multiclass recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 45–52, San Diego, California, USA.

Bhattarakosol, P. (2003). It direction in Thailand: Cultivating an e-society. *IT Professional*, **5**(5), 16–20.

Bishop, C. M. (1995). *Neural Network for Pattern Recognition*. Oxford university press.

Bounnady, K., Kruatrachue, B., and Matsuura, T. (2008). Online unconstrained handwritten Thai character recognition using multiple representations. In *International Symposium on Communications and Information Technologies*, pages 135–140, Vientiane, Lao PDR.

Bunke, H. (2003). Recognition of cursive Roman handwriting - past, present and future. In *7th International Conference on Document Analysis and Recognition*, volume 1, pages 448–459, Edinburgh, Scotland.

Bunke, H., Roth, M., and Schukat-Talamazzini, E. G. (1995). Off-line cursive handwriting recognition using hidden Markov models. *Pattern Recognition*, **28**(9), 1399–1413.

Cha, S.-H., Yoon, S., and Tappert, C. C. (2005). On binary similarity measures for handwritten character recognition. In *8th International Conference on Document Analysis and Recognition*, volume 1, pages 4–8, Seoul, Korea.

Chanda, S., Terrades, O. R., and Pal, U. (2007). SVM based scheme for Thai and English script identification. In *9th International Conference on Document Analysis and Recognition*, volume 1, pages 551–555, Curitiba, State of Parana, Brazil.

Chatfield, C. (1995). *Statistics for Technology: A Course in Applied Statistics*. Chapman & Hall, third (revised) edition.

Chim, Y. C., Kassim, A. A., and Ibrahim, Y. (1999). Character recognition using statistical moments. *Image and Vision Computing*, **17**(17), 299–307.

Choruengwiwat, P., Jitapunkul, S., Wuttisittikulkij, L., and Seehapan, P. (1998). Distinctive feature analysis for Thai handwritten character recognition based on modified stroke changing sequence. In *IEEE Asia-Pacific Conference on Circuits and Systems*, pages 543–546, Chiangmai , Thailand.

Dougherty, E. R. and Lotufo, R. A. (2003). *Hands-on Morphological Image Processing*. SPIE Press.

Drucker, H. (1996). Fast decision tree ensembles for optical character recognition. In *The Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 137–147, Las Vegas, Nevada, USA.

Duangphasuk, P. and Thammano, A. (2006). Thai vehicle license plate recognition using the hierarchical cross-correlation ARTMAP. In *3rd International IEEE Conference Intelligent Systems*, pages 652–655, London, UK.

Duangphasuk, S. and Premchaiswadi, N. (1999). Document image skew detection. In *3rd National Computer Science and Engineering Conference*, pages 146–151, Bangkok, Thailand. (in Thai).

Dubuisson, M.-P. and Jain, A. K. (1994). A modified Hausdorff distance for object matching. In *12th International Conference on Pattern Recognition*, volume 1, pages 566–568, Jerusalem , Israel.

El-Hajj, R., Likforman-Sulem, L., and Mokbel, C. (2005). Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. In *8th International Conference on Document Analysis and Recognition*, volume 2, pages 893–897, Seoul, Korea.

El-Yacoubi, A., Gilloux, M., Sabourub, R., and Suen, C. (1999). An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(8), 752–760.

Flusser, J. and Suk, T. (1994). Affine moment invariants: A new tool for character recognition. *Pattern Recognition Letters*, **15**(4), 433–436.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**, 119–139.

Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, **14**(5), 1–13.

Fu, Q., Ding, X., and Liu, C. (2008). A new AdaBoost algorithm for large scale classification and its application to Chinese handwritten character recognition. In *International Conference on Frontiers in Handwriting Recognition*, pages 43–48, Montral, Qubec, Canada.

Garain, U. and Chaudhuri, B. (2003). On machine understanding of online handwritten mathematical expressions. In *7th International Conference on Document Analysis and Recognition*, volume 1, pages 349–353, Edinburgh, Scotland.

Gunter, S. and Bunke, H. (2002). A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In *16th International Conference on Pattern Recognition*, volume 2, pages 332–337, Montral, Qubec, Canada.

Hao, W. and Luo, J. (2006). Generalized multiclass AdaBoost and its applications to multimedia classification. In *Computer Vision and Pattern Recognition Workshop*, pages 113–118, New York City, New York, USA.

Hiranvanichakorn, P. and Boonsuwan, M. (1993). Recognition of Thai characters. In *The Symposium on Natural Language Processing*, pages 123–166, Bangkok, Thailand.

Hiranvanichakorn, P., Agui, T., and Nakajima, M. (1982). A recognition method of Thai characters. *The Institute of Electronics and Communication Engineers of Japan*, **E65**(12), 737–744.

Hiranvanichakorn, P., Agui, T., and Nakajima, M. (1984). Recognition method of Thai characters by using local features. *The Institute of Electronics and Communication Engineers of Japan*, **E67**(8), 425–432.

Hiranvanichakorn, P., Agui, T., and Nakajima, M. (1985a). An on-line recognition method of Thai characters. *The Institute of Electronics and Communication Engineers of Japan*, **E68**(9), 594–601.

Hiranvanichakorn, P., Agui, T., and Nakajima, M. (1985b). A recognition method of handprinted Thai characters by local features. *The Institute of Electronics and Communication Engineers of Japan*, **E68**(2), 83–90.

Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(9), 850–863.

Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. McGraw-Hill.

Jelinek, F. (1998). *Statistical Methods for Speech Recognition*. The MIT Press.

Jun, G. and Ghosh, J. (2009). Multi-class boosting with class hierarchies. In *International Workshop on Multiple Classifier Systems*, pages 32–41, Reykjavik, Iceland.

Khorsheed, M. (2003). Recognising handwritten Arabic manuscripts using a single hidden Markov model. *Pattern Recognition Letters*, **24**(14), 2235–2242.

Kimpan, C. and Walairacht, S. (1993). Thai characters recognition. In *The Symposium on Natural Language Processing*, pages 196–276, Bangkok, Thailand.

Koerich, A. L., Sabourin, R., and Suen, C. Y. (2003). Lexicon-driven HMM decoding for large vocabulary handwriting recognition with multiple character models. *International Journal on Document Analysis and Recognition*, **6**(2), 126–144.

Kortungsap, P., Lekhachaivorakul, P., and Madarasmi, S. (1999). On-line handwriting recognition system for the Thai, English, numeral and symbol characters. In *3rd National Computer Science and Engineering Conference*, pages 255–259, Bangkok, Thailand. (in Thai).

Lee, L. L. and Coelho, S. I. (2005). A simple and efficient method for global handwritten word recognition applied to Brazilian bankchecks. In *8th International Conference on Document Analysis and Recognition*, volume 2, pages 950–954, Seoul, Korea.

Limmaneewichid, P. and Premchaiswadi, N. (1999). Repairing broken Thai printed characters using feature extraction. In *3rd National Computer Science and Engineering Conference*, pages 152–157, Bangkok, Thailand. (in Thai).

Liu, C., Kim, I., and Kim, J. H. (2001). Model-based stroke extraction and matching for handwritten Chinese character recognition. *Pattern Recognition*, **34**(12), 2339–2352.

Lorette, G. (1999). Handwriting recognition or reading: What is the situation at the dawn of the 3rd millenium? *International Journal on Document Analysis and Recogntion*, **2**(1), 2–12.

Lorigo, L. M. and Govindaraju, V. (2006). Offline Arabic handwriting recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(5), 712–724.

Madarasmi, S. and Lekhachaiworakul, P. (2000). Customizable on-line Thai/English handwriting recognition system. In *4th Symposium on Natural Language Processing*, pages 142–153, Chiangmai, Thailand.

Manmatha, R. and Rothfeder, J. L. (2005). A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(8), 1212–1225.

Margner, V., Pechwitz, M., and Abed, H. E. (2005). ICDAR 2005 Arabic handwriting recognition competition. In *8th International Conference on Document Analysis and Recognition*, volume 1, pages 70–74, Seoul, Korea.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, **8**(1), 3–30.

Meknavin, S., Kijsirikul, B., Chotimongkol, A., and Nuttee, C. (1998). Combining trigram and Winnow in Thai OCR error correction. In *36th annual meeting on Association for Computational Linguistics*, volume 2, pages 836–842, Montral, Qubec, Canada.

Methasate, I. and Sae-Tang, S. (2002). On-line Thai handwriting character recognition using stroke segmentation with HMM. In *20th IASTED International Conference on Applied Informatics - Artificial Intelligence and Applications*, pages 59–62, Innsbruck, Austria.

Methasate, I., Marukatat, S., Sae-Tang, S., and Theeramunkong, T. (2005). The feature combination technique for off-line Thai character recognition system. In *8th International Conference on Document Analysis and Recognition*, pages 1006–1009, Seoul, Korea.

Miletzki, U. (1997). Character recognition in practice today and tomorrow. In *4th International Conference on Document Analysis and Recognition*, volume 2, pages 902–907, Ulm, Germany.

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

Mozaffari, S., Faez, K., and Ziaratban, M. (2005). Structural decomposition and statistical description of Farsi/Arabic handwritten numeric characters. In *8th International Conference on Document Analysis and Recognition*, volume 1, pages 237–241, Seoul, Korea.

Nopsuwanchai, R. (2005). *Discriminative Training Methods and Their Applications to Handwriting Recognition*. Ph.D. thesis, University of Cambridge.

Nopsuwanchai, R. and Clocksin, W. F. (2003). Hidden Markov models for off-line Thai handwriting recognition. In *11th International Conference on Artificial Intelligence Applications*, pages 180–189, Cairo, Egypt.

Nopsuwanchai, R. and Povey, D. (2003). Discriminative training for HMM-based offline handwritten character recognition. In *7th International Conference on Document Analysis and Recognition*, volume 1, pages 114–118, Edinburgh, Scotland.

Nopsuwanchai, R., Biem, A., and Clocksin, W. F. (2006). Maximization of mutual information for offline Thai handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(8), 1347–1351.

Parker, J. R. (1996). *Algorithms for Image Processing and Computer Vision*. Wiley, New York.

Pati, P. B. and Ramakrishnan, A. G. (2007). A blind Indic script recognizer for multi-script documents. In *9th International Conference on Document Analysis and Recognition*, volume 2, pages 1248–1252, Curitiba, State of Parana, Brazil.

Pechwitz, M. and Margner, V. (2003). HMM based approach for handwritten Arabic word recognition using the IFN/ENIT - database. In *7th International Conference on Document Analysis and Recognition*, pages 890–894, Edinburgh, Scotland.

Phokharatkul, P. and Kimpan, C. (1998). Recognition of handprinted Thai characters using the cavity features of character based on neural network. In *Asia Pacific Conference on Circuits and Systems*, pages 149–152, Chiangmai , Thailand.

Phokharatkul, P. and Kimpan, C. (2002). Handwritten Thai character recognition using Fourier descriptors and genetic neural networks. *Computational Intelligence*, **18**(3), 270–293.

Phokharatkul, P., Sankhuangaw, K., Somkuarnpanit, S., Phaiboon, S., and Kimpan, C. (2005). Off-line hand written Thai character recognition using Ant-Miner algorithm. *World Academy of Science, Engineering and Technology*, **8**, 276–281.

Phokharatkul, P., Chatchawalanonth, D., and Kimpan, C. (2006). A fuzzy and rough sets approach for recognition of handwritten Thai characters. In *International Conference on Signal Processing, Robotics and Automation*, pages 147–152, Madrid, Spain.

Pomchaikajomsak, A. and Thammano, A. (2003). Handwritten Thai character recognition using fuzzy membership function and fuzzy ARTMAP. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 1, pages 40–44, Kobe, Japan.

Pornpanomchai, C., Batanov, D. N., and Dimmitt, N. (2001). Recognizing Thai handwritten characters and words for human-computer interaction. *International Journal of Human-Computer Studies*, **56**(3), 259–279.

Premchaiswadi, N. (2001). *A Study on Printed Thai Character Recognition*. Ph.D. thesis, Waseda University.

Premchaiswadi, N., Premchaiswadi, W., Pachiyanukul, U., and Narita, S. (2003). Broken characters identification for Thai character recognition systems. *WSEAS Transactions on Computers*, **2**(2), 430–434.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

Rucklidge, W. (1996). *Efficient Visual Recognition Using the Hausdorff Distance*. Springer-Verlag New York, Inc.

Sae-Tang, S. and Methasate, I. (2002). Thai online handwritten character recognition using windowing backpropagation neural networks. In *20th IASTED International Conference on Modelling, Identification, and Control*, pages 337–340, Innsbruck, Austria.

Sae-Tang, S. and Methasate, I. (2004). Thai handwritten character corpus. In *IEEE International Symposium on Communications and Information Technologies*, volume 1, pages 486–491, Sapporo, Japan.

Santinanalert, C. (1999). *Design and Development of Thai-OCR Program*. Master's thesis, Chulalongkorn University.

Schwenk, H. and Bengio, Y. (1997a). AdaBoosting neural networks: Application to on-line character recognition. In *International Conference on Artificial Neural Networks*, pages 967–972, Lausanne, Switzerland.

Schwenk, H. and Bengio, Y. (1997b). Training methods for adaptive boosting of neural networks for character recognition. In *International Conference on Artificial Neural Networks*, pages 1–9, Lausanne, Switzerland.

Schwenk, H. and Bengio, Y. (2000). Boosting neural networks. *Neural Computation*, **12**(8), 1869–1887.

Sheskin, D. J. (2004). *Handbook of Parametric and Nonparametric statistical Procedures Third Edition*. Chapman & Hall.

Shi, D. (2002). *An Active Radical Approach to Handwritten Chinese Character Recognition*. Ph.D. thesis, University of Southampton.

Shi, D., Damper, R. I., and Gunn, S. R. (2003). Offline handwritten Chinese character recognition by radical decomposition. *ACM Transactions on Asian Language Information Processing*, **2**(1), 27–48.

Sornlertlamvanich, V., Potipiti, T., Potipiti, T., and Mittrapiyanuruk, P. (2000). The state of the art in Thai language processing. In *38th Annual Meeting on Association for Computational Linguistics*, pages 1–2, Hong Kong.

Suen, C. Y., Mori, S., Kim, S. H., and Leung, C. H. (2003). Analysis and recognition of Asian scripts - the state of the art. In *7th International Conference on Document Analysis and Recognition*, volume 2, pages 866–878, Edinburgh, Scotland.

Sun, Y., Todorovic, S., and Li, J. (2007). Unifying multi-class AdaBoost algorithms with binary base learners under the margin framework. *Pattern Recognition Letters*, **28**, 631–643.

Tanprasert, C. and Koanantakool, T. (1996). Thai OCR: A neural network application. In *IEEE TENCON Digital Singal Processing Applications*, volume 1, pages 90–95, Perth, Australia.

Tanprasert, C., Sinthupinyo, W., and Dubey, P. (1996). On the printed Thai optical character recognition software project. In *7th NECTEC Annual Conference*, pages 1–4, Bangkok, Thailand. (in Thai).

Tanprasert, C., Sinthupinyo, W., Dubey, P., and Tanprasert, T. (1997). Improved mixed Thai & English OCR using two-step neural net classification. In *International Conference on Neural Information Processing and Intelligent Information System*, volume 2, pages 1227–1230, Dunedin, New Zealand.

Tappert, C. C., Suen, C. Y., and Wakahara, T. (1990). The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(8), 787–808.

Thammano, A. and Duangphasuk, P. (2005). Printed Thai character recognition using the hierarchical cross-correlation ARTMAP. In *17th IEEE International Conference on Tools with Artificial Intelligence*, pages 695–698, Hong Kong.

Thammano, A. and Ruxpakawong, P. (2002). Printed Thai character recognition using the hybrid approach. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E85-A**(6), 1236–1241.

Therramunkong, T., Wongtapan, C., and Sinthupinyo, S. (2002). Offline isolated handwritten Thai OCR using island-based projection with n-gram models and hidden Markov models. In *International Conference on Asian Digital Libraries*, pages 340–351, Singapore.

Thongprasirt, R., Charoenporn, T., Sinthupinyo, W., and Sortlertlamvanich, V. (2001). Development of very large corpora in Thailand. In *6th Natural Language Processing Pacific Rim Symposium*, pages 1–7, Tokyo, Japan.

Tomai, C. I., Zhang, B., and Govindaraju, V. (2002). Transcript mpping for historic handwritten document images. In *8th International Workshop on Frontiers in Handwriting Recognition*, pages 413–418, Ontario, Canada.

Trier, O. D., Jain, A. K., and Taxt, T. (1996). Feature extraction methods for character recognition – a survey. *Pattern Recognition*, **29**(4), 641–662.

Yingsaeree, C. and Kawtrakul, A. (2005). Rule-based middle-level character detection for simplifying Thai document layout analysis. In *8th International Conference on Document Analysis and Recognition*, volume 2, pages 888–892, Seoul, Korea.

Young, S., Evermann, G., Gales, M., Gales, M., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2009). *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department.

Zell, A., Mamier, G., Vogt, M., Mache, N., Hubner, R., Doring, S., Herrmann, K.-U., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., and Wielang, J. (1995). *SNNS Stuttgart Neural Network Simulator User Manual, Version 4.1*. Institute for Parallel and Distributed High Performance systems (IPVR), University of Stuttgart.

Zhang, H., Yang, J., Deng, W., and Guo, J. (2008). Handwritten Chinese character recognition using local discriminant projection with prior information. In *19th International Conference on Pattern Recognition*, pages 1–4, Tampa, Florida, USA.

Zhao, C., Shi, W., and Deng, Y. (2005). A new Hausdorff distance for image matching. *Pattern Recognition Letters*, **26**(5), 581–586.

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class AdaBoost. *Statistics and Its Interface*, **2**, 349–360.