SAMPLE-SYNTHESIZED SIMULATED LIGHT SENSITIVE MODEL FOR STRUCTURAL CATEGORIZATION OF HANDWRITTEN THAI CHARACTER

Mr. KAMOLCHOL JITWONGTRAKUL

Submitted in Partial Fulfillment of
the Requirements for the Degree of

MASTER OF SCIENCE

In Computer Science

ASSUMPTION UNIVERSITY

May, 2001

Sample-Synthesized Simulated Light Sensitive Model for Structural Categorization of Handwritten Thai Character

by

Mr. Kamolchol Jitwongtrakul

Submitted in Partial Fulfillment of
The Requirements for the Degree of
Master of Science
In Computer Science
Assumption University

May, 2001

The Faculty of Science and Technology

Thesis Approval

Thesis Title	Sample-Synthesized Simulated Light Sensitive Model for Structural Categorization of Handwritten Thai Character
By	Mr. Kamolchol Jitwongtrakul
Thesis Advisor	Dr. Thitipong Tanprasert
Academic Year	1/2001

The Department of Computer Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **SC7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Approval Committee:

(Dr. Thitipong Tanprasert)
Advisor

(Dr. Jirapun Daengdej)
Committee Member

(Asst.Prof.Dr. Dobri Batovski)
Committee Member

(Professor Dr. Chidchanok Lursinsap)
Representative of Ministry of
University Affairs

Faculty Approval:

(Asst.Prof.Dr. Tang Van To)
Director

(Asst.Prof.Dr. Pratit Santiprabhob)
Dean

June / 2001

ABSTRACT

Thai Handwritten Recognition is a challenging problem. The researches on this topic have been started since 1989 but many solutions depend on some constraints and they are not applicable for using in real life environment.

From other research [6], a technique for a rough classification of 44 Thai alphabets into groups has been proposed. The aim of this research is trying to find the fine classification step that will distinguish alphabets within groups resulted from rough classification. The proposed classification was based on local contour analysis and it uses Fuzzy and Neural Network for recognition engine system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to handwritten recognition problem

Off-line handwritten recognition is not a new problem. It is one of the challenging problems in image understanding. This problem involves many factors such as vast variations of the character shapes and orientation, scanning noise and the lack of information on the handwriting strokes. The recognition procedure typically consists of feature extraction and classification or recognition.

Off-line recognition lacks information about the movement of pen, number of floating pen, starting point and ending point like in the online system. These would make off-line recognition more complicated and difficult than the online system.

Off-line handwritten recognition has been developed for several languages such as English, Chinese, and Japanese. At first aid, researchers performed many techniques with the printed characters. In the second aid they expanded their scopes to handwritten characters recognition. Unfortunately, almost all of these techniques cannot be used directly, without any modification, with Thai characters because the structure of Thai characters is different from others. Basically, structure of Thai characters is composed by the combination of straight line, curve line and circle. Some Thai characters look very similar to each other, only different on some special features.

Most off-line Thai handwritten recognition systems can not perform well in typical handwriting because those systems depend on some constraints such as head, number of head, position of head, cross, junction, and notch. These constraints are easily detected by human. However, for computer, it is too complex to be deterministically synthesized due to a set of rules used for computer detection. Most works were induced from human perception that has ability to ignore noises. Thus, the noise or any distortion that can always occur in the real world is not efficiently handled.

Generally, recognition system can be separated into two main parts; there are feature extraction and classification. Each part has their significances which would be discussed later.

1.2 Background and motivation

Most of existing Thai handwritten recognition systems [1], [8], [9], [10] and [11] claimed their systems perform with very high accuracy (98% -99%) in recognition rate. However, most of them could not apply in the typical handwriting of real life because they ignored the common real factor such as noise when scanning or different style of character sets. They try to scope their system limitation with some constraints such as head, number of head, position of head, cross, junction, and notch. As a result, their systems can not work with other variation character sets by their rules.

Because some of Thai characters looked quite similar, we can only distinguish them by some features like notch, tail and location of head at the specific area. For example, Thai characters, ข and ฃ can be distinguished by notch near head, or characters ก, ภ, ถ can be distinguished only by head at the area left bottom side.

1.3 Thesis Organization

This thesis is organized into 7 chapters as follow:

Chapter 1 (**"INTRODUCTION"**) discusses about introduction of character recognition system, their scope, and problems. Next is the motivation in doing the research.

Chapter 2 (**"LITERATURE REVIEW"**) reviews other existing recognition systems and makes a comparison of weak & strong point of each system. It derives us to design our own system.

Chapter 3 (**"PROPOSED SYSTEM"**) describes the system architecture overview. Next, we scope for our work boundary.

Chapter 4 (**"LOCAL CONTOUR ANALYSIS"**) is a summary of local contour technique and how it is to be adapted to use with our task.

Chapter 5 (**"NEURAL NETWORK"**) describes about the initial experiment result and problems of our experience.

Chapter 6 (**"EXPERIMENT and RESULT"**) explains the experiment environment and the result that we got. We also give our opinion on the correctness percent result. It also describes the lack of some abilities and limitation of our system. At the end of this chapter there is a discussion for future research and development by explaining our goal or how to improve our system.

Chapter 7 (**"CONCLUSION"**) finally, we give our conclusion for this system.

"BIBLIOGRAPHY" contains all reference papers that are ordered by ascending and by last name of the first author.

The last section is **("appendix")**, we provide some examples of alphabet images and our rule to do smooth and spur.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The problem of off-line handwritten character recognition system is a challenging one because of the different styles of writing, the noise from scanning to a raster image, the lack of information about stroke structure, trajectory of handwritten, and the movement of floating pen. Differences of writing style are inconsistent. This makes the feature extraction and matching processing more difficult and unstable.

In off-line recognition system, the input characters are scanned optically and saved as raster image. Several techniques are suffered with noise sensitive and which does not follow their rules.

2.2 Existing Thai Character Recognition System and problems

In this section we will describe the existing Thai and other recognition system and their concepts. First, we discuss the one that [5] used the integration of structural and statistical pattern recognition for unconstrained Handwritten Numeral Recognition system. It used the information of the traveling around only outer contour. However, the traversing method that required width of every point on the image after preprocessing must be at least three pixels wide.

[7] use topological properties with Thai Printed characters. The K-L transform find the Covariance matrix, Eigen value and get bound, skeleton and contour data. The result shows that it can be applied with a variety of writing style but it cannot distinguish similar characters from each other.

[10] presents the syntactic method with Thai Printed characters. It extracts the data from input image to the vector and mapping together with Binary-Tree of Order 3 that compose data of Root Point, End Point, Circle Point, Junction Point for input of fuzzy system. This system is tides with rules and they used only 2 kinds of Thai Fonts for testing.

[11] implements inductive logic programming to Thai Printed Character Recognition. This method looks similar to [10] but recognition technique was changed from fuzzy to back propagation neural network. The performance of this system is close to [10].

[1] They define the rule of Thai character classification as very tide such as "Number of head", "Tail". The input characters come from drawing program that does not have any noise so that it is smooth and does not require any noise removal step.

2.3 Goal of System

Our goal is to find the system that can be used in reality. Therefore, the goal of our system is to find the system that:

-Can perform well in each data set that has different database style of writing

-Can perform in high noise-tolerance environment

-Can distinguish the similar alphabets like "ก", "ภ" and "ถ"

Because the correct rate in two-stage system comes from multiple correct rate of first and second stage, the error recognition in two-stage system will be higher than in one-stage system. However, although one-stage recognition system will give lower error recognition rate, unfortunately, it is hard or impossible to find any existing recognition system that can perform well in only one stage.

From Template matching and Structure Analysis technique, each technique has their own advantages. If we can combine them together, we will get the advantages of both techniques to achieve our goal.

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction

In this chapter we will explain our system. First section is “System Overview” that describes the whole picture of system and its features. Second section is “Scope of work”, which refer to the previous initial environment research, we continue doing the research for developing the whole system. Finally, “System architecture” section describes the reasons why we design our system in the following way.

3.2 System Overview

As our system goal stated in section 2.3, system must be able to perform:

- In different styles of writing data set
- In High noise-tolerance environment
- In open system
- With low error recognition rate
- Distinguish the similar alphabets that need to be distinguished

To achieve every goal, it is necessary to separate our system into two main phases. The first phase requires template matching to do rough classification and the second phase is the use of structure analysis for work on fine recognition.

Our goal is trying to find the way to recognize a real-life handwritten Thai alphabet. We can apply the different techniques in each phase because each of the existing technique has its strong and weak points.

3.2.1 One phase and Two phases system

Almost all of the traditional recognition systems use only one phase system. However, this system lacks some important functions. The advantages and disadvantages of each system will be discussed as follows:

Comparison of One phase and Two phases system

One Phase system

- Higher recognition rate. One phase system will get lower error recognition rate compared to two phase's recognition system. Because two phases recognition error rate comes from the multiplication of first and second phase, the error rate in two phases would be more than in one phase system.
- Fast computation time. Most of one phase systems can run faster than two phase's system.
- Close system. Most of one phase systems are close system, so it cannot respond as reject if the input is not alphabet characters. It will answer one alphabet then the recognition rate will be reduced.
- Lack of concurrency. An input image must wait until the recognition result returned from the system.

Two Phases system

- Able to use different techniques. We can use different techniques in each phase. Therefore, we can gain advantages from each technique.
- Able to work concurrency. Each stage can have more than one component and work independently like parallel. Each component can work concurrency. We can have the decision maker for control and choose the best answer for any phases that have more than one component.
- Open system. Two phases system is an open system so it would reject if the input image is not alphabet in their known set

From the above analysis, the advantages of two phases system are more than one phase system. For example, such advantages are open system, Modularity, Scalability and Concurrency function. Therefore, we will apply two phases system although it is more complicated and more difficult to implement.

3.2.2 Classification & Recognition

In our design system, we separate system in to two phases. First is the classification path and second is the recognition that has some different keys to achieve.

For classification path, after doing rough classification, it must perform well in high noise tolerance environment because the non-alphabet will be rejected. The recognition part concerns fine classification so that it must be able to distinguish the similar characters. In addition, both parts should have high accuracy rate and can perform in different alphabet writing image style.

3.3 Scope of work

Due to [6], the first phase (rough classification) had already been finished. Therefore, this research will focus only on the second phase (fine classification). We will categorize alphabets on each group into group one to eleven.

Table 3.1, [6] classified all 44 characters into 12 groups. Each group has different number of members.

- This research only cover all 44 Thai alphabets which are

  ก ข ฃ ค ฅ ฆ ง

  จ ฉ ช ซ ฌ ญ

  ฎ ฏ ฐ ฑ ฒ ณ

  ด ต ถ ท ธ น

  บ ป ผ ฝ พ ฟ ภ ม

  ย ร ล ว ศ ษ ส ห ฬ อ ฮ

- Allows various kinds of handwritten style
- High recognition rate for both clear and noisy input images.
- The system assumes that Thai alphabet image is isolated by preceding stage before.

Table 3.1 12 groups of Thai alphabets classification.

Group No.	Character in group
1	ก ถ ภ
2	ข ฃ บ ย ษ
3	ค ฅ ด ต ศ
4	ฆ ม
5	ง ว อ
6	จ ฉ ล ส
7	ฌ ญ ฒ ณ
8	ฎ ฏ
9	ฑ ท น ห
10	ช ซ ฐ ธ ร ฮ
11	ป ผ ฝ พ ฟ
12	ฬ

3.4 System architecture

As previously stated, our focus will be only on the second phase (fine classification). We can separate our system into three main parts. Part one is preprocessing; it is required for smoothing data and removing some noises. The next part is Local Contour in feature extraction part and classifier part. This is the new approach that recognizes the input character by using local contour analysis technique (The description of this technique will be given later in the next chapter). Each group will use one Neural Network but in some groups we will add a special stage classification shown in Figure 3.1. From the above 11 groups we use 20 Neural Network as described in Figure 3.1.

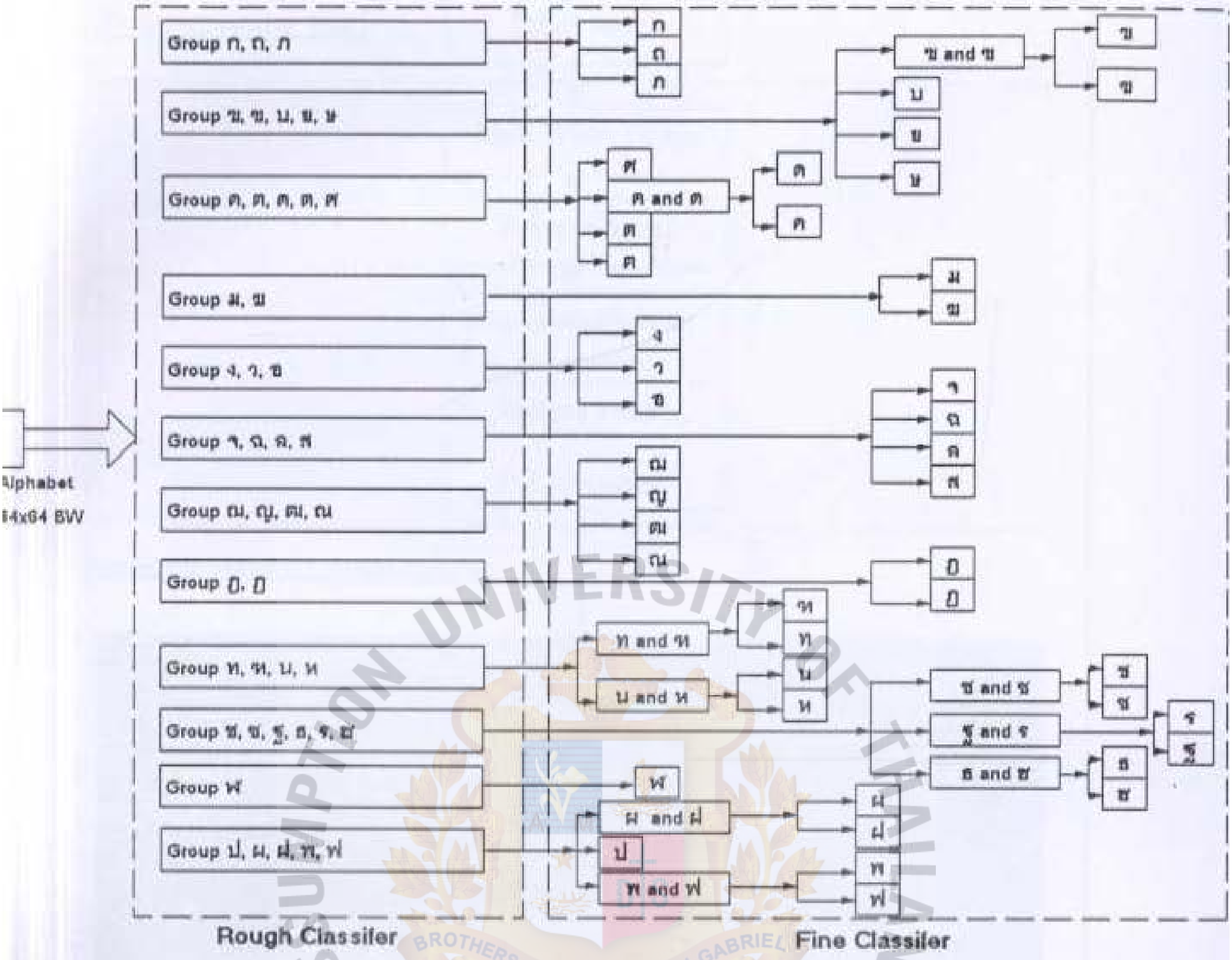


Figure 3.1: Group classification stage

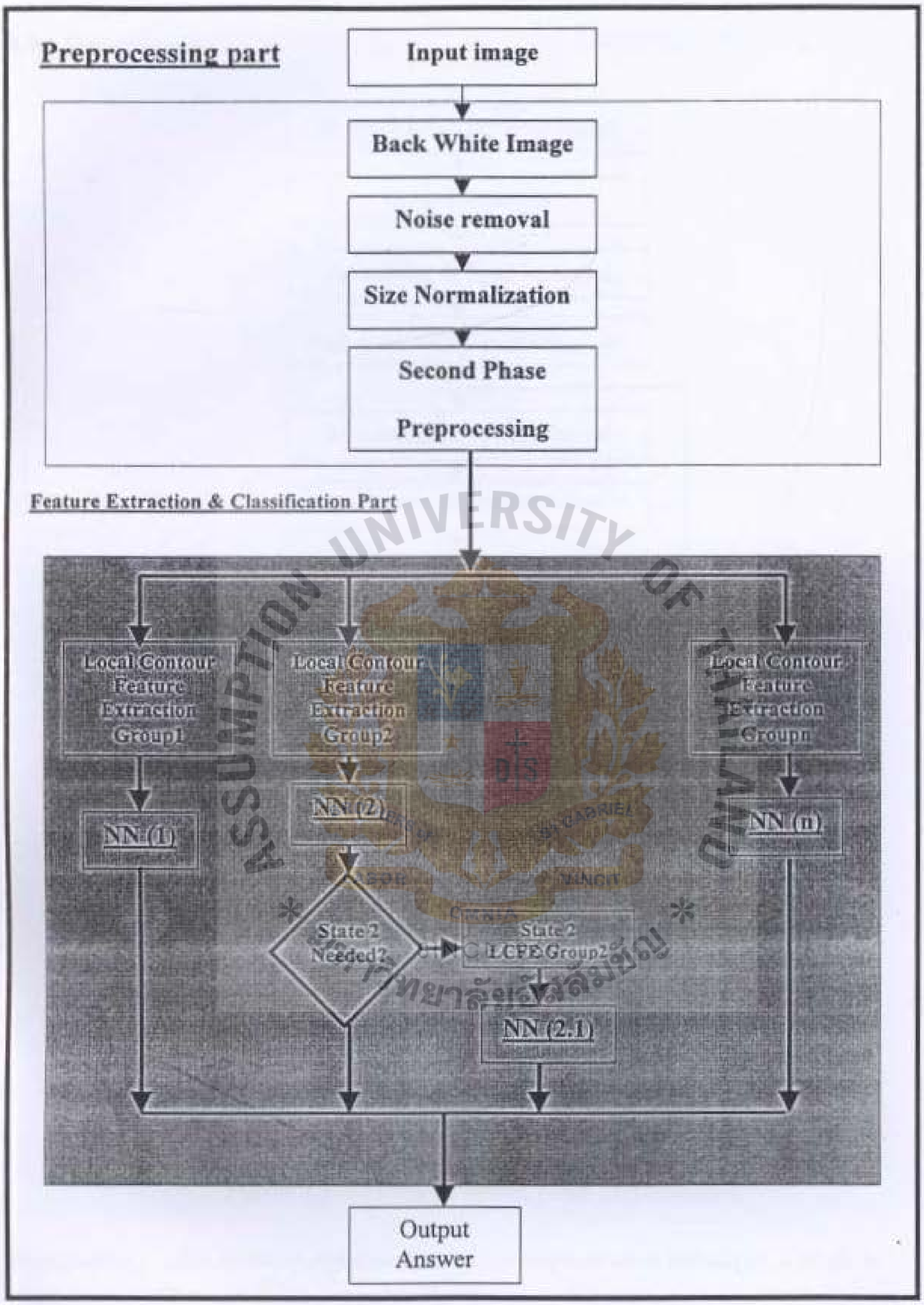


Figure 3.2 System Architecture

3.4.1 Preprocessing

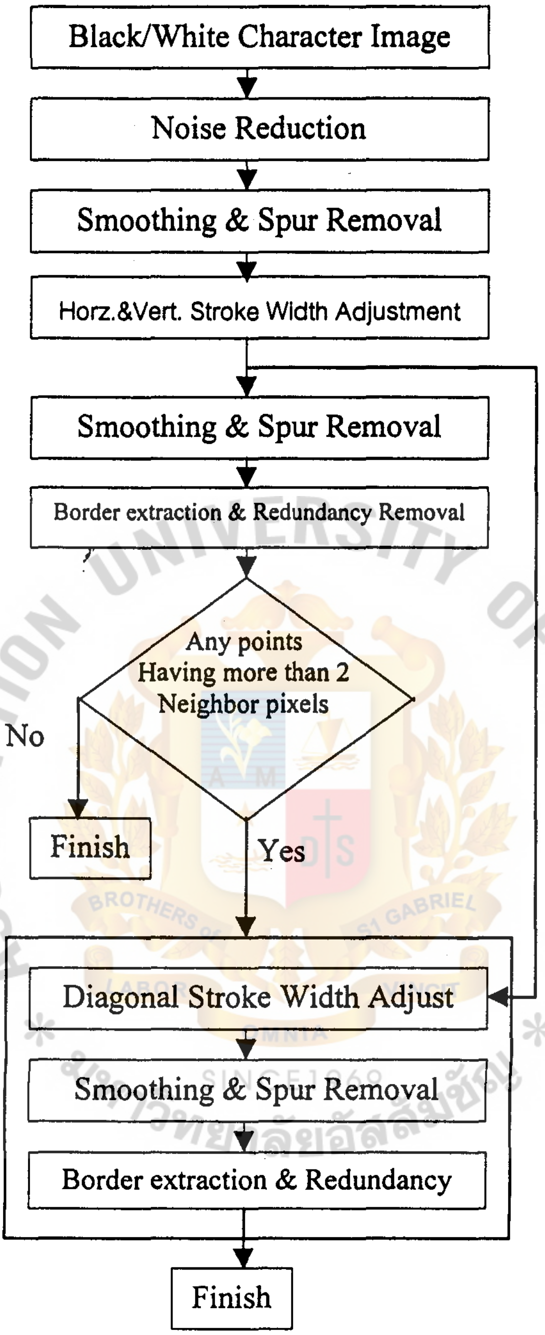

Black/White Character Image
Noise Reduction
Smoothing & Spur Removal
Horz.&Vert. Stroke Width Adjustment
Smoothing & Spur Removal
Border extraction & Redundancy Removal
Any points Having more than 2 Neighbor pixels
No
Finish
Yes
Diagonal Stroke Width Adjust
Smoothing & Spur Removal
Border extraction & Redundancy
Finish

***Figure 3.3** All steps of second phase preprocessing*

Preprocessing, after isolating alphabets out by any segmentation technique, it needs to preprocess before continuing in order to get the best classification result. We can separate the preprocessing into two stages. The first stage is the normal preprocessing like other technique and the second stage is the preprocessing before applying Local Contour technique that is designed especially to Local Contour technique.

For the normal preprocessing, first we convert input image into Back/White image, after that we do Noise removal that is cleaning which get rid of a number of small blob and Size normalization to guarantee input data is the same size with our Local Contour Data.

Next, Input character must pass the second preprocessing which are Noise removal, Smoothing & Spur Removal, Stroke Width Adjustment, Stroke Width Adjustment, Border extraction and Redundancy Removal step. Main purpose of the second preprocessing step applies with Local Contour technique. Each preprocessing step has its own features which are described below

- **Noise removal**: Like in the first proposing stage, it may be skipped if we surely pass it. This is the first step of removing noise like the point(s) alone that do not have any neighbor. The small blobs of black pixel are then removed.

- **Smoothing & Spur Removal**: It goal is to smoothen the character image by sliding 3x3 local window sized through every black pixel of image. Figure 3.4 shows the results after smoothing and spur removal is applied.

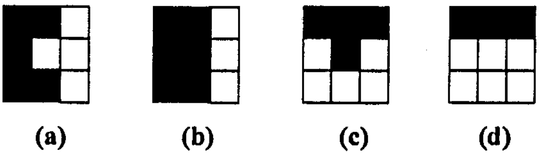


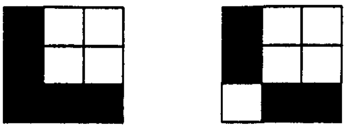
Figure 3.4 Example of 3x3 window size
(a) Before smoothing (b) After smoothing
(c) Before spur removal (d) After spur

- **Stroke Width Adjustment**: In [4], the black run length in both horizontal and vertical is guaranteed to be at least 4 pixels. This is the key to obtain the perfect contour. However, both directions are not enough in this application and therefore we added both diagonals into consideration. To minimize the processing time, the diagonal stroke width adjustment is applied only after failure on horizontal and vertical stroke width adjustment.

- **Border extraction**: After this process, only outer and inner(s) contour are left. Only outer contour will be analyzed as it contains sufficient information for recognizing important detail feature.

- **Redundancy Removal**: For contour analysis, this step is required to remove redundant pixels, which cause confusion in the contour tracking.

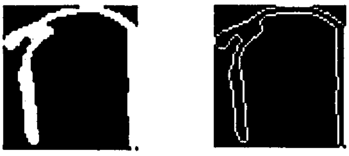
Figure 3.5 shows some image examples of redundancy removal. The resultant image is shown in Figure 3.6



(a) (b)

Figure 3.5 (a) 3x3 local window

(b) after redundancy removal applied



(a) (b)

Figure 3.6 Thai alphabet image “ก” (a) normal

character and (b) The resultant after preprocessing

3.4.2 Feature extraction

We propose the new feature extraction “ Local Contour analysis Technique” which will be fully described later in Chapter 4.

3.4.3 Classifier

We decided to use back-propagation neural network because it was proved that it was able to classify and recognize any input pattern. However, we need some times for collecting data, training and turning neural network. In some groups, data may need more than one neural network because some Thai alphabets look very similar so that we can distinguish only by a small feature in some area.

We use 20 back-propagation multi-layered perceptron neural networks to recognize and classify the alphabets into 11 groups. The input layer and hidden layer have varied number of neurons. Only two neuron nodes are used in the output layer of all networks. The hierarchical neural network with one local region of interest is organized to recognize Thai alphabet as shown in Figure 3.2. For example, in group no. 2, in the step 1 we must subgroup the alphabet (ข with ฃ) among the others in the group first and we will refine more later in the step 2.

CHAPTER 4

LOCAL CONTOUR ANALYSIS

4.1 Introduction of Local Contour Analysis Technique

We propose a new approach to do the contour analysis. However, we are only interested in the feature obtained from the local contour instead of the global contour. The fuzzy membership function interprets the structural information such as direction among the sampling points from the alphabet boundary into the input vector before passing it through the hierarchical neural network as the final recognizer.

Feature Extraction

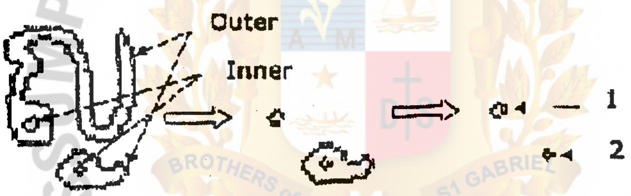


Figure 4.1 Thai alphabet image "ญ" with Inner and Outer Contour

4.2 Inner and Outer contour

In the same way as [2] and [5], only outer contour is considered. However, one difference between our approach and techniques in [2] and [5] is that our approach uses the local contour information rather than the global contour. The global contour obtained from the alphabet image of the same group such as in the group 1 tends to make the classifier easily confused. The rough classification simplifies the choice of local region that contains critical features.

4.3 Interesting Area

We use image size 64x64 (x-axis, y-axis) from Figure 3.1 we have 12 main group and 9 subgroups because in many cases, Thai alphabet can be distinguished only by checking the style of head(s), tail or notch. For example, the group no. 1's Thai alphabet "ก", "ถ", and "ถ".

```
Algorithm predefined_path_determination(T,P,StartPt)
%
% Constraint:
% StartPt  ∈ S where S = {$S_{tcg}$, $S_{lt}$, $S_{lb}$, $S_{rb}$}
%

% Starting point
If P = StartPt
   i = 1
   $C_i$ = StartPt
   $T_{startpt}$ = 2
   i = 2
   P = {$N_j$ | $min_j(N_j = 1)$ }
   $C_i$ = P
   $T_P$ = 2
Else
   i = 1
End

% From starting point
Finished = False
Do While Finished = False
If $\Sigma_{i=\{1,2,...,8\}} N_i = 0$
    Finished = True
ElseIf $\Sigma_{i=\{1,2,...,8\}} N_i = 1$
    i = i + 1
    P = {$N_j$ | $N_j$ = 1}
    $C_i$ = P
    $T_P$ = 2
ElseIf $\Sigma_{i=\{1,2,...,8\}} N_i = 2$
   $T_1$ = T
   $T_2$ = T
   Pt = {Nj | Nj = 1}
   $C_1$ = predefined_path_determination($T_1$,$Pt_1$,StartPt)
   $C_2$ = predefine d_path_determination($T_2$,$Pt_2$,StartPt)
   If size(C1) > size(C2)
       P = $Pt_1$
   Else
       P = $Pt_2$
   End
    i = i + 1
    P = {$N_j$ | $N_j$ = 1}
    $C_i$ = P
     $T_P$ = 2
End
Return C
```

Listing 1: Predefined contour

These 3 alphabets' structure is very similar. The only difference is in the style of the head as shown in Figure 4.2

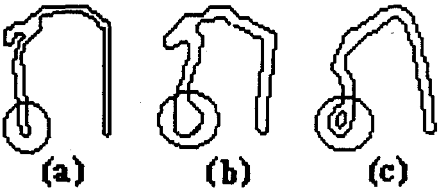


Figure 4.2 Sample of closed similar Thai alphabets (a) is "ก" (b) is "ถ"and (c) is "ภ".

Therefore, the region of interest would be focused only in certain position that is dependent upon the character group. Figure 4.3 shows the cut-part region of interest for group 1 recognition.

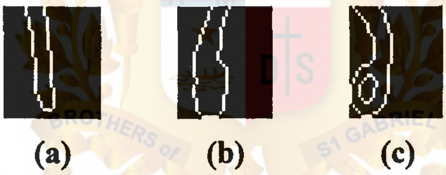


Figure 4.3 Region of interest for group 1(Left-Bottom Region) recognition. (a) Alphabet "ก" (b) Alphabet "ถ" and (c) Alphabet "ภ".

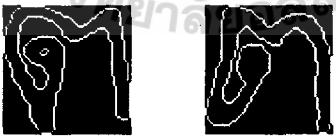


Figure 4.4 Same character "ฑ "in different style of writing can make mistake when finding the starting point.

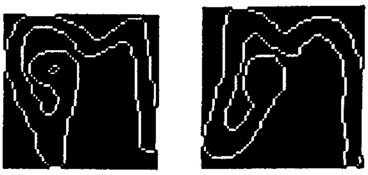


Figure 4.5: Starting point using Centre of Gravity and most top.

4.4 Starting Point

The starting point plays an important role in the contour traversing technique. The shift in starting point would affect misclassification because the order of data will be shifted. Therefore, it is very important to find consistent starting point by using the point at Center of Gravity [12], most Top-Left, most Top-Right, most Bottom-Left, or most Bottom-Right varied among the groups as shown in Figure 4.4 and Figure 4.5, by applying Center of Gravity and most top for starting point.

4.5 Travel contour algorithm

From the starting point, the boundary-tracking path must be determined to establish the *predefined contour*, from which the sampling points can be chosen. We do not use all of them because it will be sensitive to the noise. The predefined contour is the list of contour points ordering from the starting point to the ending point (one of neighbor points of the starting point) where the path length is the longest. Algorithm listed in Listing 1 is used to find the path where

- P is the current point position (x,y). StartPt is starting point and is the member of Starting set *S*.
- S is set of all possible points of starting point used in our approach and has St_{cg}, S_{lb}, S_{rt}, and S_{rb} as member where St_{cg} is most top point of CG position,

S_{lt} be the most left-top point of image. S_{lb} be the most left-bottom point. S_{rt} is most right-top point and S_{rb} is most right-bottom point.

- *C is list of predefined contour position ordered from the starting point to the* end point obtained from the algorithm in the Listing 1.
- T is an *mxk* black/white image already passed through the preprocessing step.
- N_1, N_2, N3, N_4, N_5, N_6, N_7, and N_8 are adjacent pixels of point P and defined as in Figure 4.6.

	X-1	**X**	**X+1**
Y-1	N_8	N_1	N_2
Y	N_7	P	N_3
Y+1	N_6	N_5	N_4

Figure 4.6 Neighbor of pixel P

4.6 Sampling point

The sampling point is defined as a point on the character boundary, which holds the following properties:

1. It is the starting point.
2. It lies on the distance D adjacent to last sampling point in traversal order as follows:

```
case $l < N-1$,  D is undefined. The sampling point
selection is failed.
  case $l$ is multiple of N-1,   $D = l/(N-1)$
  otherwise,
      $R = (l \bmod N-1)$
      if $R \neq 0$ and $R > 3(N-1)/5$ then
          $D = R$
      Else
          $D = l$ div N-1
      End
```

where l is the length of predefined contour, N is the number of desired sampling point, and D is the distance threshold. Figure 4.7 shows the example of sampling point obtained from predefined contour of the alphabet image “ส”

4.7 Information Retrieval

The input vector X is formed by the relative angular direction between

1) Starting point and the rest of sampling points
2) The consecutive sampling points.

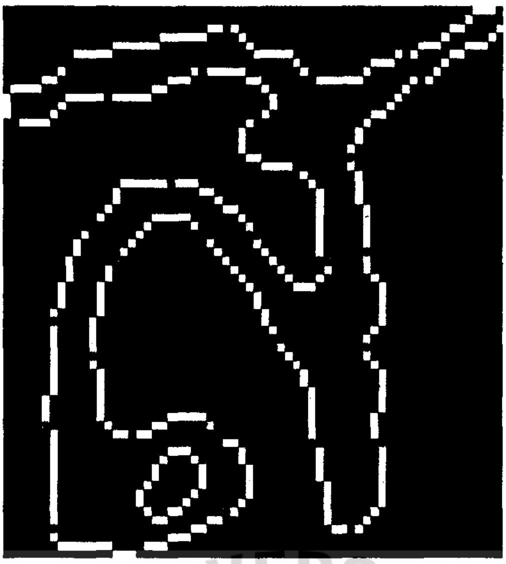


Figure 4.7 Thai alphabet contour "ส" with the gray boundary points represent the sampling points.

4.8 Fuzzy membership of Left, Right, Top, Down

The direction between two points would be assigned to the fuzzy membership function [13] shown in Figure 4.8. It interprets the degree of 4 directions: Right, Left, Top, and Down, from the angle. Therefore, the input vector will have dimension of 1x (4x*n*) for direction in 1) and 1x(4x(*n*-1)) where n is sampling point size.

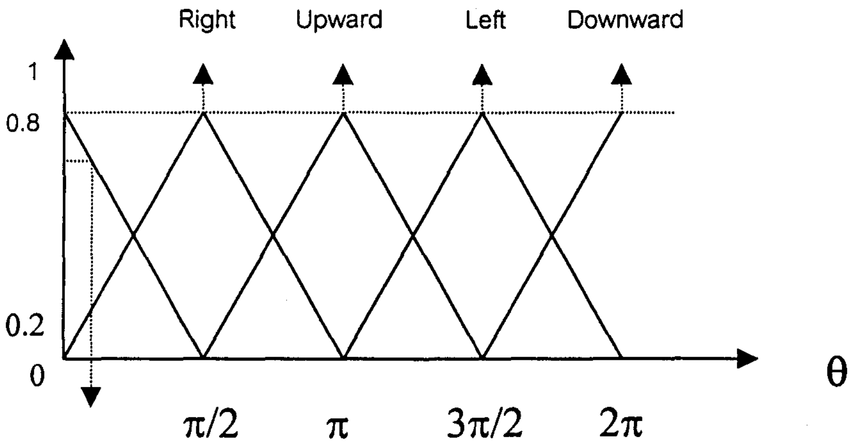


Figure 4.8 Fuzzy membership function of degree in 4 directions: Right, Left, Upward, and Downward.

CHAPTER 5

NEURAL NETWORK

5.1 Neural Network

The field of neural networks can be thought of as being related to artificial intelligence, machine learning, parallel processing, statistics, and other fields. The attraction of neural networks is that they are best suited to solving the problems that are the most difficult to solve by traditional computational methods.

Consider an image processing task such as recognizing an everyday object projected against a background of other objects. This is a task that even a small child's brain can solve in a few tenths of a second. But building a conventional serial machine to perform as well is incredibly complex. However, that same child might not be capable of calculating 2+2=4, while the serial machine solves it in a few nanoseconds.

A fundamental difference between the image recognition problem and the addition problem is that the former is best solved in a parallel fashion, while simple mathematics is best done serially. Neurobiologists believe that the brain is similar to a massively parallel analog computer, containing about 10^10 simple processors which each require a few milliseconds to respond to input. With neural network technology, we can use parallel processing methods to solve some real-world problems where it is very difficult to define a conventional algorithm.

Neural network is one kind of most popular recognition system. It is widely used for the pattern recognition area. Neural network has many types and the one that we decided to use is back propagation. Because we have 12 groups and 11 groups need to be distinguished, total numbers of neural network that we use are 21 numbers. The reason why we must use 21 neural networks is already explained in chapter 3.

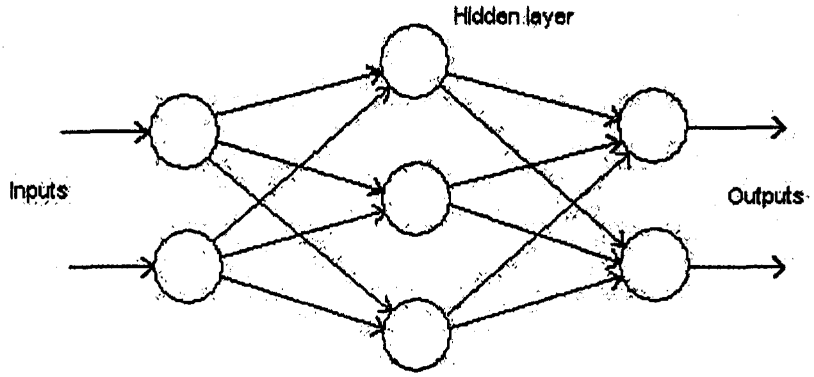


Figure 5.1 The structure of Network

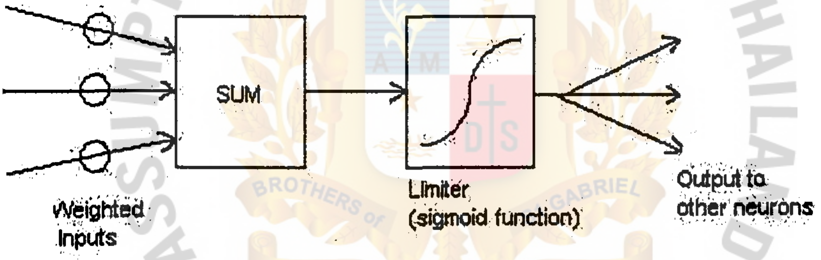


Figure 5.2 The Structure of a Neuron

Figure 5.1 The structure of Network: Stimulation is applied to the inputs of the first layer, and signals propagate through the middle (hidden) layer(s) to the output layer. In Figure 5.2: The Structure of Neuron: Inputs from one or more previous neurons are individually weighted, then summed. The result is non-linearly scaled between 0 and +1, and the output value is passed on to the neurons in the next layer.

5.2 Back propagation model

If we consider the human brain to be the 'ultimate' neural network, then ideally we would like to build a device which imitates the brain's functions. However, because of limits in our technology, we must settle for a much simpler design. The obvious approach is to design a small electronic device which has a transfer function similar to a biological neuron, and then connect each neuron to many other neurons, using RLC networks to imitate the dendrites, axons, and synapses. This type of electronic model is still rather complex to implement, and we may have difficulty 'teaching' the network to do anything useful. Further constraints are needed to make the design more manageable. First, we change the connectivity between the neurons so that they are in distinct layers, such that each neuron in one layer is connected to every neuron in the next layer. Further, we define that signals flow only in one direction across the network, and we simplify the neuron and synapse design to behave as analog comparators being driven by the other neurons through simple resistors. We now have a feed-forward neural network model that may actually be practical to build and use.

Referring to Figures 5.1 and 5.2, the network functions as follows: each neuron receives a signal from the neurons in the previous layer, and each of those signals is multiplied by a separate weight value. The weighted inputs are summed, and passed through a limiting function which scales the output to a fixed range of values. The output of the limiter is then broadcast to all of the neurons in the next layer. So, to use the network to solve a problem, we apply the input values to the inputs of the first layer, allow the signals to propagate through the network, and read the output values.

Since the real uniqueness or 'intelligence' of the network exists in the values of the weights between neurons, we need a method of adjusting the weights to solve a particular problem. For this type of network, the most common learning algorithm is called Back Propagation (BP). A BP network learns by example, that is, we must provide a learning set that consists of some input examples and the known-correct output for each case. So, we use these input-output examples to show the network what type of behavior is expected, and the BP algorithm allows the network to adapt.

The BP learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random). This output is compared to the known-good output, and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point we say that the network has learned the problem "well enough" - the network will never exactly learn the ideal function, but rather it will asymptotically approach the ideal function.

The neural network model originally consists of multiple layers. In general, the decision region required by any classification method can be generated by a three layer feed forward network input, hidden, and output layer. Each layer consists of a number of cells called units or node. Each element of input vector corresponds to a simple unit in the input layer. The output signal from the output layer is equivalent to the output of discriminated function. Each unit is connected to all units in the layer above its own. For example, a unit in the input layer will be connected to all units in the hidden layer. Each connection has an unbounded positive or negative weight associated with it. The output of the network is a function of the inputs and the weights.

CHAPTER 6

EXPERIMENT and RESULT

This chapter will discuss "Experiment environment", that is about equipment, program and database for this research. In section 6.2, a discussion is made of the preprocessing step of the input image. Section 6.3 is about classifier part, and what kind of recognition engine that we use. Section 6.4 contain the dataset information. In the last section, the result is given.

6.1 Experiment environment

SCANNER: Used for collecting input data. We use 3 different brands

- Brand: Pacific Image model 1236S
- Brand: Colorado model 19200
- Brand: Hewlett Packard model OfficeJet 1150c

SCAN MODE: 300 dpi resolutions on gray mode scale (256 levels)

COMPUTER: PC-Pentium III 600 MHz CPU with Windows 98 operating system.

SOFTWARE TOOL: MATLAB®5.3, The MathWorks, Inc. We use this tool because it is a popular tool for doing computation and programming in an easy way.

6.2 Image Preprocessing

After we get the data image from scanning, we have a few steps before sending it to the system. First, because our goal does not focus on cutting images isolation, other tools such as Photo Shop, Paint Brush are used as substitutes for cutting each alphabet out. After we get a file from scanning (TIFF format), we convert the pictures format from TIFF to JPEG. The reason of using JPEG is that the size of the file is small but it still maintains the quality.

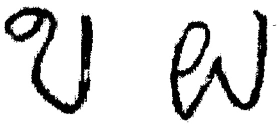


Figure 6.1: Example of scanned alphabet "ข" and "ผ"

Binarization: After converting to JPG format, we get the data that have 256 level (gray-scale). It is needed to convert the gray-scale image into binary image (back-white). Our experiment used global threshold method. The importance of this step is how to specify the global threshold value. If the intensity of each pixel is over the threshold, it will change to black. On the other hand, if that pixel is less than the threshold value, it will change to white. Our experiment used threshold value=0.8 in all data set.

Blob Remove: Next, we apply removal blob because after changing image to black-white, there is no technique that guarantees the noise-free output. In our experiment, if blob is less then 20 pixels in size, it will be removed.

Resize Image: We had to resize image because if the input does not have the same image size, it may cause miss recognition when entering the system. The size of image that we use is 64 X 64.

6.3 Classifier

We use neural network as a tool for classifier. Because it is proved that it is able to classify and recognize any input pattern. It is also widely used in the pattern recognition area. We use three layers back-propagation neural networks in our system.

Each group number of neural network is not the same as shown in Figure 3.1. For example, on group 1 (ก, ภ, ถ) we use only one back-propagation neural network. On the other hand, on group 2 (ข, ซ, บ, ย, ษ), it is needed to have two neural networks. First to distinguish บ, ย, ษ and **ข** with ซ, and another one to distinguish **ข** and **ซ**. The input layer and hidden layer have varied neural units. The number of output layer is also varied. It depends on the group already discussed in Chapter 3.

6.4 Data Set

We use different styles of data set collected from 110 people; one person writes only one data set (44 alphabets). Therefore, each character has 110 different data. We separate 4,840 input images.

Every group has 2 main sets; each set has equal sizes. We use two sets for training First set for training data set and one set for validation set and another two sets for testing as unseen data. In each group the dataset has not the same size. For example in group 1 has 3 members then all the input data of this group is 3x110= 330 input images.

Table 6.1: Interest area of classification, number of sampling point and starting point position

Group No.	Group character	Number of Sampling point	Starting point position	Area Interested	
				X – axis	Y - axis
1	ก ถ ภ	15	Top-Left	32-64	32-64
2	ข ฃ บ ย ษ	28	CG	1-64	1-64
2.1	ข ฃ	13	Bottom-Left	1-32	1-12
3	ค ด ฅ ต ศ	20	Bottom-Left	1-64	1-64
3.1	ค ด	28	CG	1-64	1-64
3.2	ฅ ต	15	Top-Left	1-40	20-64
4	ฆ ม	13	Bottom-Left	1-32	1-12
5	ง ว อ	30	CG	1-64	1-64
6	จ ฉ ล ส	30	CG	1-64	1-64
7	ฌ ญ ฒ ณ	28	CG	1-64	1-64
8	ฎ ฏ	15	Top-Right	1-64	30-64
9	ฑ ท น ห	28	CG	1-64	1-64
9.1	ฑ ท	11	Bottom-Left	1-20	1-12
9.2	น ห	32	CG	1-64	1-64
10	ช ซ ฐ ร ธ ฮ	30	CG	1-64	1-64
10.1	ช ซ	13	Top-Right	1-32	1-22
10.2	ฐ ร	32	CG	1-64	1-64
10.3	ธ ฮ	32	CG	1-64	1-64
11	ป ผ ฝ พ ฟ	35	CG	1-64	1-64
11.1	ผ ฝ	Using fuzzy determine tail			
11.2	พ ฟ	Using fuzzy determine tail			

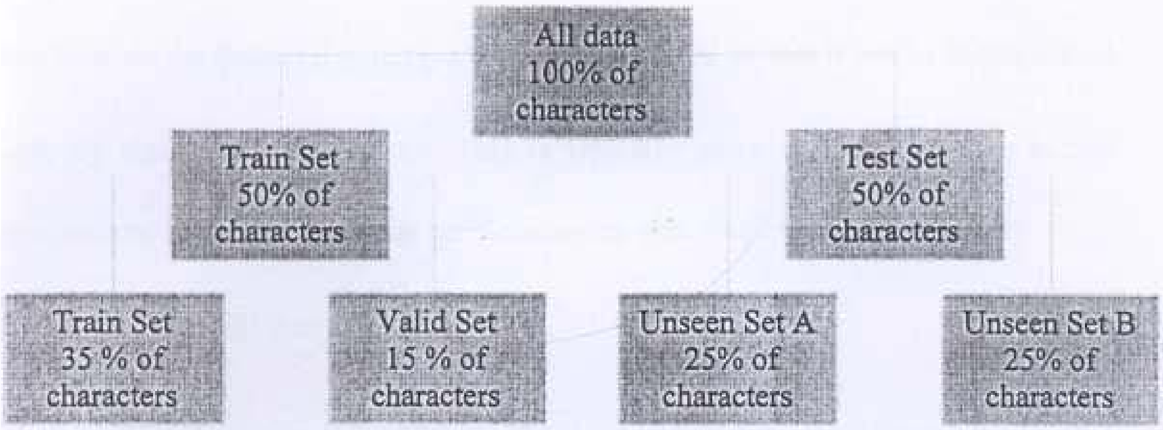


Figure 6.2: Image character organization of every group

6.5 Experiment Result

We separate testing set to group number 1 and 2. Each set has the same size. The result is shown on table 6.2 .

Table 6.1: Percent Correct of unseen Test set by Local Contour Technique (2,420 alphabets)

Group NO.	Group Alphabet	%Correct Test Data Group 1	% Correct Test Data Group 2	Average % Correct Group 1 & 2
1	ก ถ ภ	86.20	82.15	84.18
2	ข ฃ บ ย ษ	93.89	90.89	92.39
3	ค ฅ ด ต ศ	87.1	88.9	88
4	ม ฆ	88.53	86.49	87.51
5	ง ว อ	97.58	98.2	97.89
6	จ ฉ ถ ส	94.11	95.98	95.05
7	ฌ ญ ฒ ณ	93.86	91.95	92.91
8	ฎ ฏ	86.67	88.77	87.72
9	ฑ ท น ห	85.1	86.2	85.65
10	ซ ช ฐ ธ ร ฮ	86.0	87.51	86.76
11	ป ผ ฝ พ ฟ	92.47	91.52	92
12	ฬ	100	100	100
	Average on group	90.96	90.72	90.84

From the experiment result, we found that on group 8, the recognition rate is quite low because the character ฎ and ฏ are extremely similar so that it can be distinguished only by their notch on the tail. This is typically quite difficult even for human recognition. In group 10, similar problem occurs with ข and ฃ in which the only difference is a notch near the head.

The standard deviation is a measure of how widely values are dispersed from the average value (the mean). We use “nonbiased” or “n-1” method. We can calculate SD by the following formula

$$SD = \sqrt{\frac{(n\sum x^2 - \left(\sum x^2\right)}{n(n-1)}}$$

The standard deviation of our experiment result of all characters is equal to 5.008678.

6.6 Limitation of Local Contour Analysis

The weak point of our technique is that it lacks ability to answer the input image that is not a character. If our system can perform like open system, it will make the overall performance better because it will be able to answer back to the first stage system that it misses classifier into a group and request the first stage to give another answer. We leave these topics for future research.

6.7 Future research & development guide

Now our system is a close system. Therefore, it cannot reject the input whether the input is in known set. In the future, we maybe able to extend our system to be the open system which can reject if the input is not alphabet.

Another recommendation for future research is on applying this technique to the first stage for rough classification. For our point of view, this technique can distinguish the set of characters that have different structures. From this reason, it should work on rough-classification and we should separate the new group which is no longer like [6]. However, we cannot guarantee the result. It might not work because, as already discussed, each technique has their weak and strong points.

The last one that we would like to see now is extending our system to classify all of the Thai Characters that are

- Vowel
  - Upper level(อัม, อำ, อิ, อี, อึ, อือ)
  - Lower level (อุ, อู)
  - Inline level (อะ, อา, เอ, แอ, ไอ, โอ, ใอ)
- Tone (อ่, อ้, อ๊, อ๋)
- Thai Number (๑, ๒, ๓, ๔, ๕, ๖, ๗, ๘, ๙, ๐)
- Special character (ฤ, ฦ, ๆ, ฯ, อ์, อ็)

It is challenging to find whether or not our idea will work with bigger data sets that cover all Thai characters.

CHAPTER 7

CONCLUSION

7.1 Conclusion

The experimental results have confirmed that our system can be linked to existing system [6] and work effectively in classifying Thai handwritten alphabets. Local contour analysis can perform quite well in capturing small detail features that distinguish similar characters.

With two-stage classifiers, there are several advantages: First, it has noise tolerance because the rough classifier focuses on the character's structure to separate the groups of character set. Second, the alphabet recognizer then has simpler search space. Only that kind of characters classified as being members in the group would be concerned.

BIBLOGRAPHY

[1] Choruengwiwat, Prasert. “Thai Handwritten Character Recognition using Extraction Of Distinctive Features,” *Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering, Chulalongkorn University,* 1998.

[2] D. Yu and H. Yan, “An Efficient Algorithm for Smoothing, Linearization and Detection of Structural Feature Points of Binary Image Contours” *Pattern Recognition, vol.30 no.1*, pp 57-60, 1997.

[3] Gwo-En, Jhing-Fa Wang. “A New Hierarchical Approcach for Recognition of Unconstrained Handwritten Numerals”, *IEEE Transactions on Consumers*, vol. 40(no. 3), August 1994.

[4] Jianming Hu, Donaggang Yu and Hong Yan, “Algorithm for stroke width compensation of handwritten characters” *IEEE Electronics Letters Online*, No 19961501, 1996.

[5] Jinhai, C. and Zhi-Qiang, L. “Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21(no. 3): pp. 263-270, March 1999.

[6] Jitwongtrakul, Kamolchol and Tanpraset, Thitipong. “Sample-Synthesized Simulated Light Sensitive Model for Structural Categorization of Handwritten Thai Character” *Proceeding of International Conference on Intelligent Technologies*, pp214-220, Dec 2000.

[7] Kimpan, Chom “Printed Thai character recognition using topological properties method,” *INT. JOURNAL ELECTRONICS*, pp. 303-329, vol. 60(no. 3), 1993

[8] K.Skul, " Handprinted Thai Character Recognition using K-L transform with fuzzy neural network", *Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering, King Mongkut's Institute of Technology Ladkrabang,* 1996.

[9] Panich, Wicha, "A Thai character recognition system based on distinctive features of Thai characters" *Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering, Chulalongkorn University,* 1996.

[10] Rattanathan, Deecha, "Thai printed characters recognition using the fuzzy logic techinique and syntactic method" *Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering ,Chulalongkorn University,* 1995.

[11] Sinthupinyo, Sukree, "An application of inductive logic programming and backpropagation neural network to Thai printed character recognition "*Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering, Chulalongkorn University,* 1998.

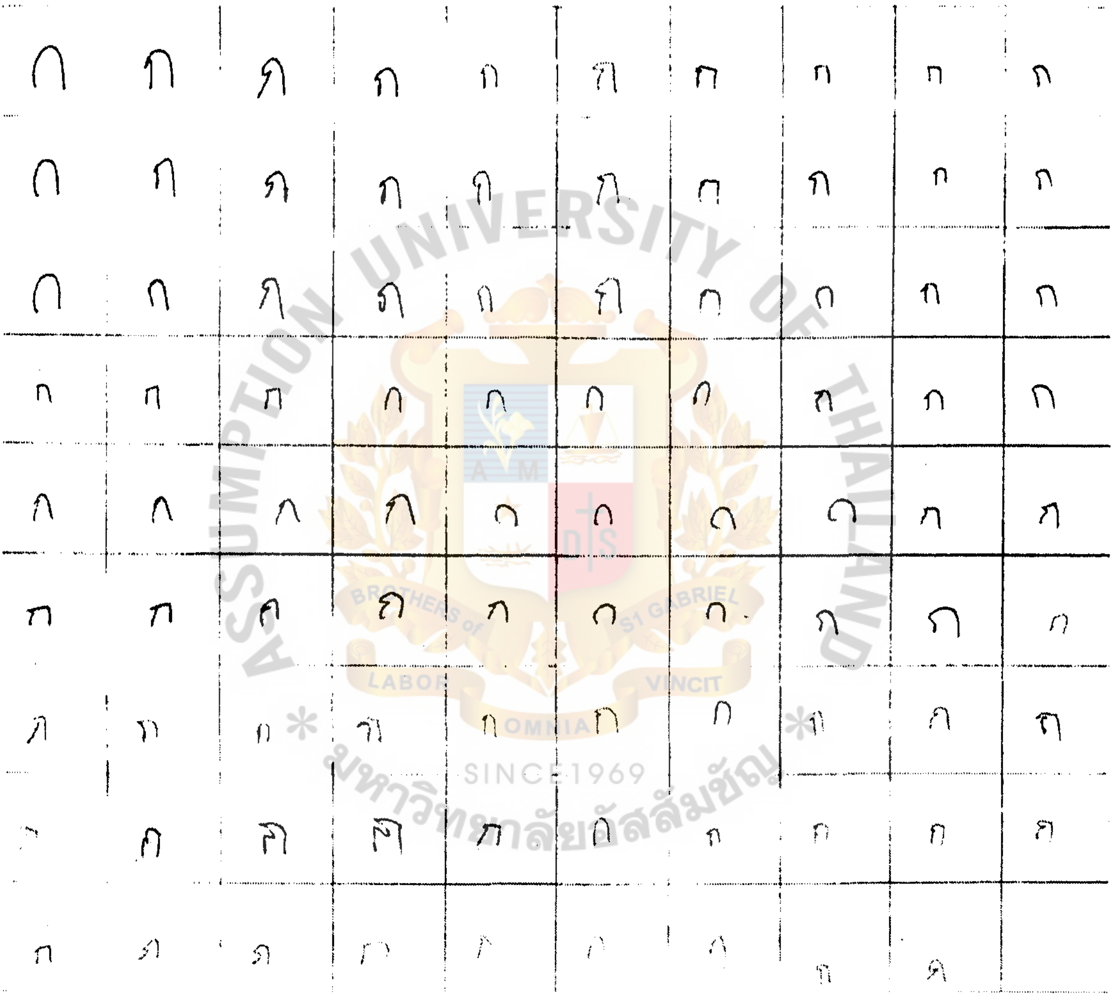
[12] Thumwarin, Pitak and Chittayasothorn Suphamit, " An Object-oriented Expert System for Thai Character Recognition "*IEEE: pp.153-156,* 1998.

[13] Toyooshi, Odahara and Junji, Nakanishi. "Handwritten Character Recognition using a Fuzzy associative Memory" *Progress in Connectionist-Based Information Systems,* pp 1161-1165, 1997.

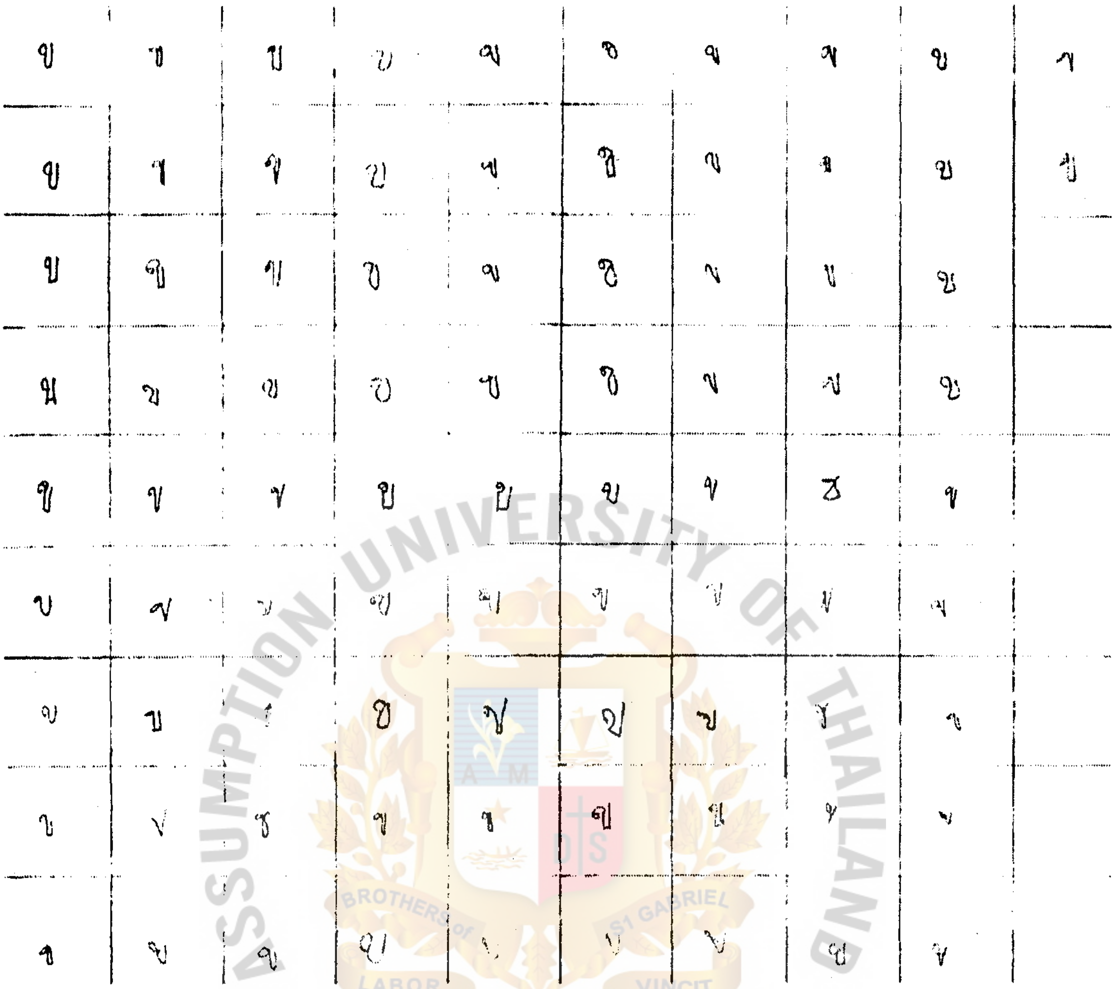
APPENDIX A: Examples of Handwritten images

In this section presents the examples data set of handwritten images from alphabet ก to ฮ total 44 characters.

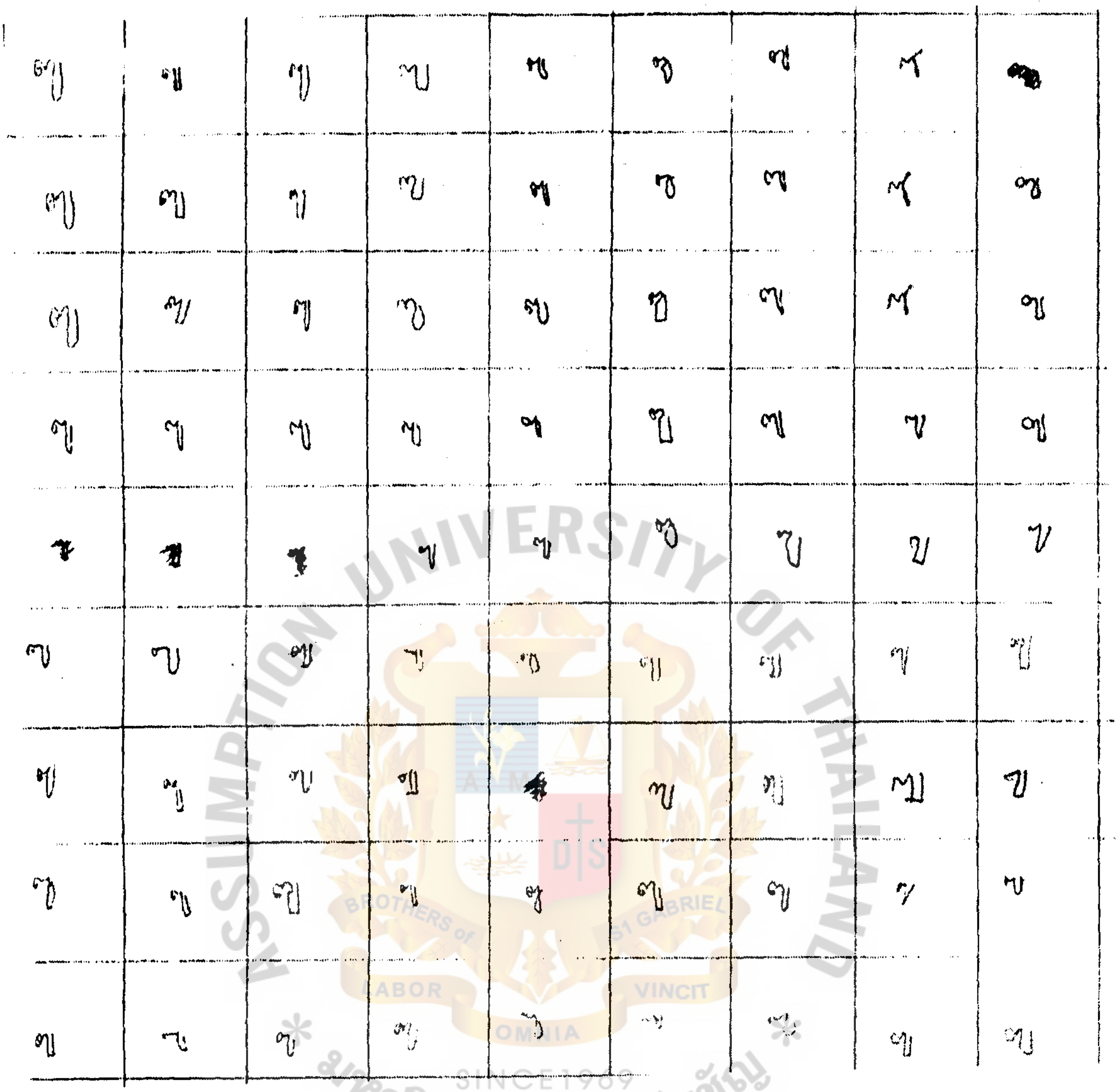
A.1 Alphabet ก



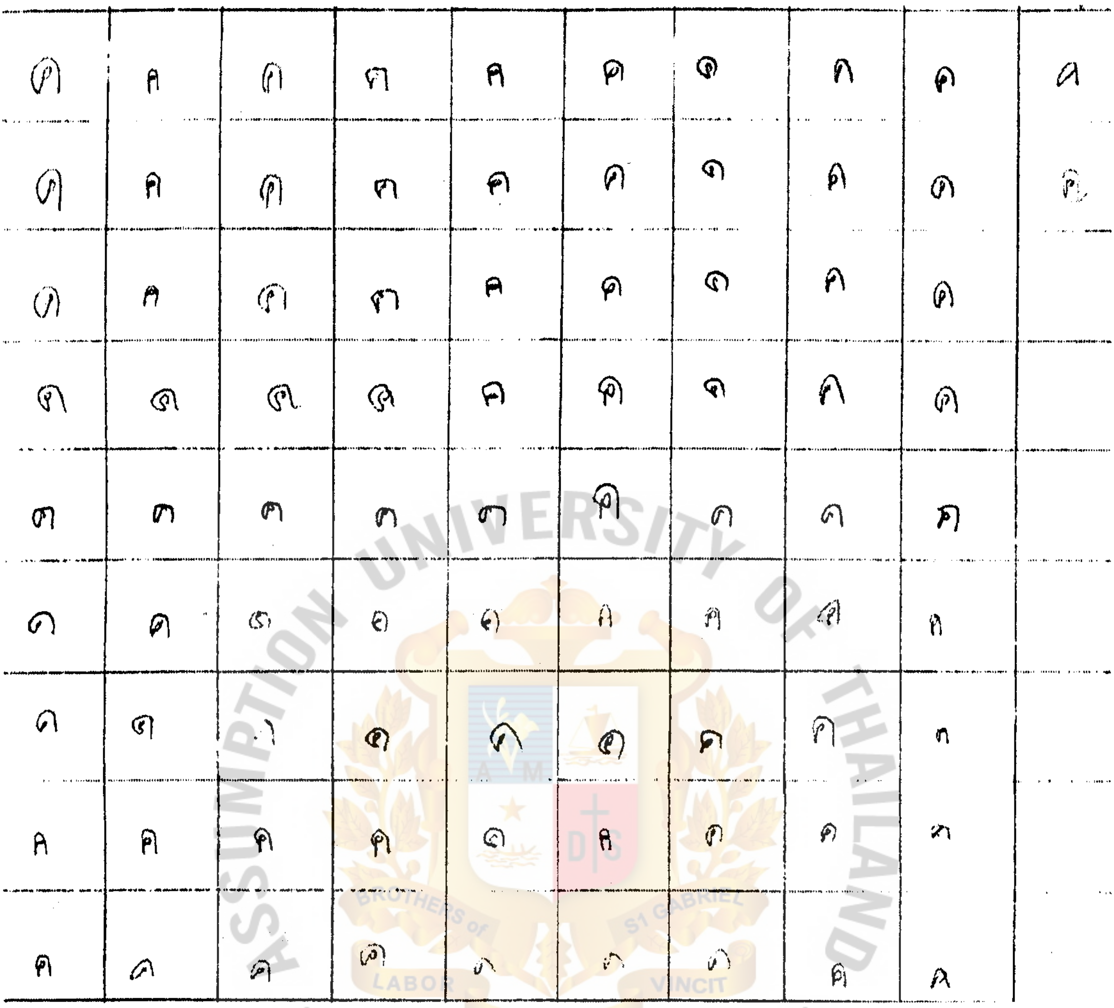
A.2 Alphabet ข



A.3 Alphabet ข



A.4 Alphabet ค



A.5 Alphabet ค

ค	ค	ค	ค	ค	ค	ค	ค	ค	ค
ค	ค	ค	ค	ค	ค	ค	ค	ค	ค
ค	ค	ค	ค	ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค	ค	ค	ค	ค	
ค	ค	ค	ค	ค	ค	ค	ค	ค	

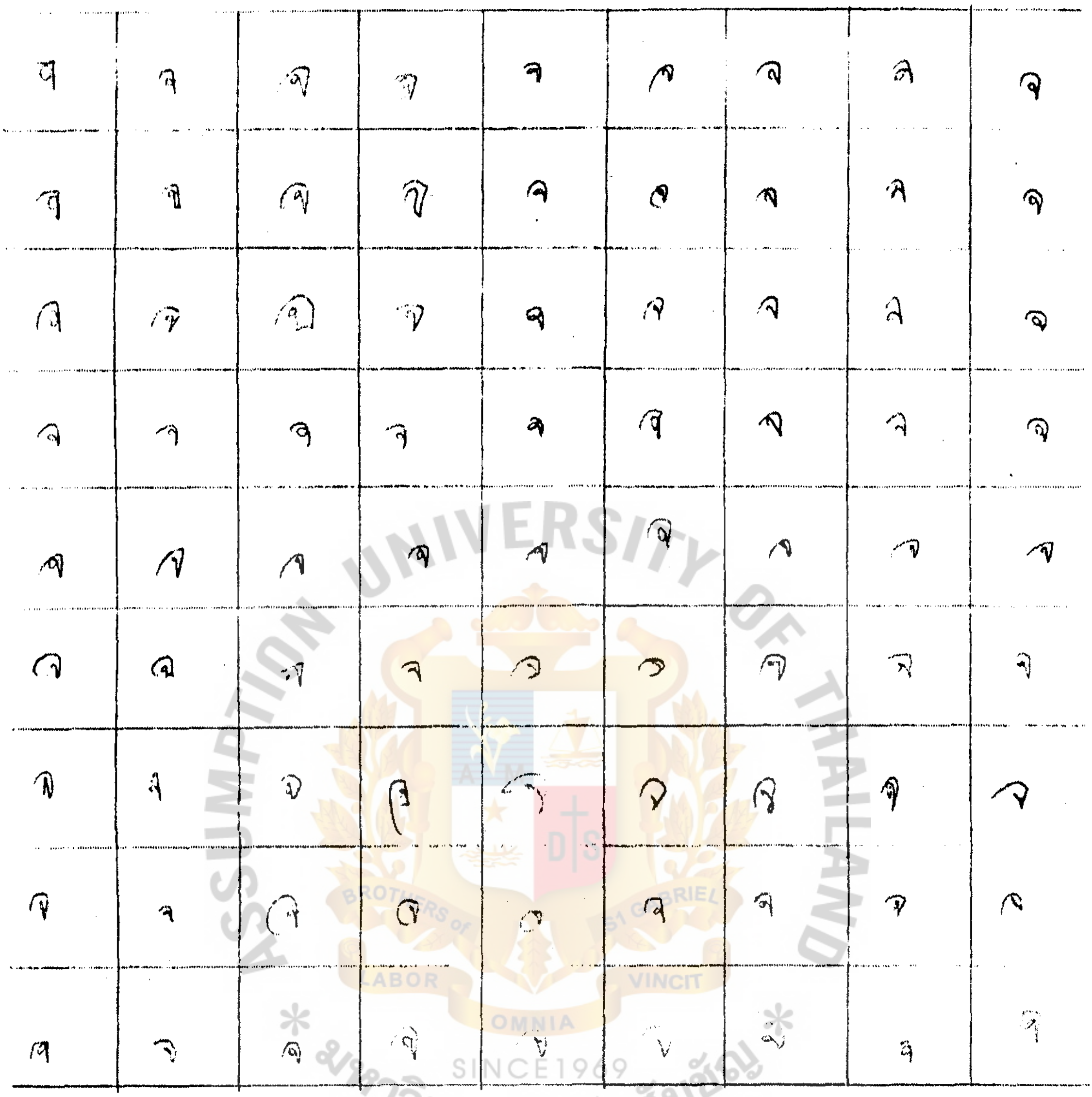
A.6 Alphabet ฆ

ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ

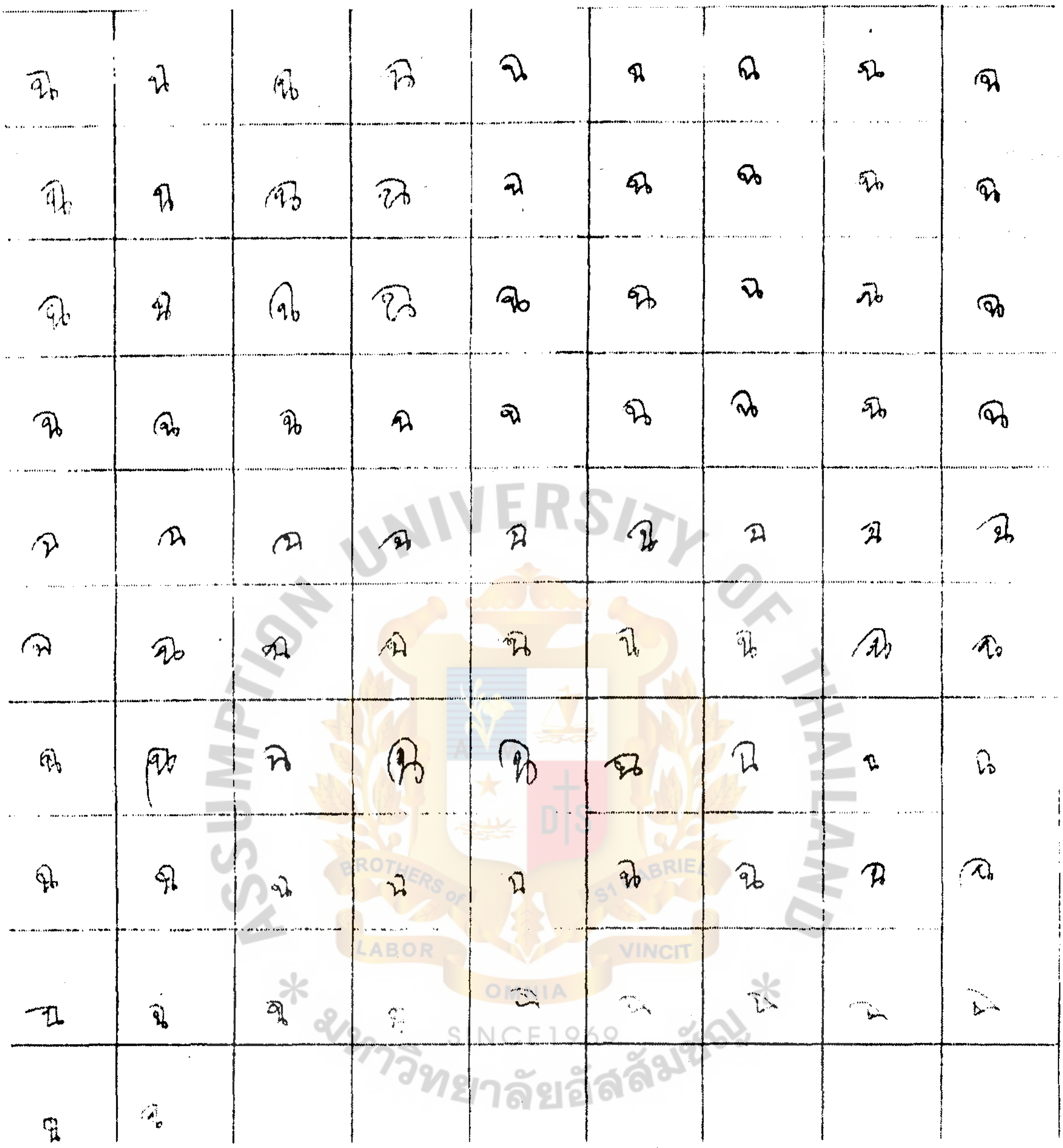
A.7 Alphabet ง

ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง	ง	ง	ง
ง	ง	ง	ง	ง	ง			

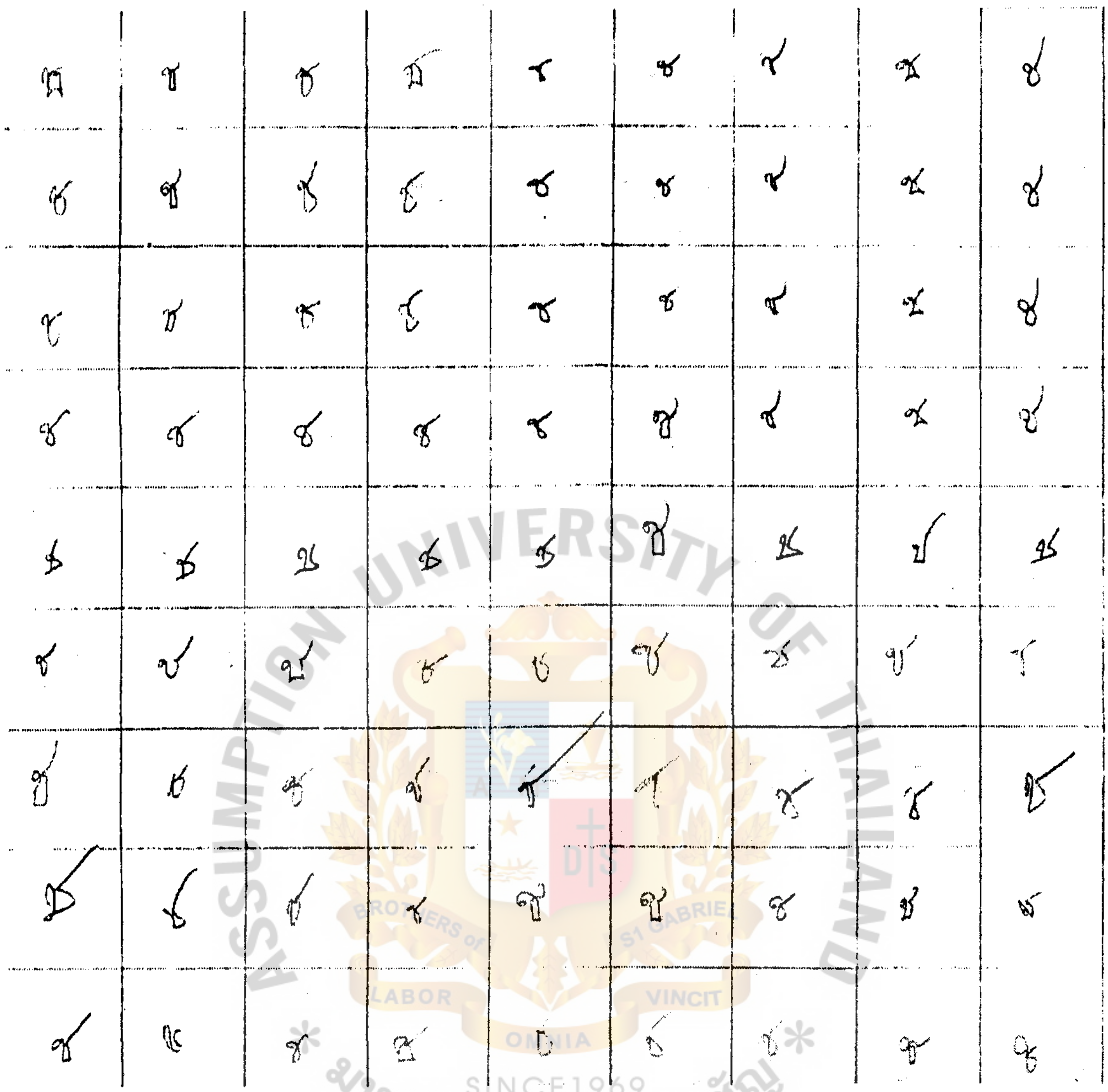
A.8 Alphabet จ



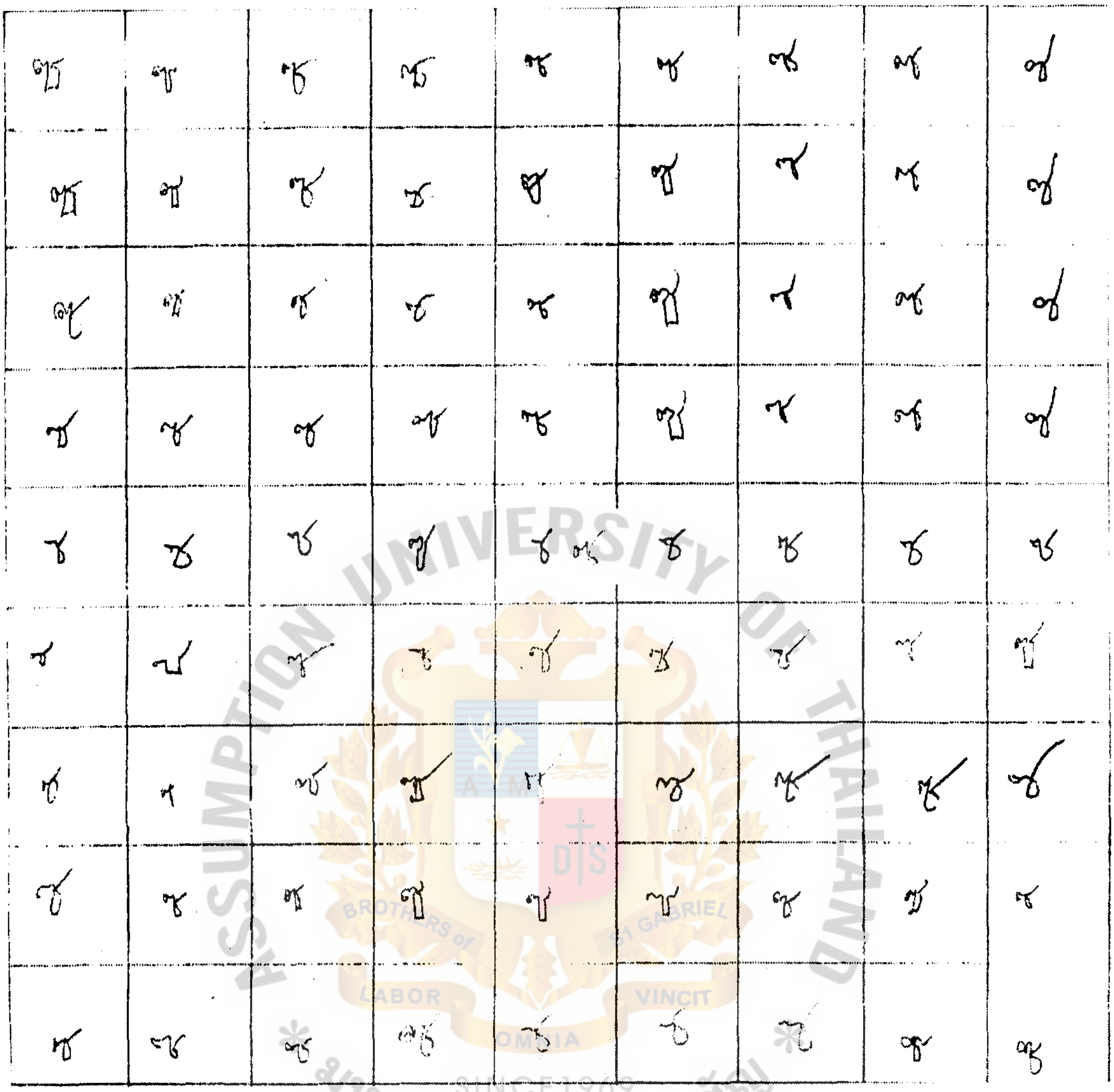
A.9 Alphabet ฌ



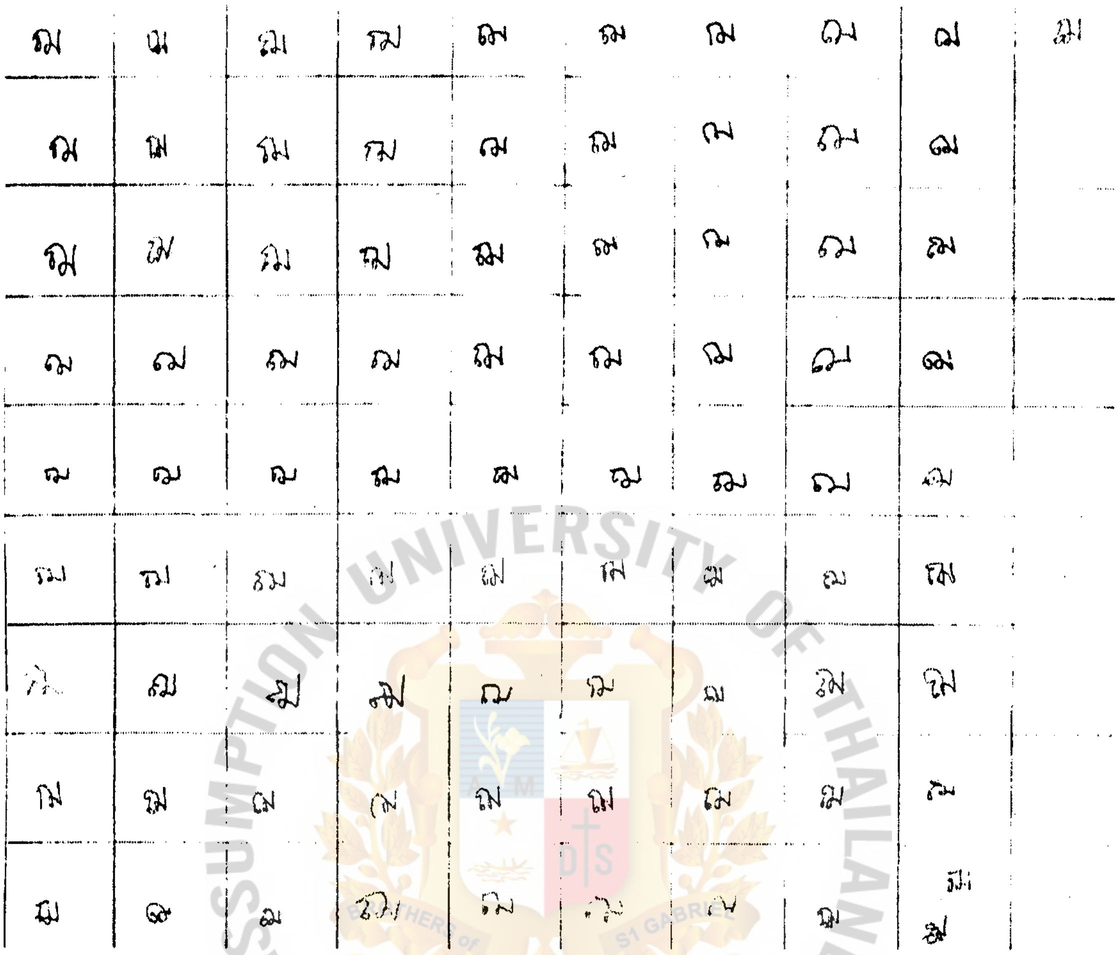
A.10 Alphabet ข



A.11 Alphabet ข



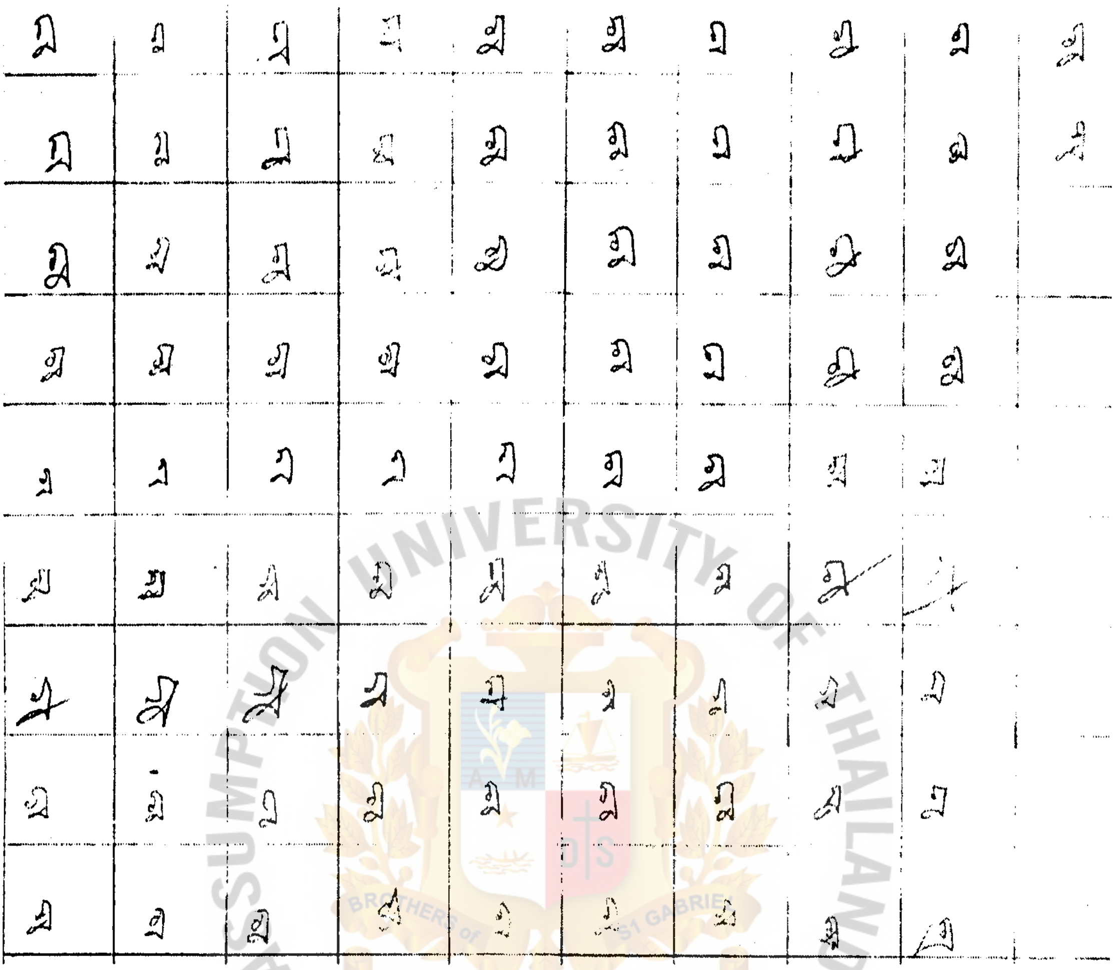
A.12 Alphabet ฌ



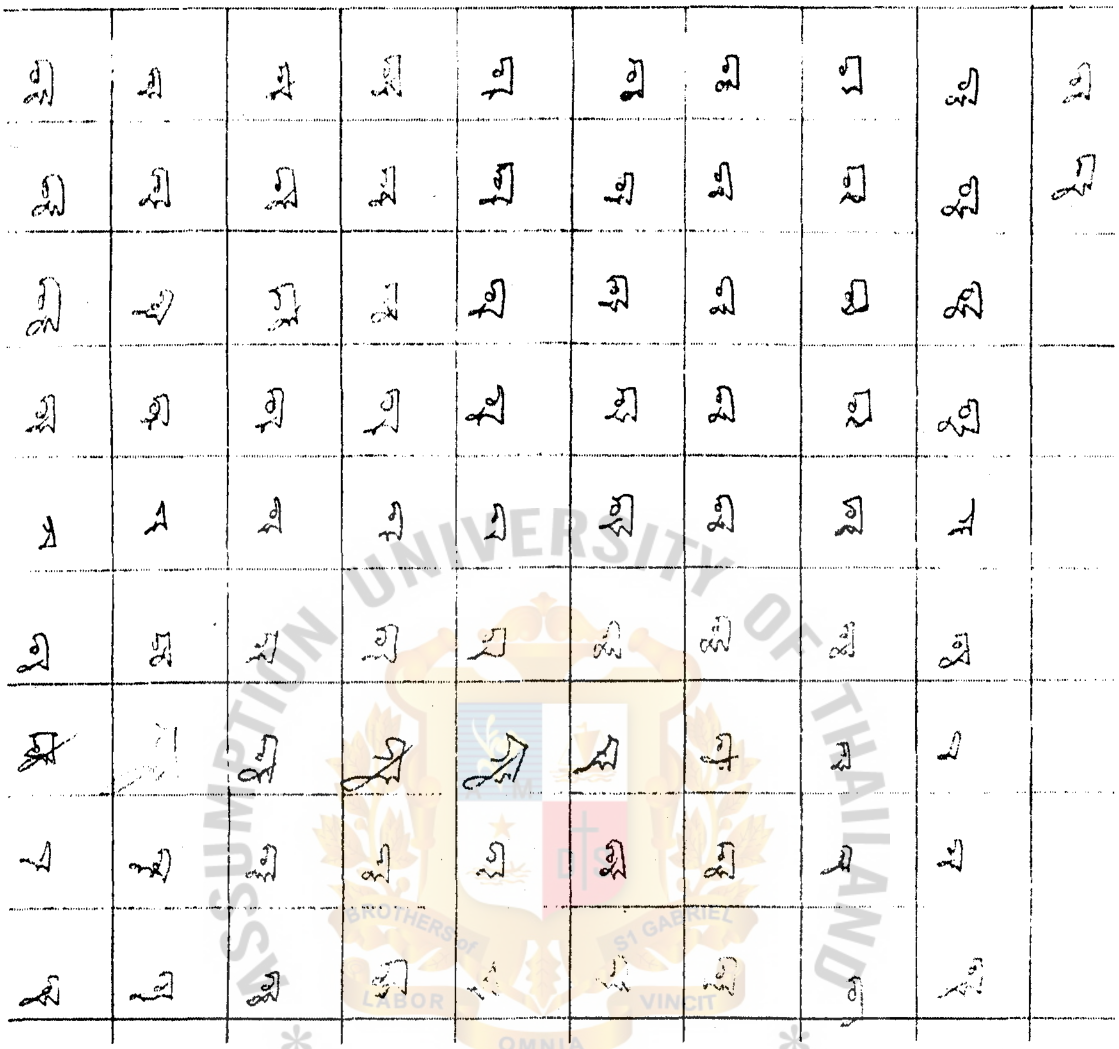
A.13 Alphabet ญ

ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ

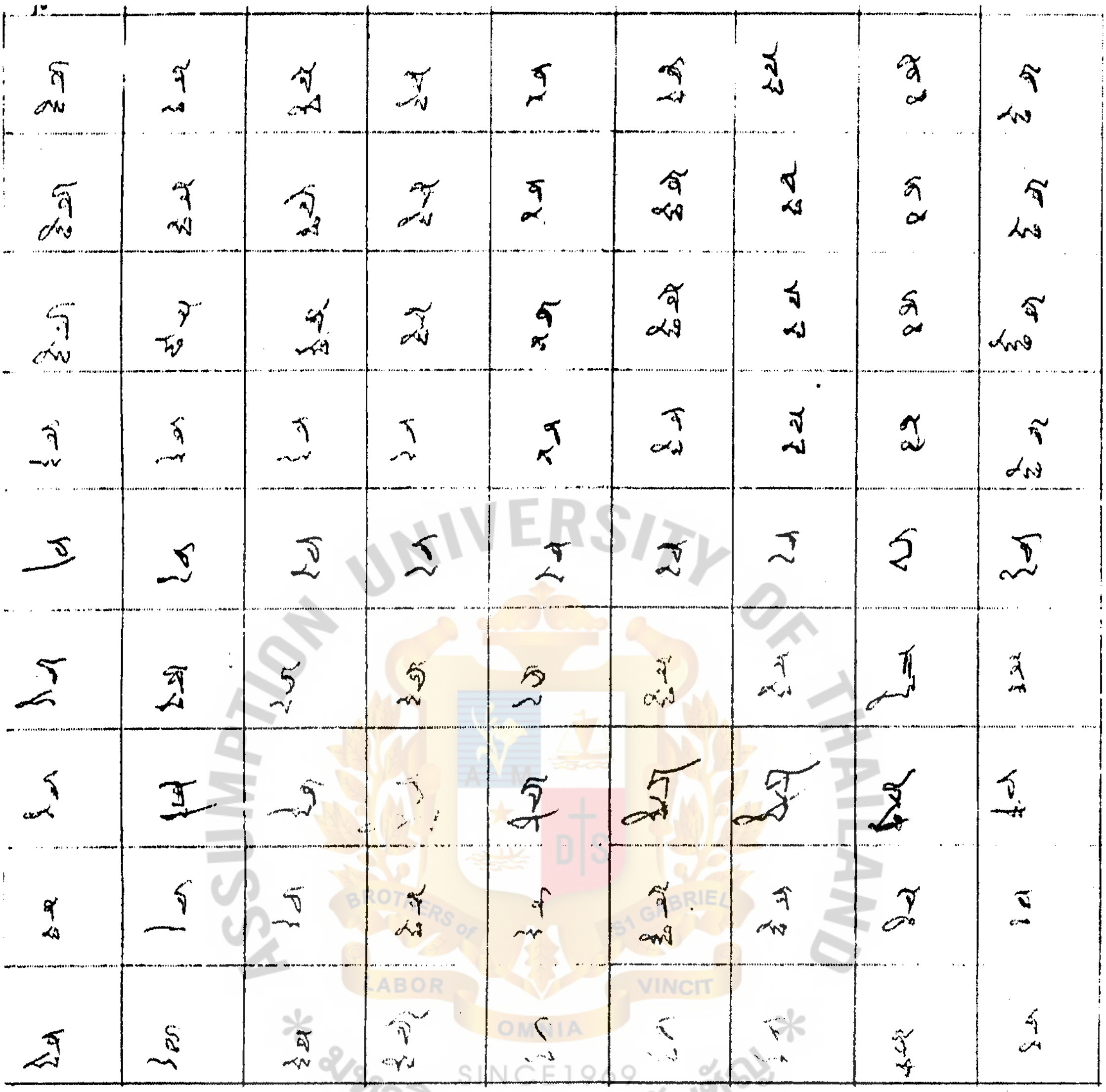
A.14 Alphabet ฎ



A.15 Alphabet ฏ



A.16 Alphabet ฐ



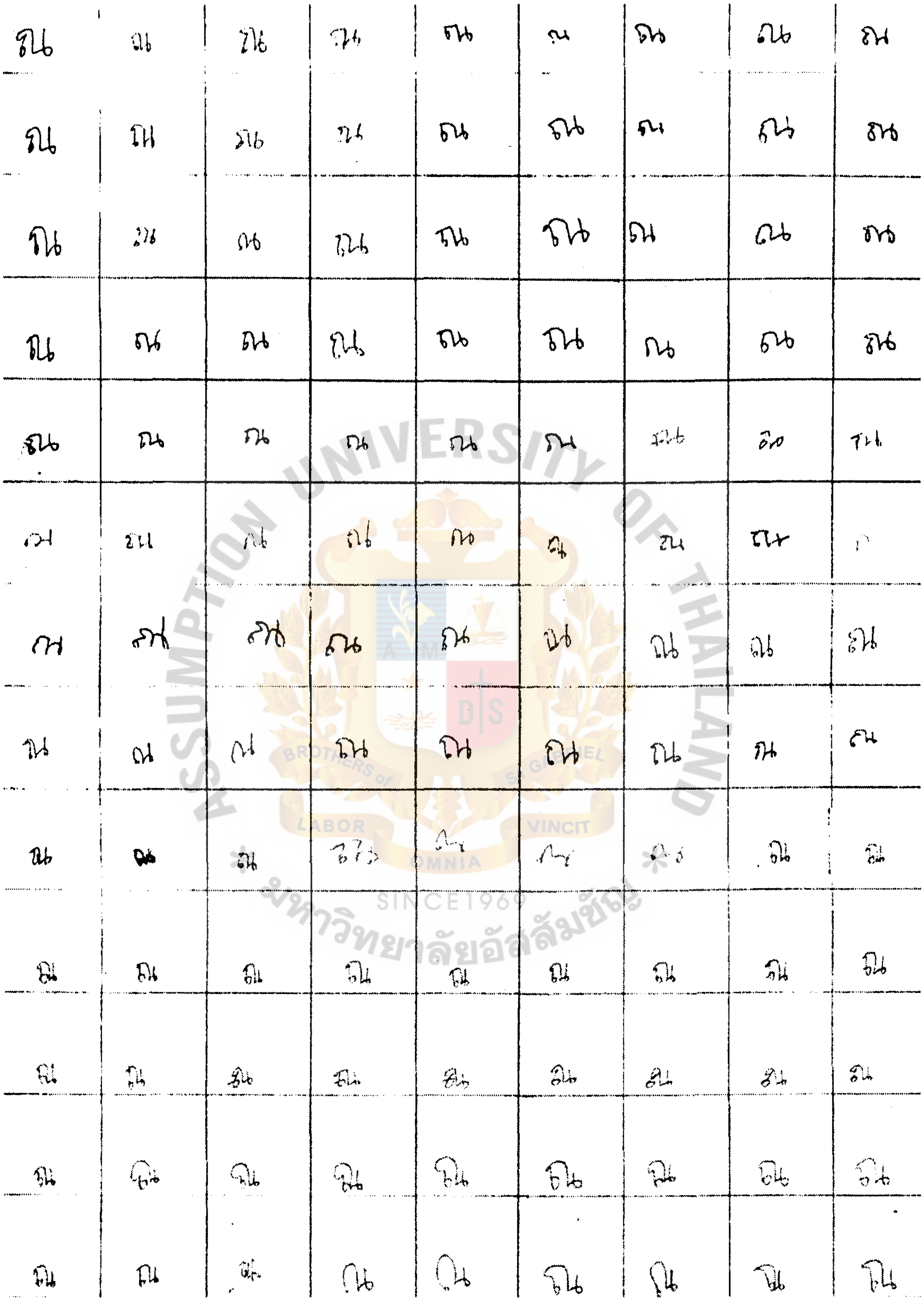
A.17 Alphabet ฑ

ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ

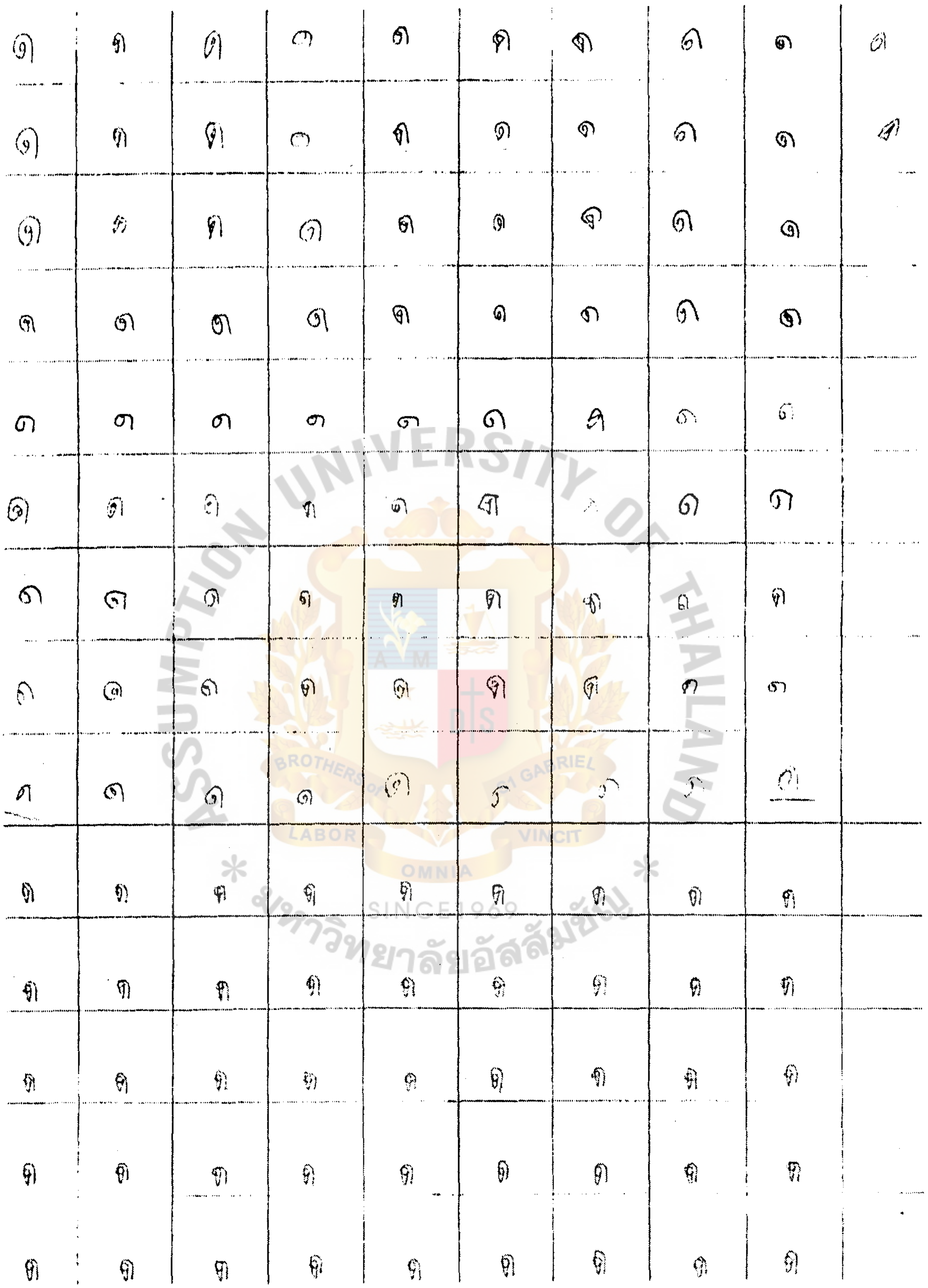
A.18 Alphabet ฒ

ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ ฒ

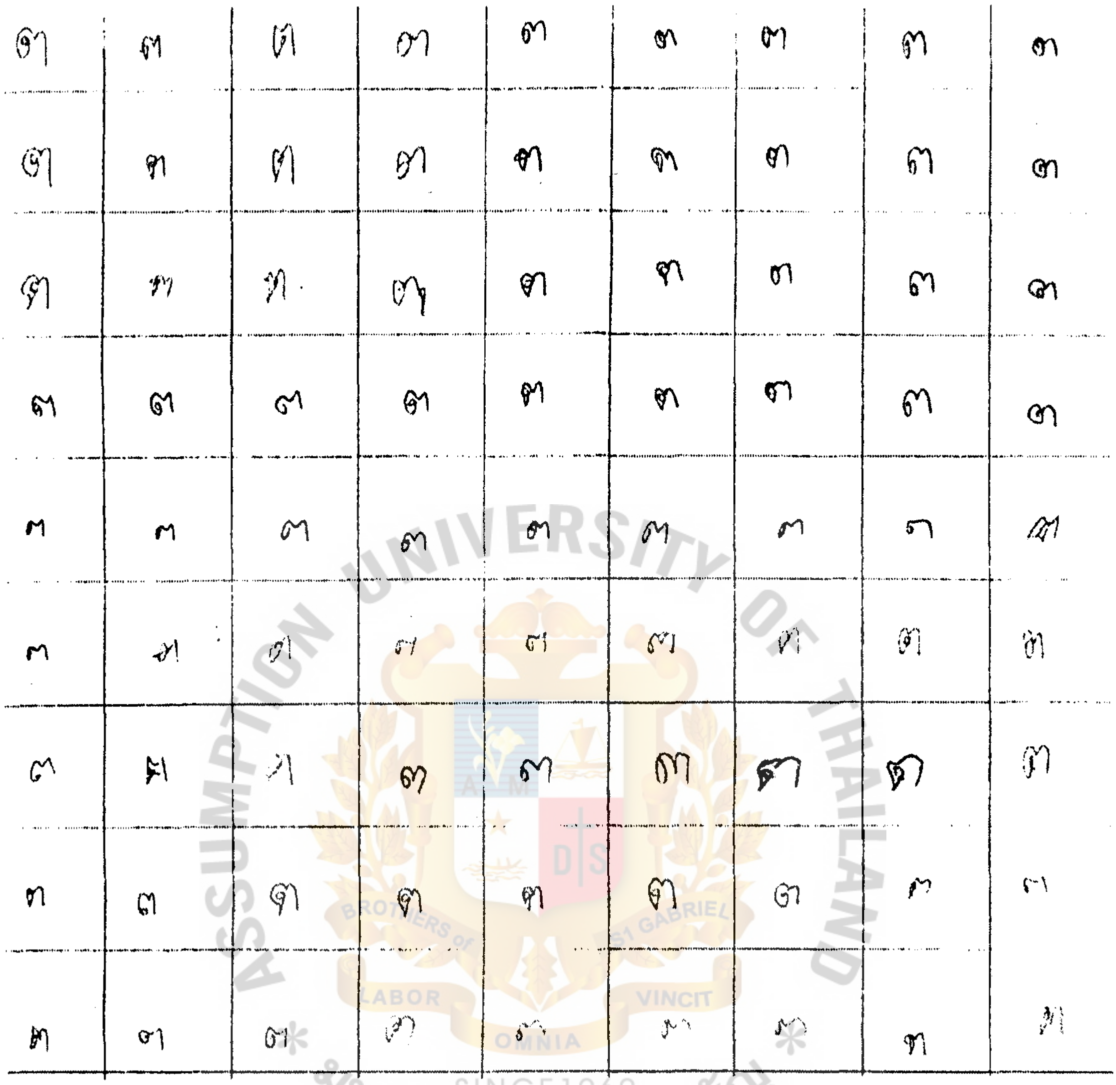
A.19 Alphabet ณ



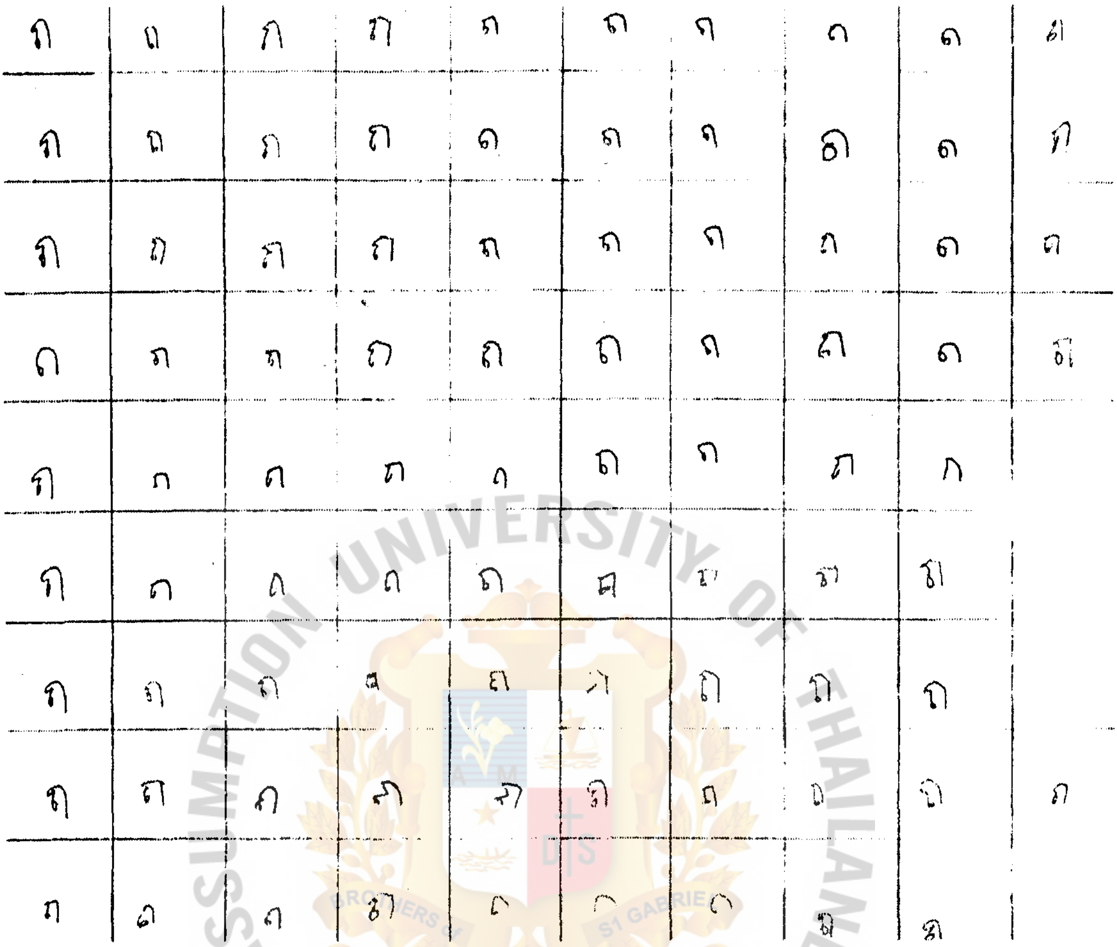
A.20 Alphabet ค



A.21 Alphabet ต



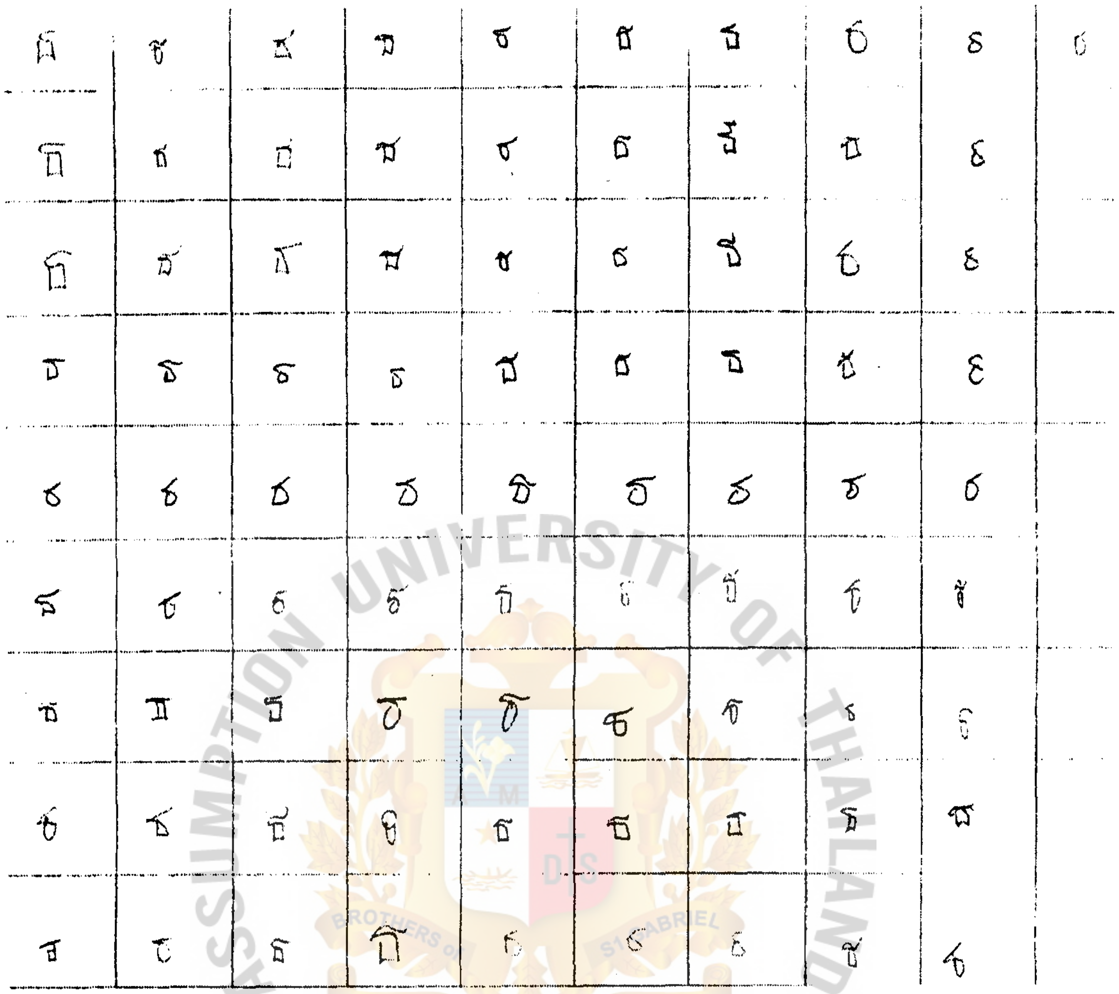
A.22 Alphabet ถ



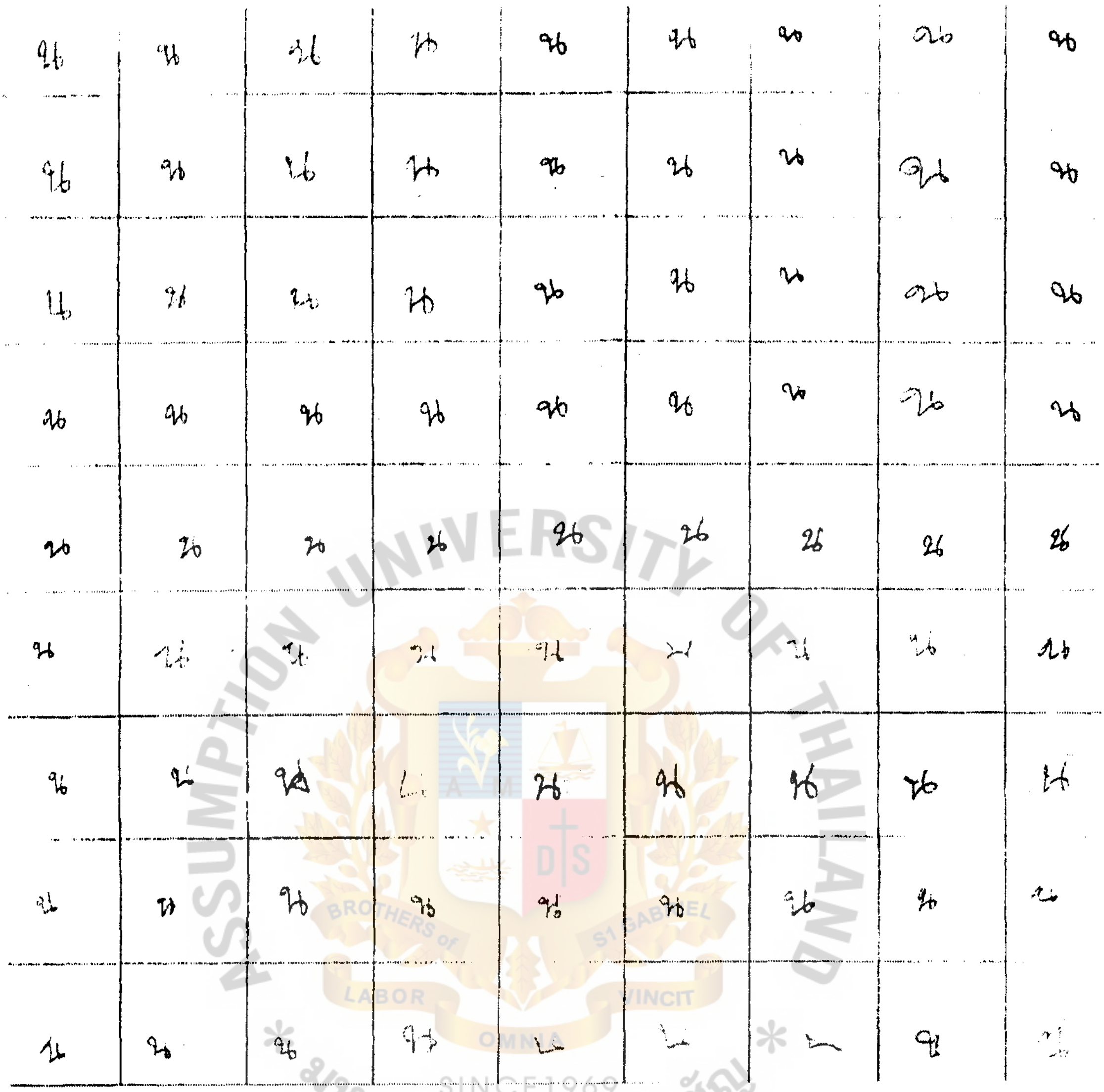
A.23 Alphabet ท

ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท
ท	ท	ท	ท	ท	ท	ท	ท	ท

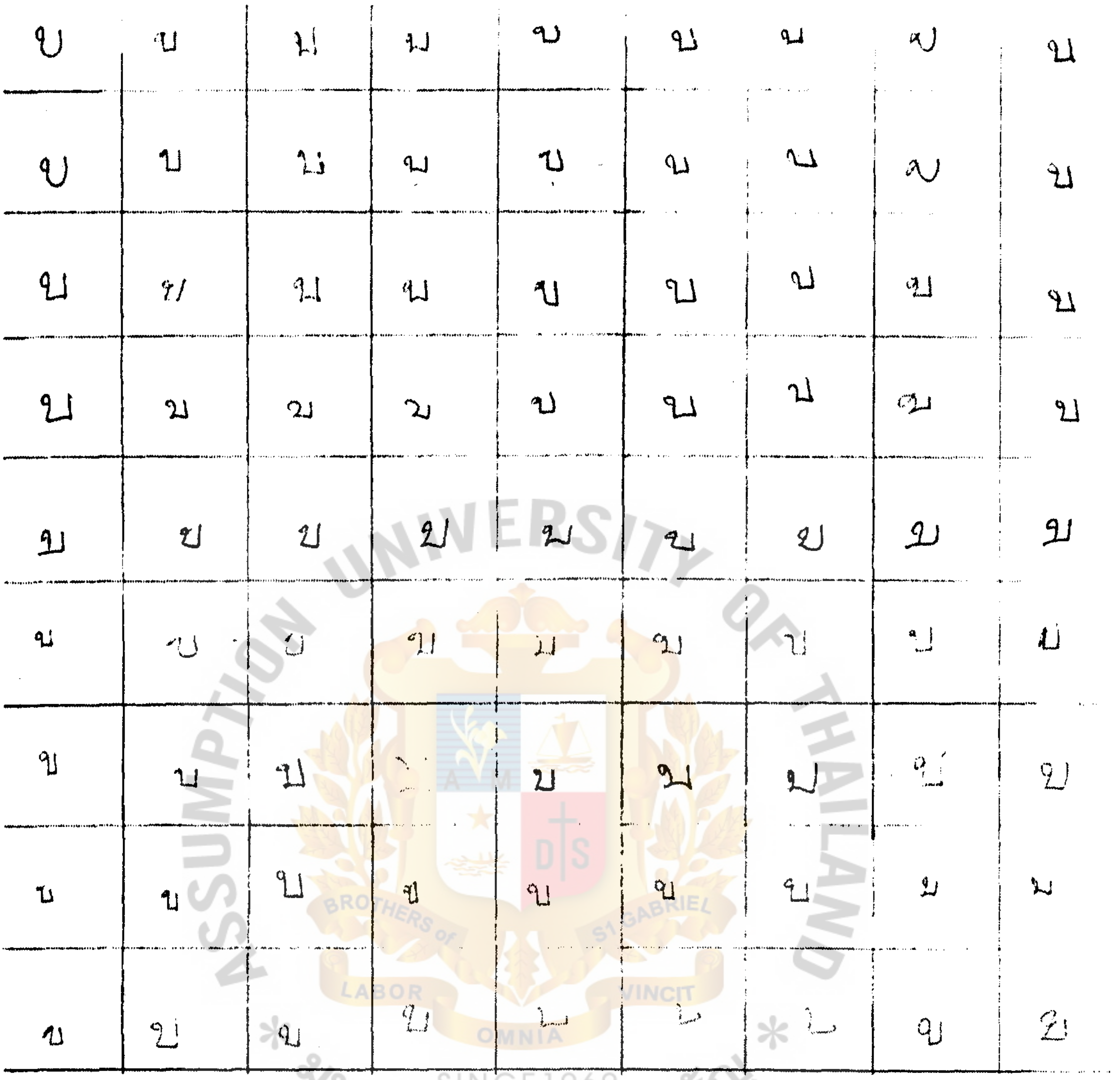
A.24 Alphabet ธ



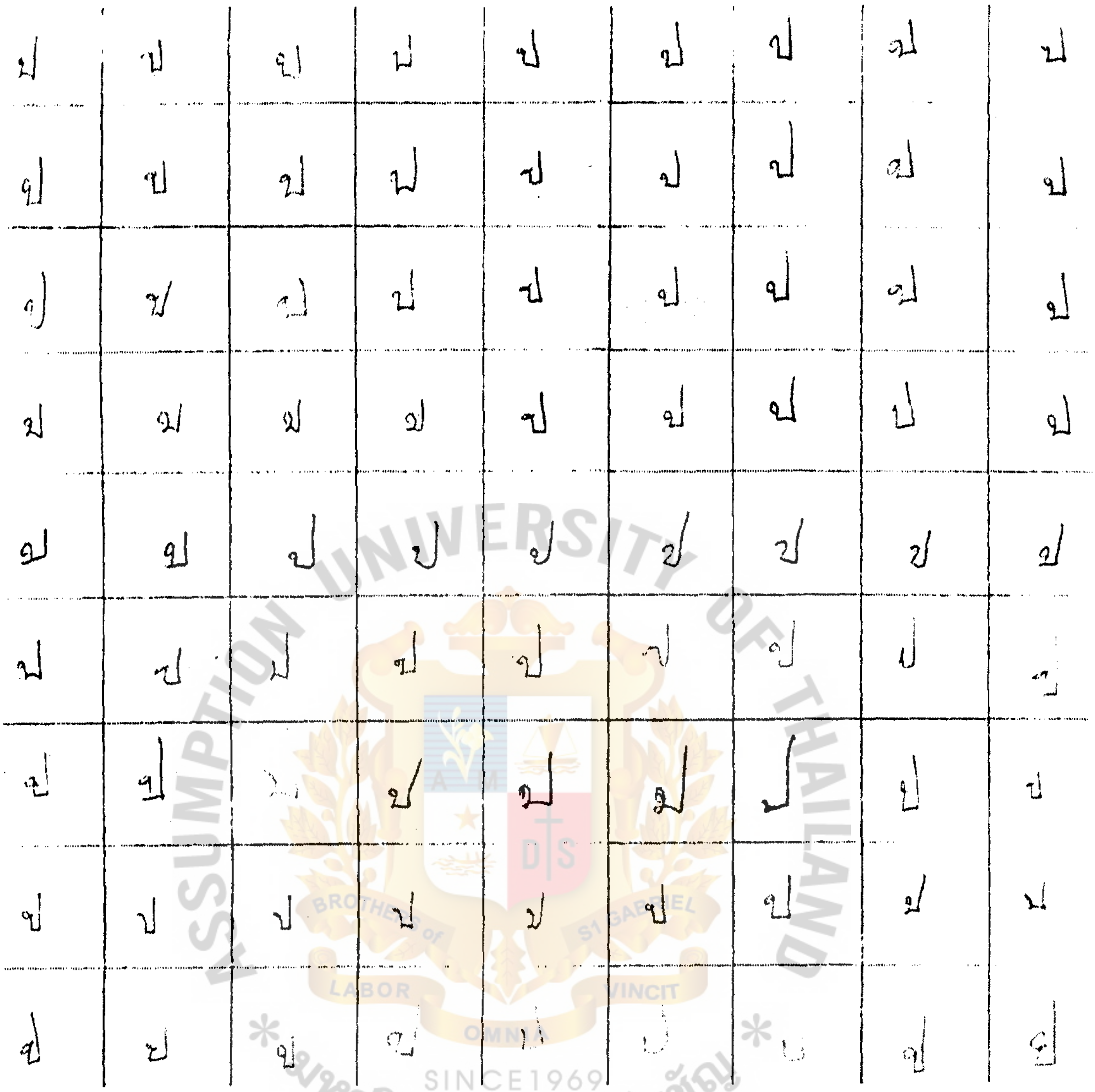
A.25 Alphabet น



A.26 Alphabet บ



A.27 Alphabet ป



A.28 Alphabet ผ

ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ผ	ผ							

A.29 Alphabet ฝ

ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ

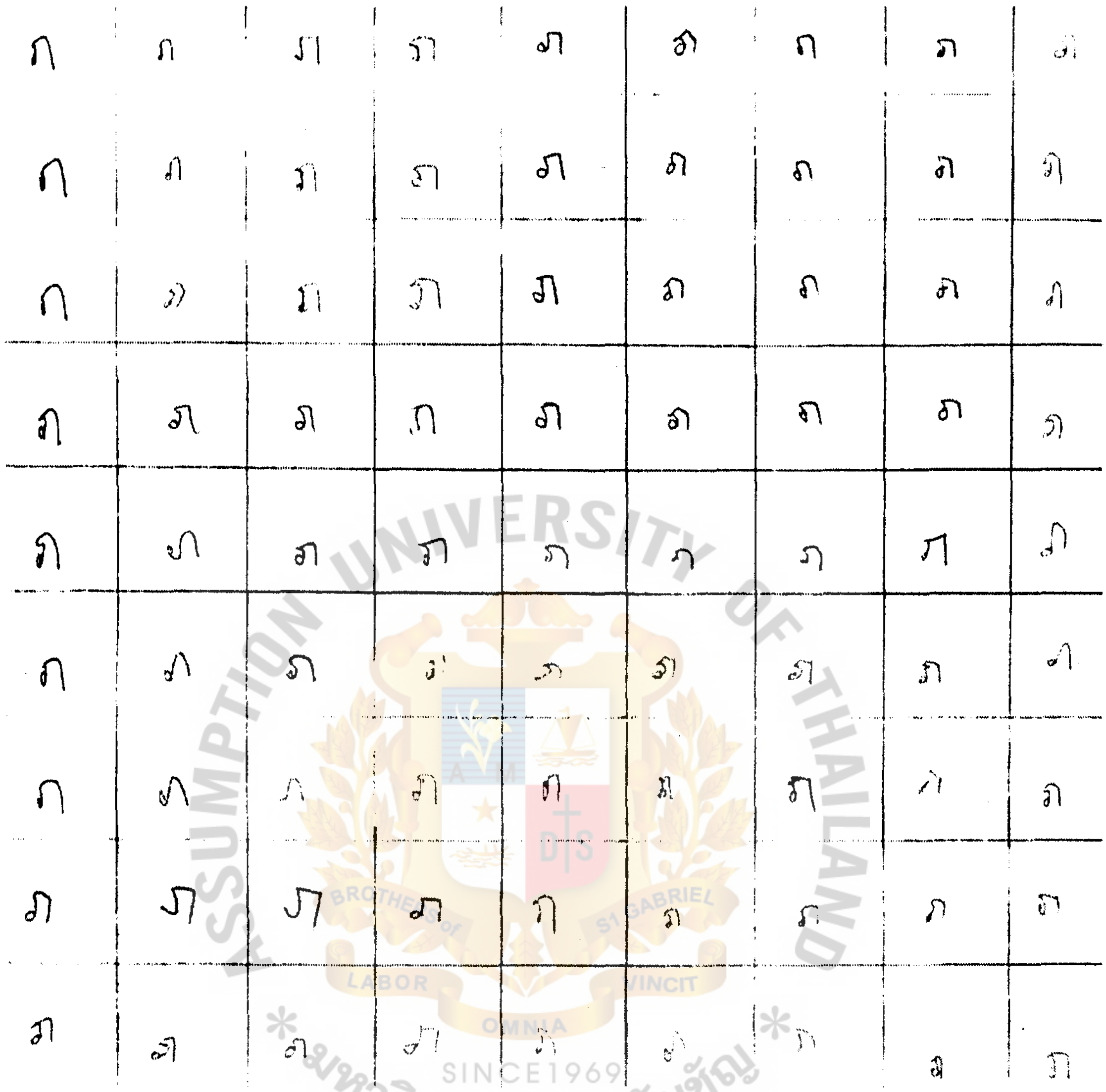
A.30 Alphabet พ

พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	[illegible]
พ	พ	พ	พ	พ	พ	พ	พ	พ
[illegible]	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	พ	พ	พ	พ	พ	พ	พ	พ
พ	[illegible]							

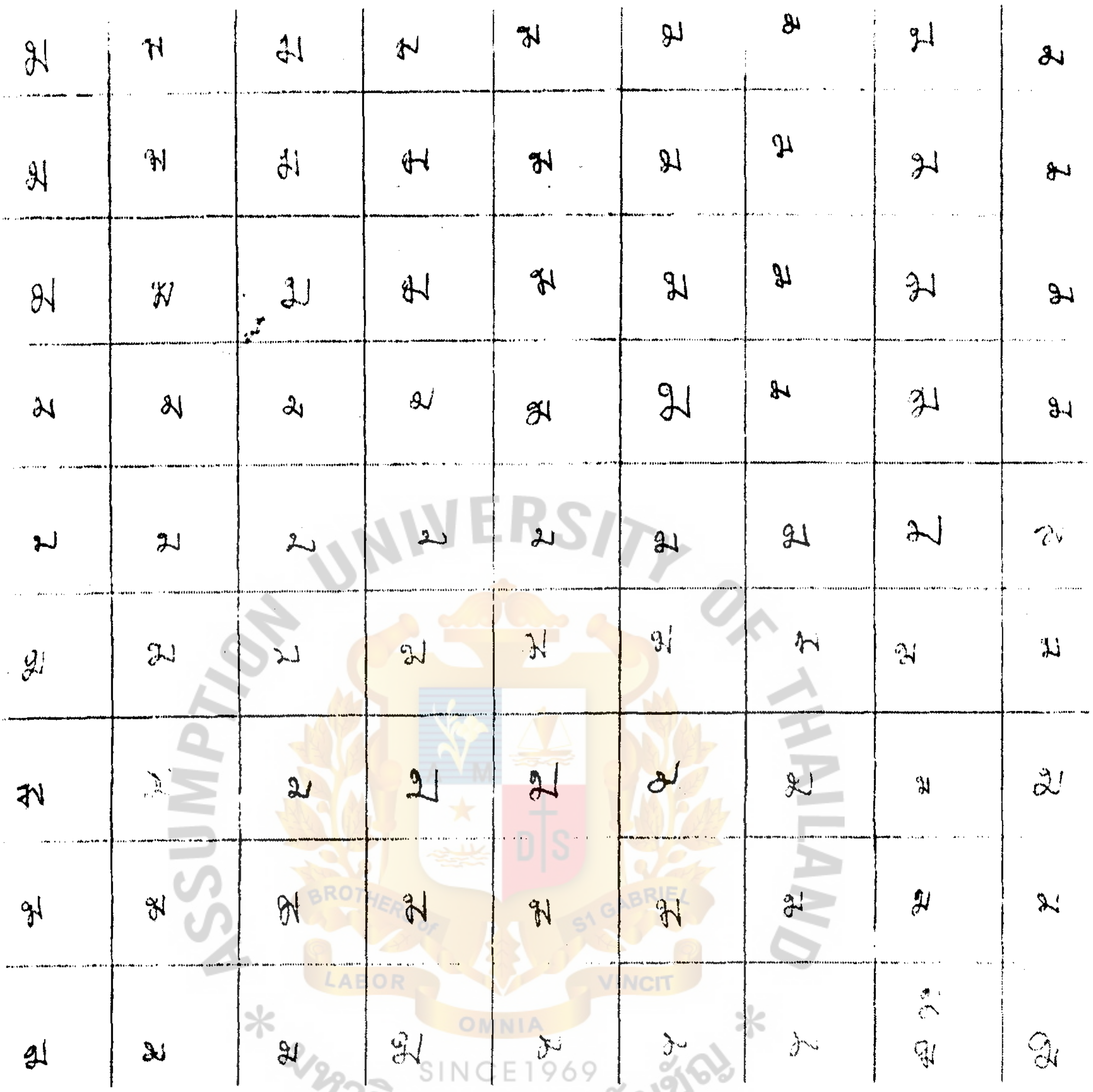
A.31 Alphabet ฟ

ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	

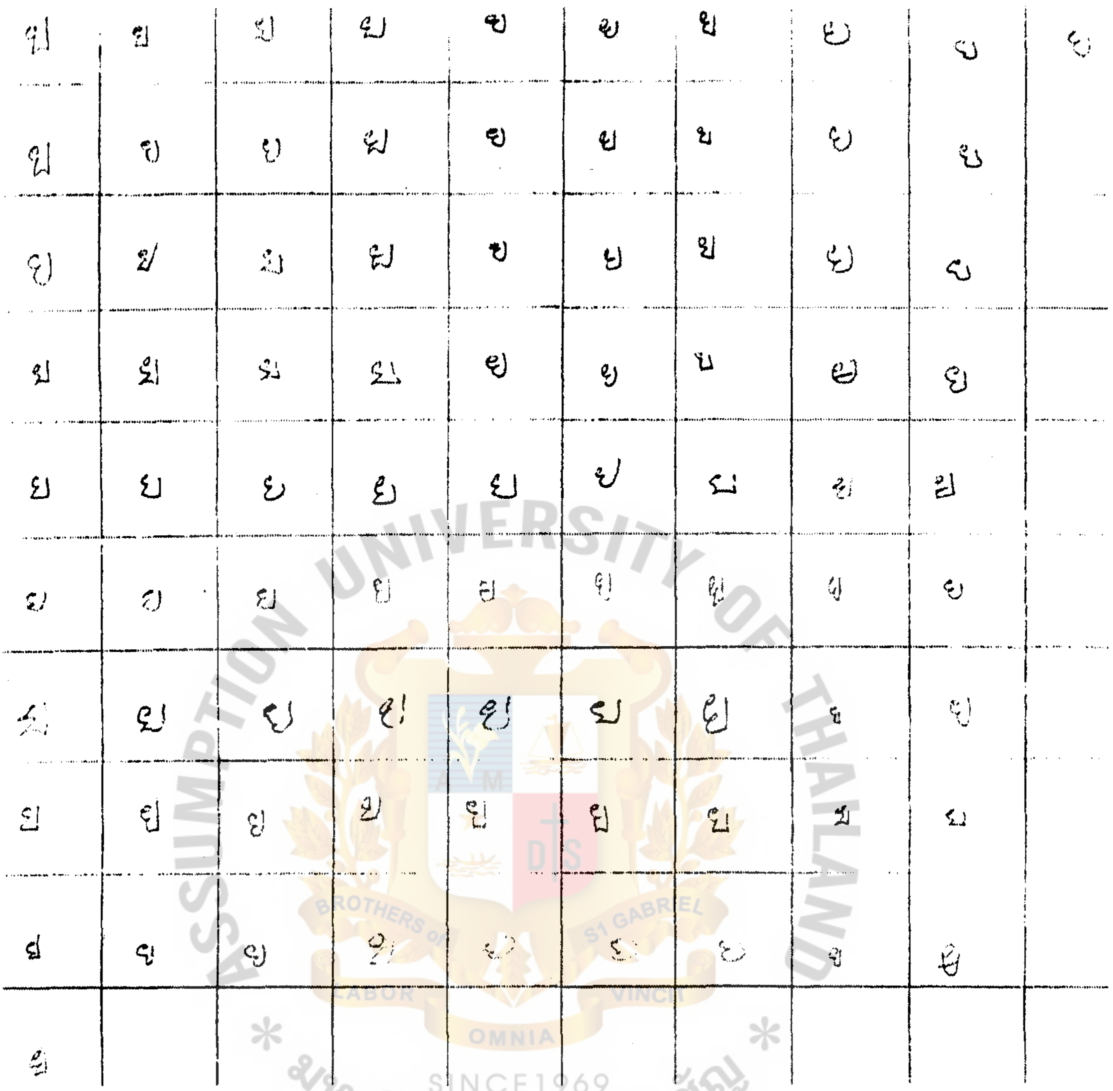
A.32 Alphabet ภ



A.33 Alphabet ม



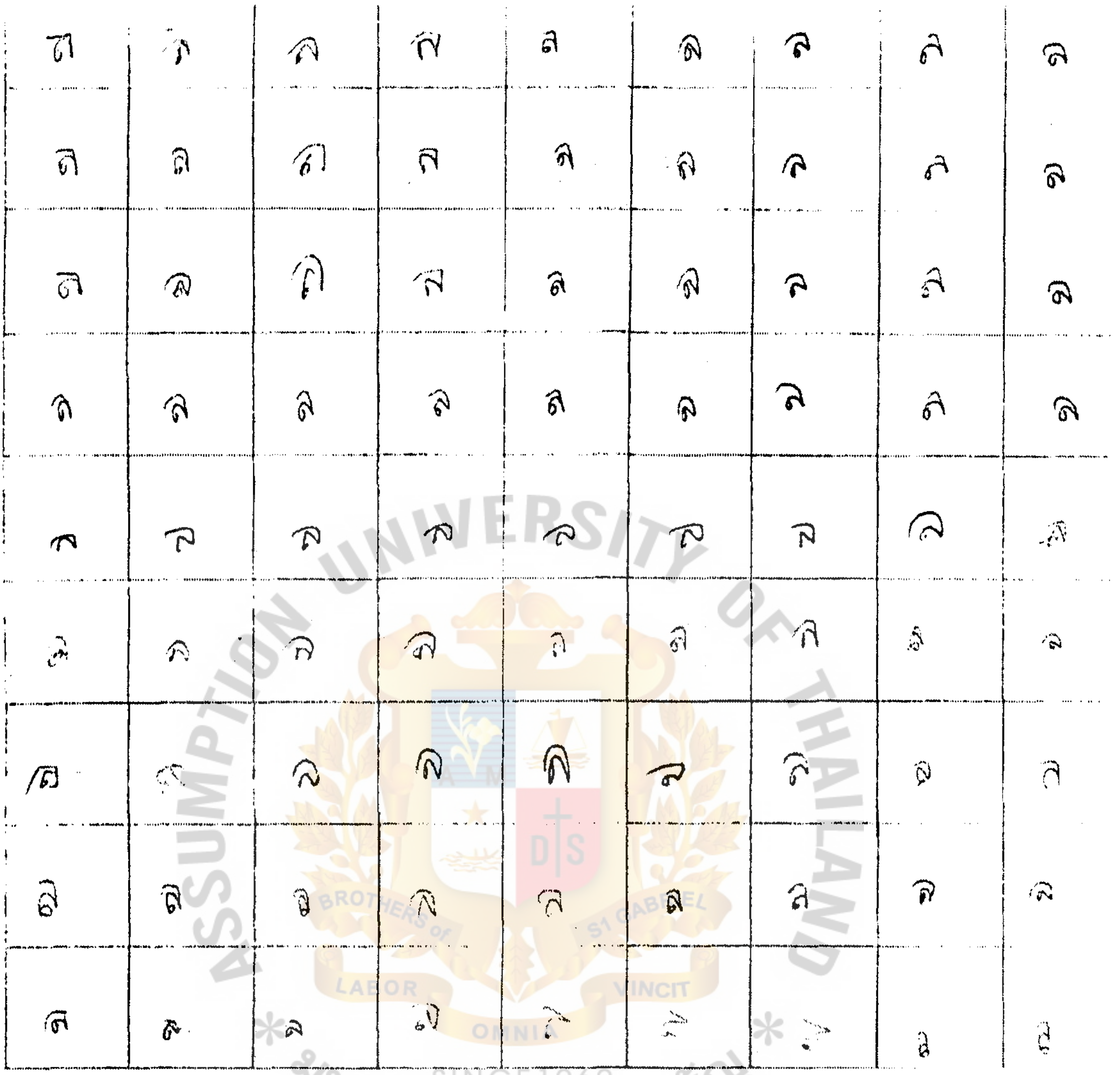
A.34 Alphabet ย



A.35 Alphabet ร

ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร
ร	ร	ร	ร	ร	ร	ร	ร	ร

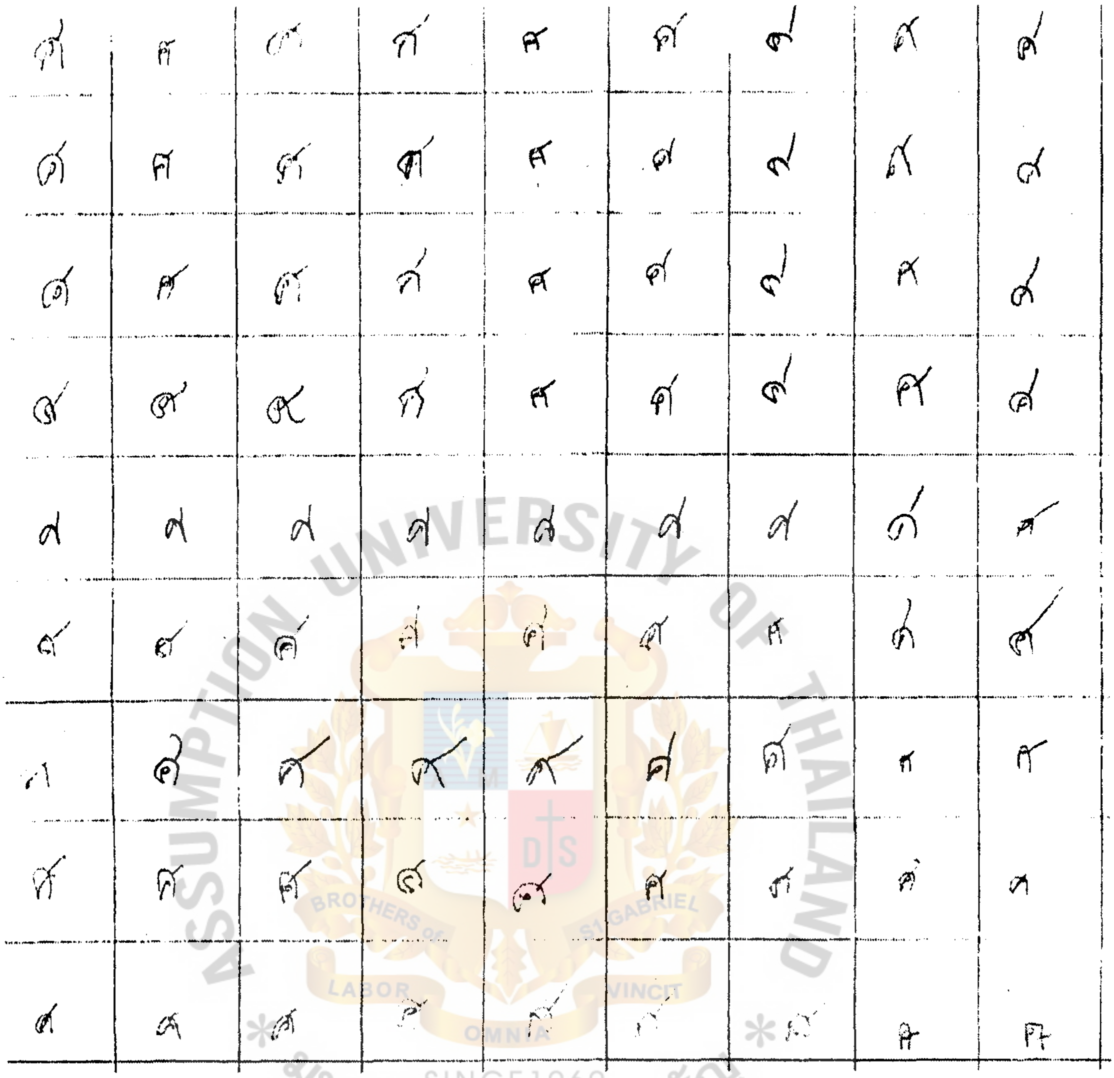
A.36 Alphabet ล



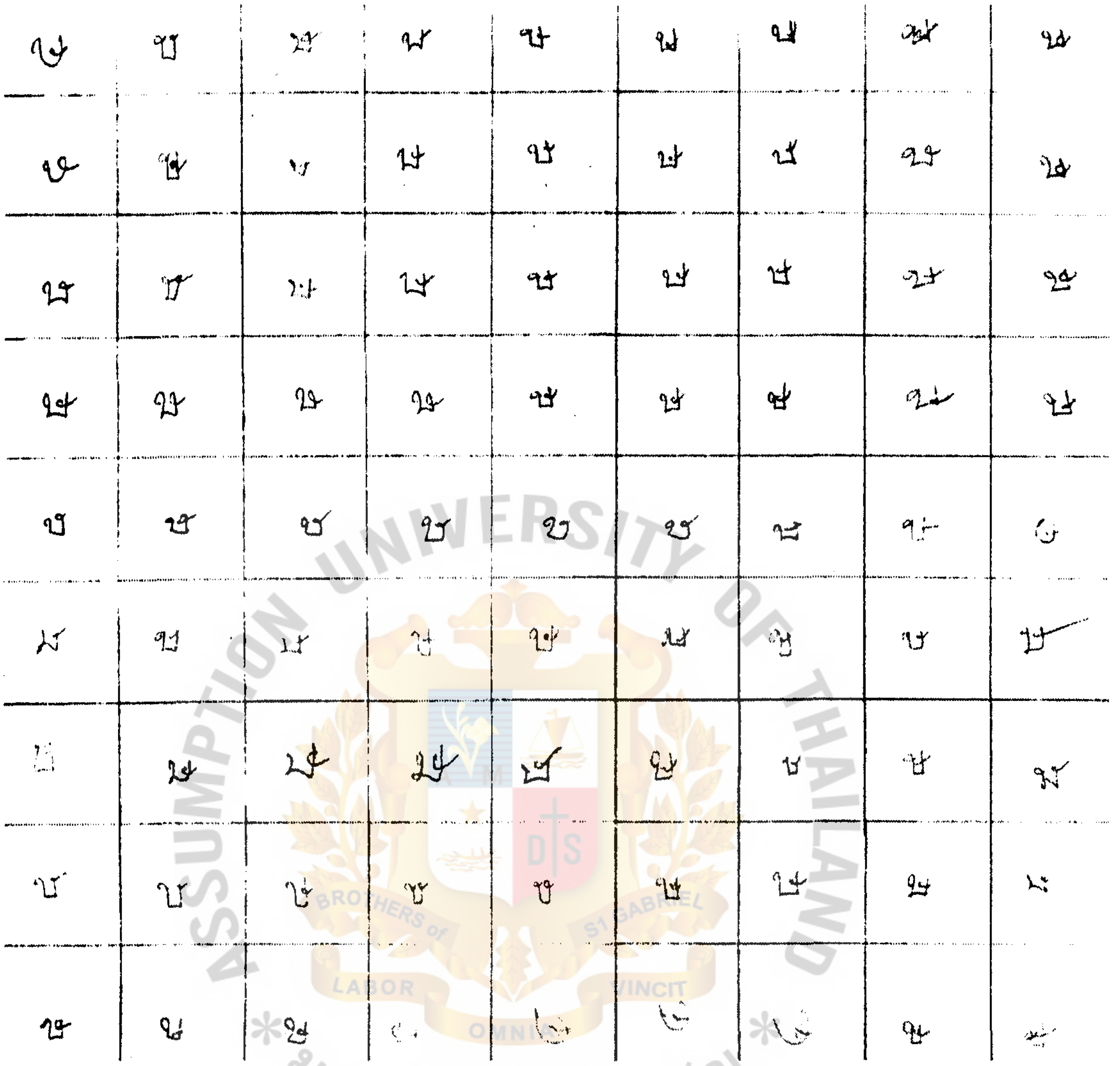
A.37 Alphabet ว

ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว	ว	ว	ว	ว	ว
ว	ว	ว	ว ว	ว	ว	ว	ว	ว

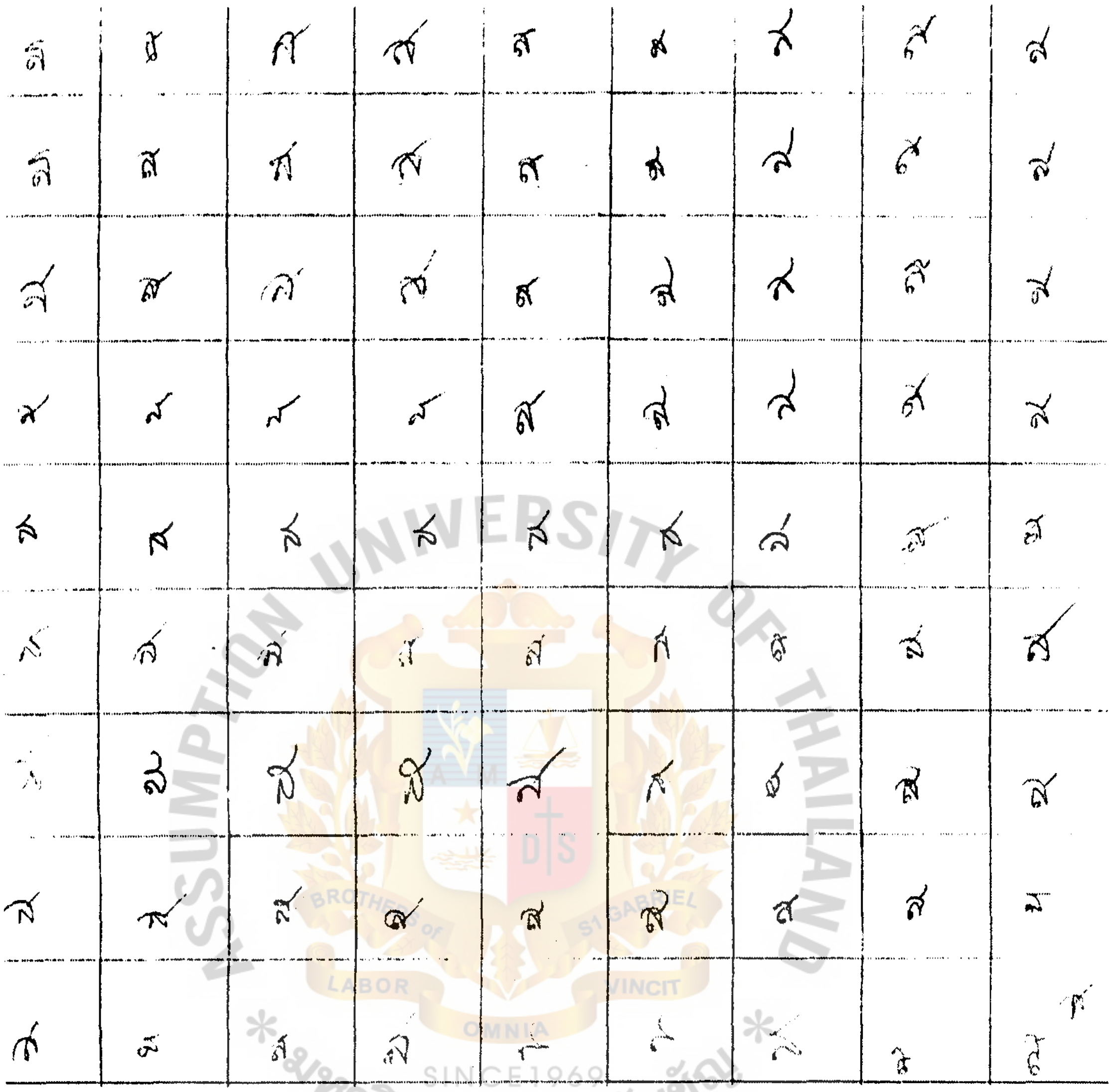
A.38 Alphabet ศ



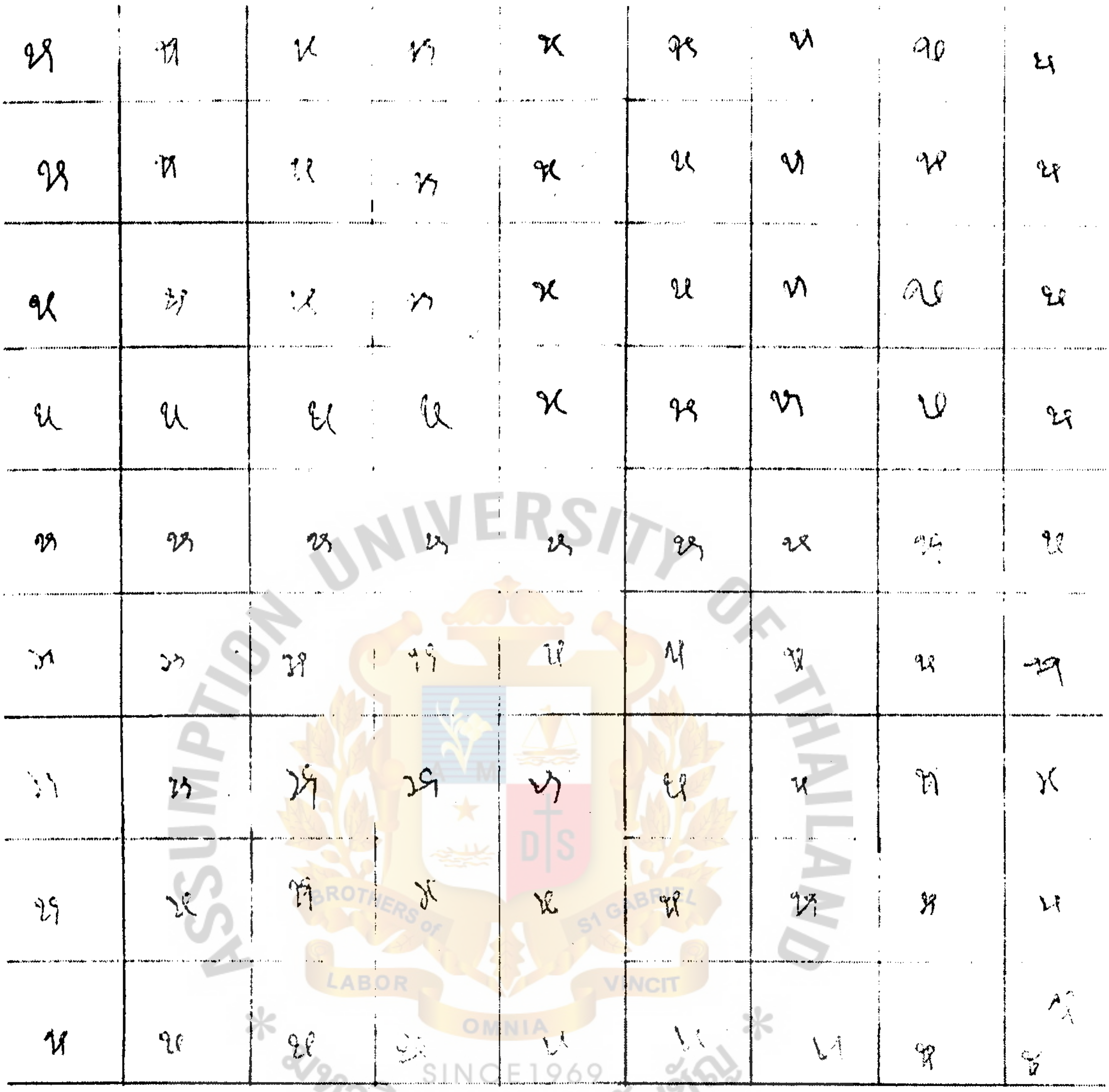
A.39 Alphabet ย



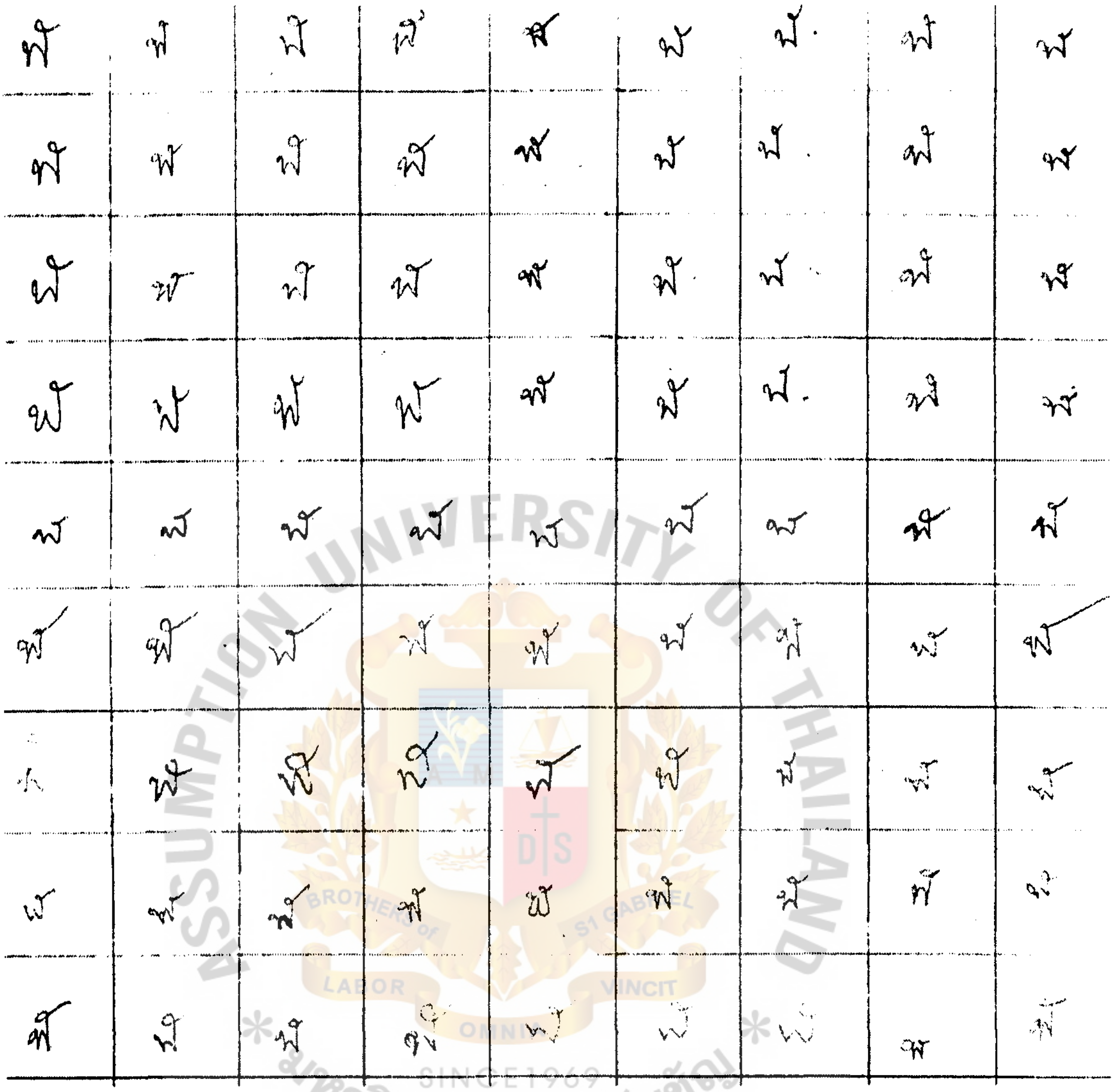
A.40 Alphabet ส



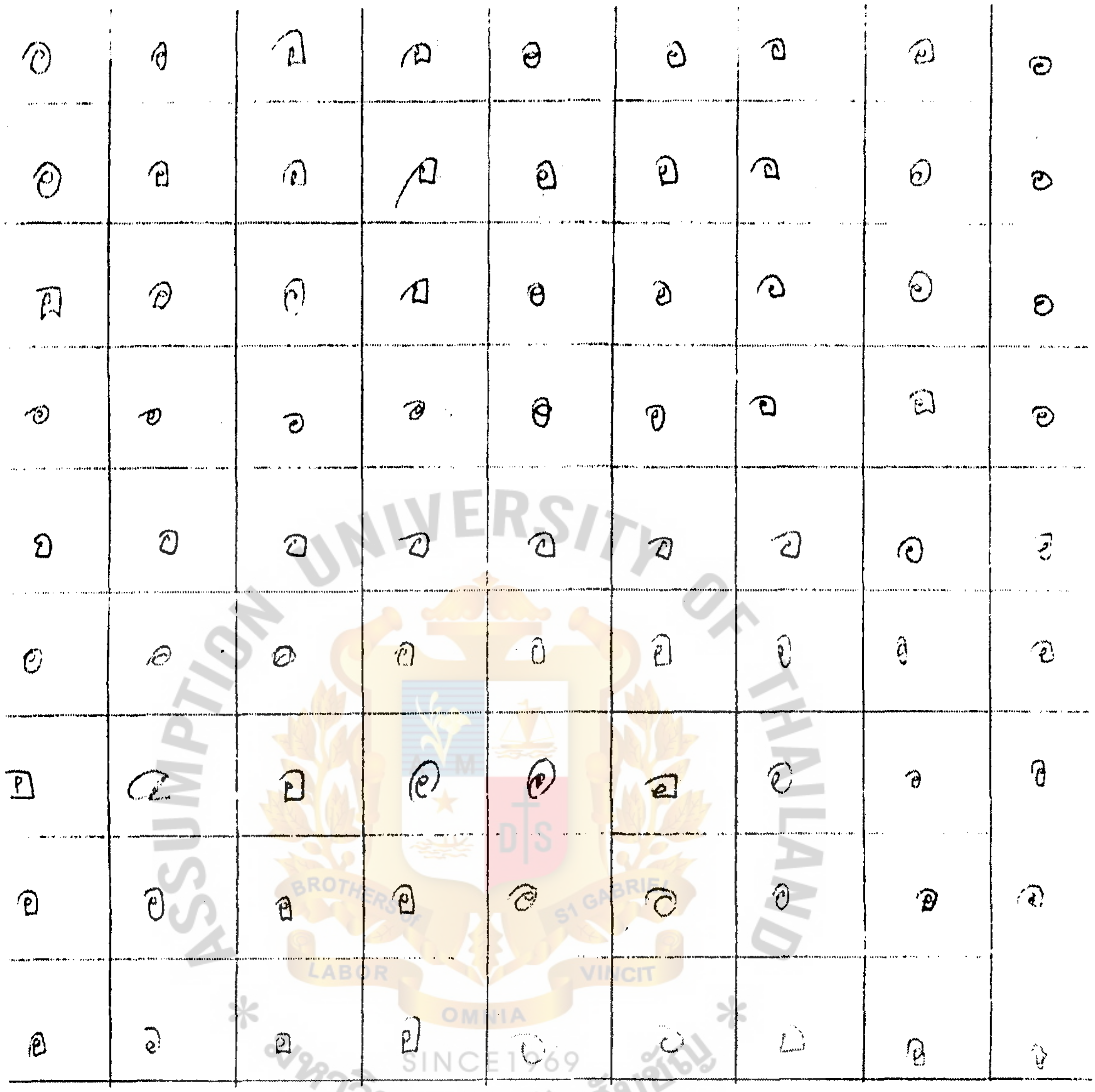
A.41 Alphabet น



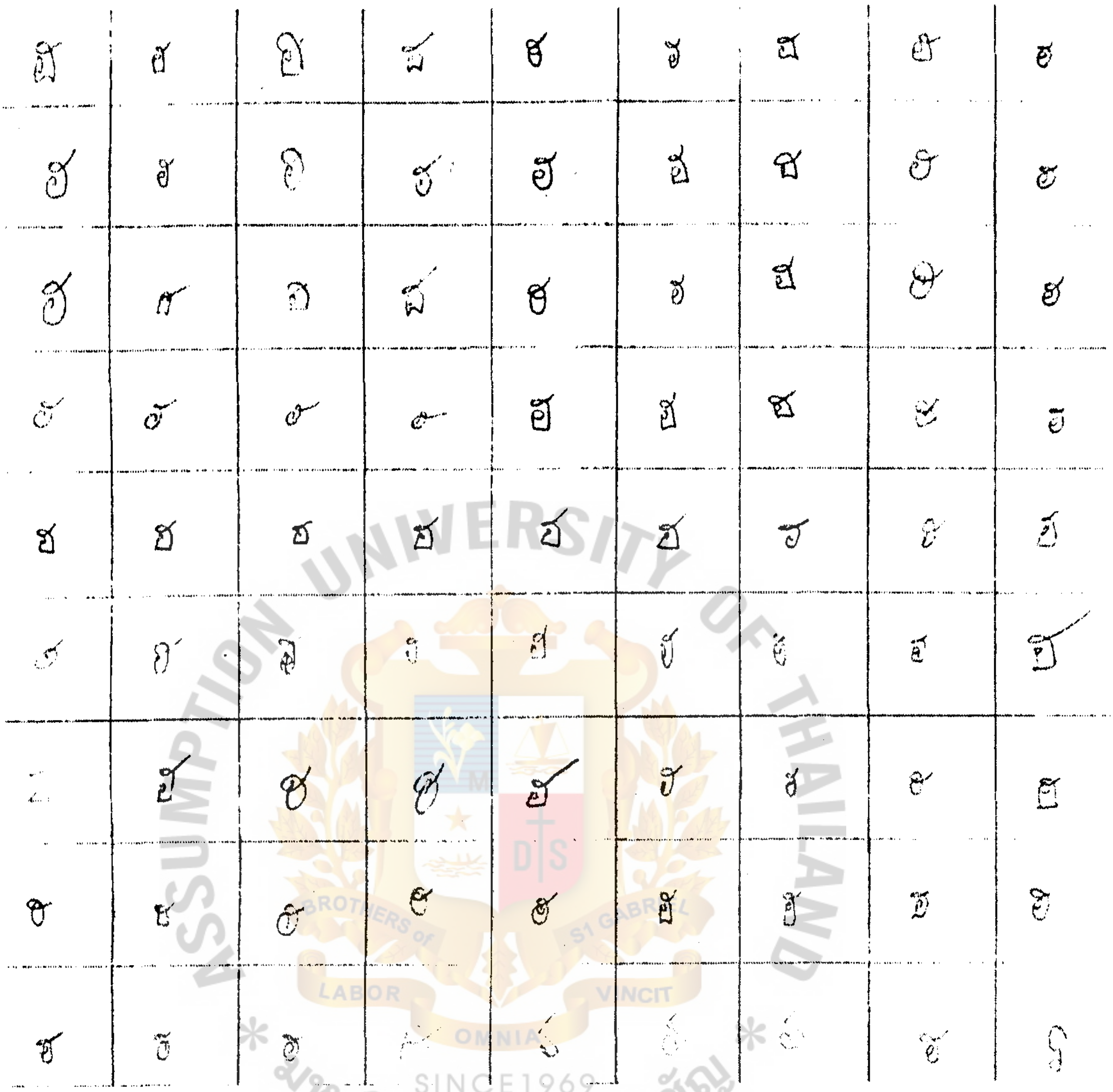
A.42 Alphabet พ



A.43 Alphabet อ

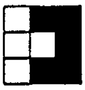
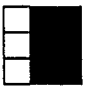
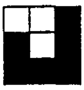
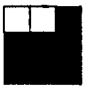
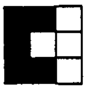
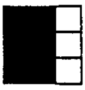
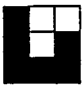
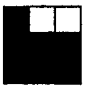
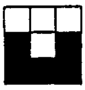
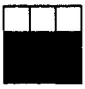
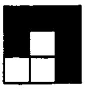
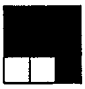
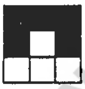
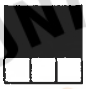
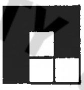
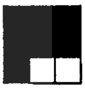
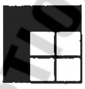
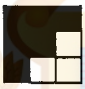
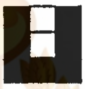
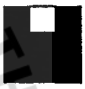
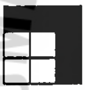
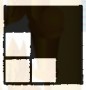
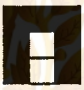
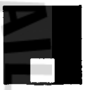
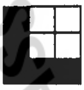
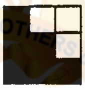
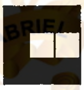
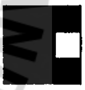
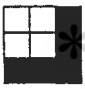
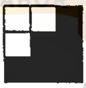
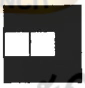
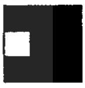


A.44 Alphabet ฮ

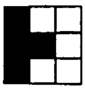
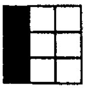
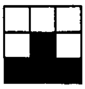
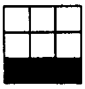
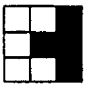
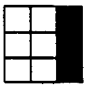
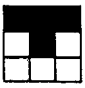
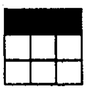


APPENDIX B: Smooth Spur and Redundant patterns

B1: Smooth Patterns

No.	Before Smooth	After Smooth	No.	Before Smooth	After Smooth
1			9		
2			10		
3			11		
4			12		
5			13		
6			14		
7			15		
8			16		

B2: Spur Patterns

No.	Before Spur	After Spur	No.	Before Spur	After Spur
1			3		
2			4		

B3: Redundant Patterns

No.	Before Redundant	After Redundant	No.	Before Redundant	After Redundant
1	□■□ □■■ □□□	□■□ □□■ □□□	11	□□□ ■■□ ■□□	□□□ ■□□ ■□□
2	□□□ □■■ □□□	□□□ □□■ □□□	12	□□□ □■□ □■■	□□□ □□□ □■■
3	□□□ ■■□ □□□	□□□ ■□□ □□□	13	□■□ ■■□ □■□	□■□ ■□□ □■□
4	□■□ ■■□ □□□	□■□ ■□□ □□□	14	□■□ □■■ □■□	□■□ □□■ □■□
5	□□□ □■□ ■□□	□□□ □□□ ■□□	15	□□□ ■■■ □■□	□□□ ■□■ □■□
6	□□■ □■□ □□□	□□■ □□□ □□□	16	□■□ ■■■ □□□	□■□ ■□■ □□□
7	□□□ □■□ □□■	□□□ □□□ □□■	17	□□□ □■■ □□■	□□□ □□■ □□■
8	■□□ □■□ □□□	■□□ □□□ □□□	18	□■■ □■□ □□□	□■■ □□□ □□□
9	□□■ □■■ □□□	□□■ □□■ □□□	19	■□□ ■■□ □□□	■□□ ■□□ □□□
10	■■□ □■□ □□□	■■□ □□□ □□□	20	□□□ □■□ ■■□	□□□ □□□ ■■□