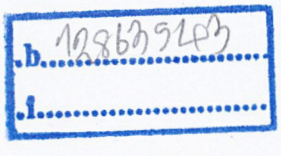
Mobile application for Retrieving information from National ID card by OCR

Kawinthat Padmindra
Teratat Sutitosatham
Piyabhut Wongsahan



BACHELOR OF ENGINEERING PROGRAM IN SOFTWARE ENGINEERING
INTERNATIONAL COLLEGE
KING MONGKUTS INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016
KMITL-2017-IC-B-003-002

Mobile Application for Retrieving Information from National ID Card by OCR

Kawinthat Padmindra

Teratat Sutitosatham

Piyabhut Wongsahan

Bachelor of Engineering in Software Engineering

International College

King Mongkut's Institute of Technology Ladkrabang

Academic Year 2016

KMITL-2017-IC-B-003-002

Thesis - Academic Year 2016

Bachelor of Engineering in Software Engineering

International College

King Mongkut's Institute of Technology Ladkrabang

Title - Mobile Application for Retrieving Information from National ID Card by OCR

Students :

Kawinthat Padmindra 56090008

Teratat Sutitosatham 56090029

Piyabhut Wongsahan 56090041

Approved by submission

montri

.....

(Dr.Montri Phothisonothai)

Advisor

Date: June 6, 2017

Abstract

Nowadays, the traditional methods to acquire information from Thai National ID card (e.g. ID number, name, address, etc.) take long time processing and also the errors may result in the situation that can be made by human easily. Therefore, in this project, Mobile Application for Retrieving Information from National ID Card using Optical Character Recognition (OCR) is proposed. The Markov chain is used as a feature and the minimum sum of absolute different method is used as a classifier. In the experiment, we tested input image of National ID card dataset of 1) printed file, 2) scanned file, and 3) captured file by mobile camera. The results showed that the proposed method could extract information into digital character codes.

Acknowledgements

First of all, we would like to express our sincere thanks to our project advisor, Dr.Montri Phothisonothai, for his invaluable helps and guidance throughout the software project. We are most grateful for his teaching and advice. We would not have achieved this far without all the support that I have always received from him.

In addition, we are grateful for the professors of software engineering and the committees: Asst.Prof.Dr. Visit Hirankitti, Asst.Prof.Dr. Chaiwat Nuthong, Dr. Isara Anantavrasilp, Dr. Ukrit Watchareeruetai and, Dr. Natthapong Jungteerapanich for the suggestion and all their helps in project seminar class and final presentation. Also heartfelt thanks to my friends for help us doing an experiment.

Finally, we most gratefully acknowledge our parents for all their support throughout the period of this project.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, obtaining the information from any kinds of cards take a lot of time because of the ways that people do like copying the cards, writing the information from the cards and typing the information. Not even take a lot of time but also a lot of resources to process. There is also the fast way to obtain an information from cards like using smart chips but it needs to use specific devices and database to import data of that card. The other main problem that occurs from these processes are human errors like misreading or miswriting.

Because of these problems, we try to create the faster way to obtain the information from the card just in a few seconds by a general device like mobile phones and to reduce errors that occur from the process.

Optical Character Recognition(OCR) is used to solve this problem. We just need only a mobile phone with some levels of camera which are used to take a photo of the card to allow the system to interpret the information in the card into an electronic data. The duration that need to take place due to OCR is less than writing and typing down the information. The error that causes by copying

will replace by using a taking photo that we have already fixed the layout and the area to take.

1.2 Objective

The objective of this project is the implementation of an image processing algorithm by creating a mobile application to prove the result of the algorithm which uses to solve the problem of the process that people used to obtain the information from the card nowadays. Because it takes a long time to do, so the main feature of this project is to reduce the duration of the process obtaining the information from the card. This project changes the process from writing or typing down the information from the card to just taking the photo of the card and then automatically interprets those information into the electronic data. Moreover, we try to adapt the project to support other types of the card.

1.3 Scope of Work

The scope of this project can be summarized as follows:

- Developing the mobile application.
- Using with national ID card.
- Converting input images into digital character codes.
- Real-time processing.
- Storing the output in database.

1.4 Primary Contributions

In this project, the implementation of the image processing algorithm to extract the information on the national ID card into editable texts.

1. Propose a new algorithm to extract Thai character.
2. Implement mobile application for showing the result.
3. Create training set of character (Thai, English, Numerical).
4. Apply Markov value as the individual characteristics of character.

1.5 Procedure

We separate the work into 6 phases.

1. Plan the project
2. Research and study on OCR
3. Doing the mobile application
4. Implement Image processing method
5. Compare the correctness with a look-up table
6. Test the application

Chapter 2

Related work

2.1 Problem Description

Nowadays, the ways to obtain the data from national ID card need to use the specific device and also use quite a lot of time. The ways that usually use are copying an ID card, writing down on paper or typing down which also causes human errors easily. Moreover, to obtain the data in the faster way, users need to have a specific device for scanning bar code or smart chip on the card. For our project, we decide to make basic device like mobile phone be able to make a quick process of obtaining data from national ID cards with highly correctness. We use OCR technique to help the application to interpret the image into the electronic data for the information user need.

2.2 Review of Related Works

2.2.1 Realization of a High Performance Bilingual OCR System for Thai-English Printed Document [1]

The research is about using an OCR system to recognize printed Thai and English texts using dictionary look-up for classifying the characters. The system contains 3 steps : identifying a language, character recognition and checking the accuracy with dictionary look-up.

Identify Language : use suitable geometric properties for differentiation which will separate the document into different zone

Character Recognition : We categorized characters into groups according to its shape, height and various directional stroke features. Then the classifier will take the character classification in 2 consecutive steps which are coarse level and fine level.

Dictionary Look-up: to obtain the better level of accuracy, this state attempts to make use of a dictionary look-up as to check the recognized characters and formed as word tokens.

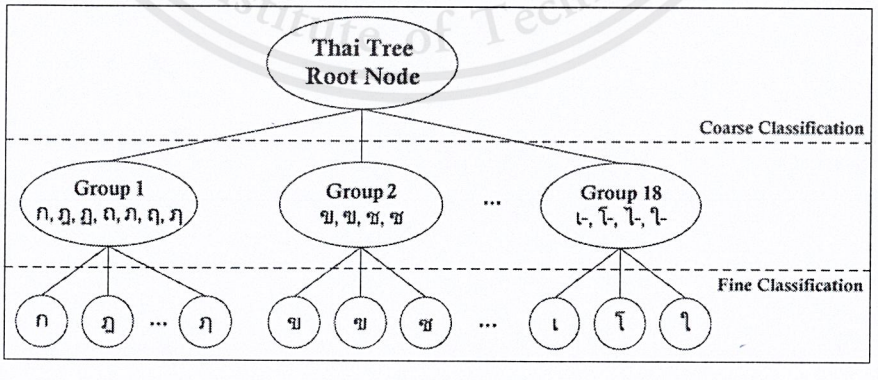


Figure 2.1: A part of the decision tree for Thai character recognition

2.2.2 Thai Style Recognition [2]

This research is from Chularat Tanprasert and Sutat Sae-tang. The purpose of this research is to classify the character that contains in a long sentence or a phase.

The first process is data preparation which contains 3 steps. The first step of the process is binary transformation which transforms the bitmap image into a binary code. The second step is to identify the character image from the document. The last step is to identify the text line of characters.

The second process is to identify what character it is by dividing into groups using leg of the character to separate characters into each group. Then classify the character in each group again by dividing the areas of character into 28 areas and checks for the percentage of black pixels in the areas.

Number of Members	Number of Legs	Member of the Group
13	1	ร เ โ ไ ใ ง จ ฐ ธ ว ฯ า ๆ
42	2	ก ข ค ฉ ช ซ ฎ ฏ ฑ ถ ท น บ ป ผ ฝ ภ ม ย ฤ ล ฦ ศ ษ ส ห ฬ อ ฮ ฆ ง จ ฐ ด ต ธ พ ฟ ว ฯ า ๆ
9	>2	ณ ญ ฒ ฌ ฆ ด ต พ ฟ

Figure 2.2: Member of character in three groups

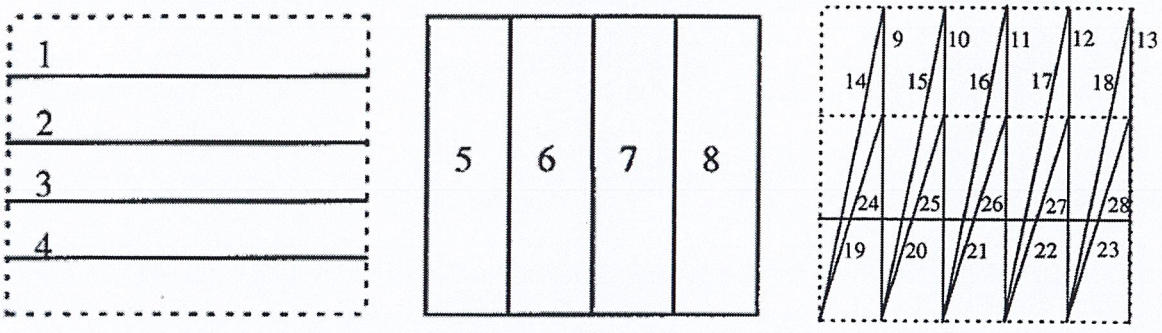


Figure 2.3: 28 areas for each character

2.2.3 Spell Checker for Thai Document [3]

This research is from Sarin Watcharabutsarakhum. The research is about the post- process of OCR which uses a dictionary to check the token of string to detect the error spelling. The technique that is used in this research is n-grams technique.

For examples, "สามารถเข้าถึงได้..."

n = 1; S_1 = {ส,ม,า,ร,ถ,เ,ข,ง,ไ,ด, ้, ึ, ้,...}

n = 2; S_2 = {สา,าม,มา,าร,รถ,ถเ,เข,ข้, ้า,าถ,ถึ, ึง,งไ,ได,ด้,...}

n = 3; S_3 = {สาม,ามา,มาร,ารถ,รถเ,ถเข,เข้,ข้า, ้าถ,าถึ,ถึง, ึงไ,งได,ได้,...}

n = 4; S_4 = {สามา,ามาร,มารถ,ารถเ,รถเข,ถเข้,เข้า,ข้าถ, ้าถึ,าถึง,ถึงไ, ึงได,งได้,...}

n = 5; S_5 = {สามาร,ามารถ,มารถเ,ารถเข,รถเข้,ถเข้า,เข้าถ,ข้าถึ, ้าถึง,าถึงไ,ถึงได, ึงได้,...}

n = 6; S_6 = {สามารถ,ามารถเ,มารถเข,ารถเข้,รถเข้า,ถเข้าถ,เข้าถึ,ข้าถึง, ้าถึงไ,าถึงได,ถึงได้,...)

Figure 2.4: An example process to extract n-grams

2.2.4 OCR with Word Prediction Technique for Bilingual Documents [4]

This research is from Sarin Watcharabutsarakhum. The paper describes a working model of a bilingual OCR for printed Thai and English text with word prediction technique. The system proposed in this paper is called BOCR-WP which stands for Bilingual Optical Character Recognition which consists of 4 main steps: Pre-processing, Language Identification, Character Recognition, and Post-processing.

Pre-processing : is a step of binarization of the image, reduction of the noises, segmentation of the line and character

Language Identification : is a step of identifying the language by considering the geometric properties of the character.

Character Recognition : is a step of recognizing the character by using feature extraction and classification technique. In this step, there will be an additional sub step which is a word prediction step that will use n-gram to determine the word verification.

Post-processing : is a step of error correction if there is an error to the recognized character in the previous step by which the correction will be corrected by looking up the dictionary.

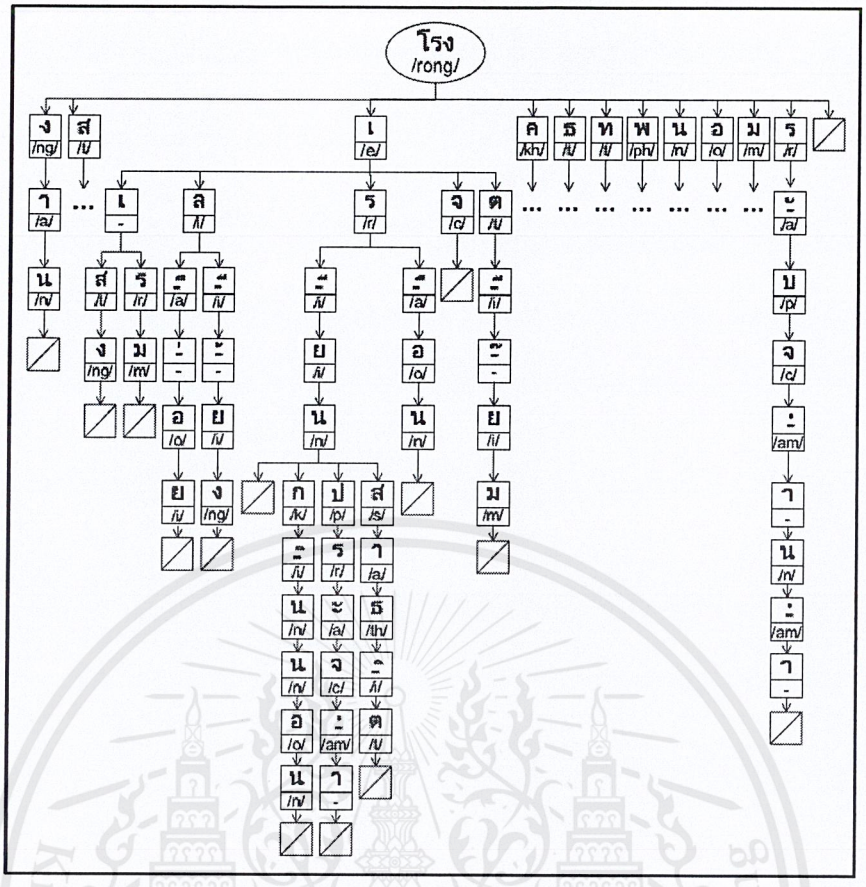


Figure 2.5: Word prediction in Thai using n-grams

2.2.5 Camera Based Mixed-lingual Card Reader for Mobile Device [5]

The paper introduced the design and the implementation of a mixed-lingual business card reader based on build-in camera which has the capability to recognize business cards with Chinese or English characters.

Firstly, the image of the business card is captured with the build-in camera Then the image processing modules detect lines, words, and characters. Next, the OCR module is incurred to recognize the detected characters. After that the recognized texts are categorized to extract information contained in the business card. Finally, human interference is needed to correct errors.

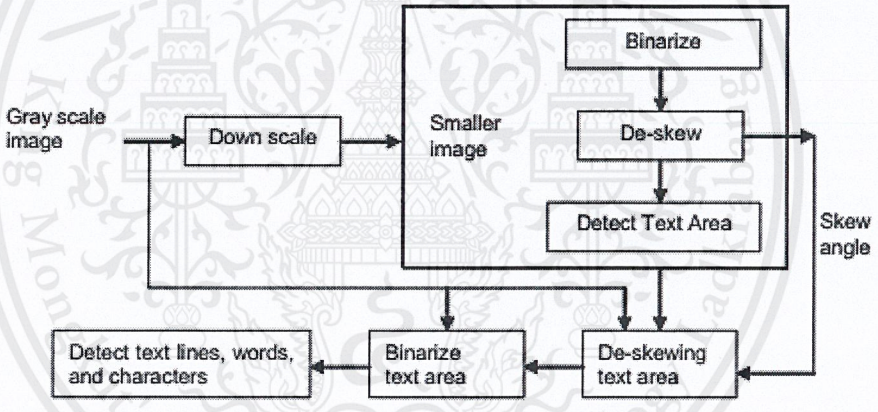


Figure 2.6: image processing steps

2.2.6 Mixed Thai-English Character Classification Based on Histogram of Oriented Gradient Feature [6]

The research proposes on how to classify Thai characters which have 2 processes.

The first process is to prepare the input data. This process has 4 steps. First, separate a character into nxn blocks. Next, within each block, a histogram of D directions is created. Thus, within each block, there are D numbers representing D directions. Then, each number in the histogram is computed by summing all the magnitudes of the gradient of all the pixels within the block n the represented direction of that number. Finally, all the histograms are stringed together and scale-normalized to make one descriptor vector for the input image.

The second process is Classification. This process separated into two processes: training and predicting. Both processes start by taking in vectors. For the training process, a set of descriptor vectors are gathered to be used for training Support Vector Machine. On the other hand, the predicting process takes the input vector and uses Support Vector Machine to predict the closest possible character.

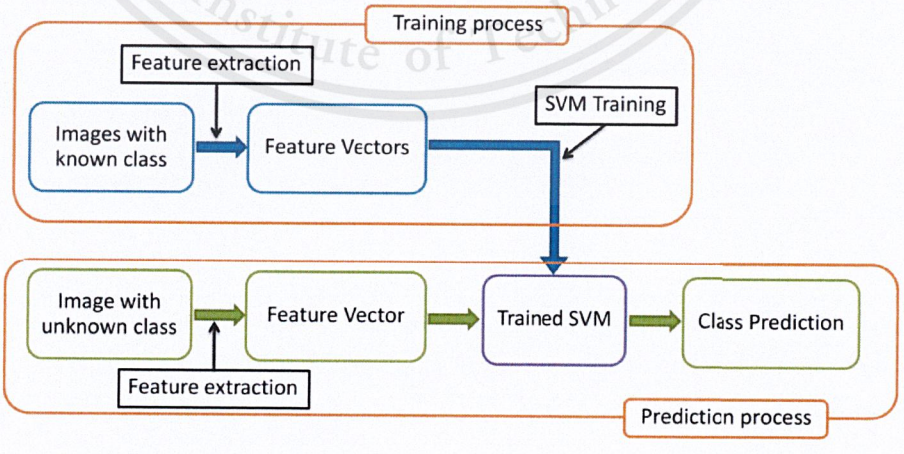


Figure 2.7: A typical classification system which usually separated into training and predicting (classifying) process.

Chapter 3

Background Knowledge

3.1 Optical Character Recognition (OCR)

OCR is a technique used to convert typed handwritten or printed texts into machine-encoded texts, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image.

3.2 Region of Interest (ROI)

ROI is a selected subset of samples within a data set identified for a particular purpose. ROI commonly used in a reference to a machine vision field of view.

Figure 3.1: Each area in the national ID card

3.3 Look-up Table

Look-up table is an array that replaces runtime computation with a simpler array indexing operation.Look-up tables are also used extensively to validate input values by matching against a list of valid (or invalid) items in an array and in some programming languages, may include pointer functions (or offsets to labels) to process the matching input.

3.4 Image Acquisition

Image acquisition is a process of the creation of photographic images.The process commonly includes the processing, compression, storage, printing, and displays of such images. The information will be converted by image sensors into digital signals that are processed by a computer and outputted as a visible-light image.

3.5 Feature Extraction

Feature Extraction is the process which represents the interesting part of the image. This technique is helpful when images' size are large.

3.6 Binary Image

Binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white, though any two colors can be used. The color used for the object in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit like 0,1 . The names black-and-white, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as gray-scale images.

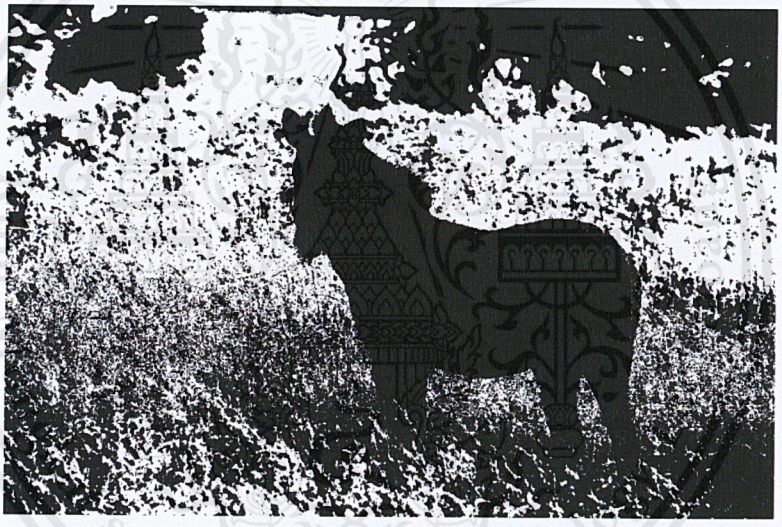


Figure 3.2: Example of Binary image [7]

3.7 Image Convolution

Convolution is a general purpose filter effect for images which apply matrix to an image and a mathematical operation comprised of integers. To calculate, we multiply the pixels and its neighboring pixels color value by a matrix.

3.8 Image Histogram

Image histogram is a graphical representation of the tonal distribution in a digital image. Each graph shows the numbers of pixels in each rows and columns of the image.

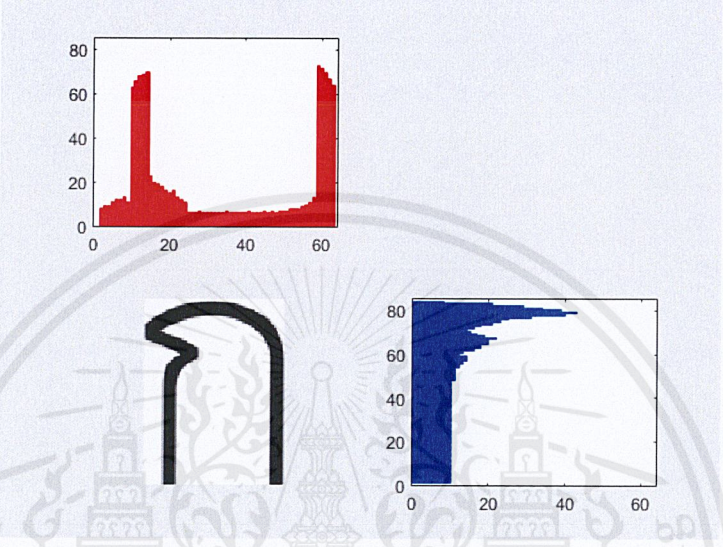


Figure 3.3: Histogram in x-axis and y-axis of character

3.9 Sum of Absolute Difference (SAD)

SAD is a measurement of the similarity between image blocks. It is calculated by taking the absolute differences between each pixel in the original block and the corresponding pixel in the block being used for the comparison.

Template	Search Image
2 5 2	2 7 5 8 6
4 0 7	1 7 4 2 7
7 5 9	8 4 6 8 5

Then we separate the search image's matrix into 3 3x3 matrices which are

Search Image 1	Search Image 2	Search Image 3
2 7 5	7 5 8	5 8 6
1 7 4	7 4 2	4 2 7
8 4 6	4 6 8	6 8 5

After that we compare each pixel of search image to template image and calculate the absolute differences for each pixel.

Absolute Difference 1	Absolute Difference 2	Absolute Difference 3
0 2 0	5 0 3	3 1 1
3 7 3	3 4 5	0 2 0
1 1 3	3 1 1	1 3 4

Finally, we sum up the number in each matrix and we will get 20,25 and 17. From these SAD values, it could be asserted that the right side of the search image is the most similar to the template image, because it has the lowest sum of absolute differences as compared to the other two locations.

3.10 Image Filtering

Image filtering contains noise removal, sharpening and deblurring.

Noise removal used with the dirty image (image with dots, speckles and stains). This method uses to remove speckles or dots in the image by taking mean or median values of neighboring pixels. The problems that can be occurred are blurring edges of the image.

Image sharpening is to enhance line structures or other details in an image. The image that will be received after this method is the original image with a scaled version of the line structures and edges in the image.

Image deblurring used to fix the blurred image which can occur when taking a picture while moving.

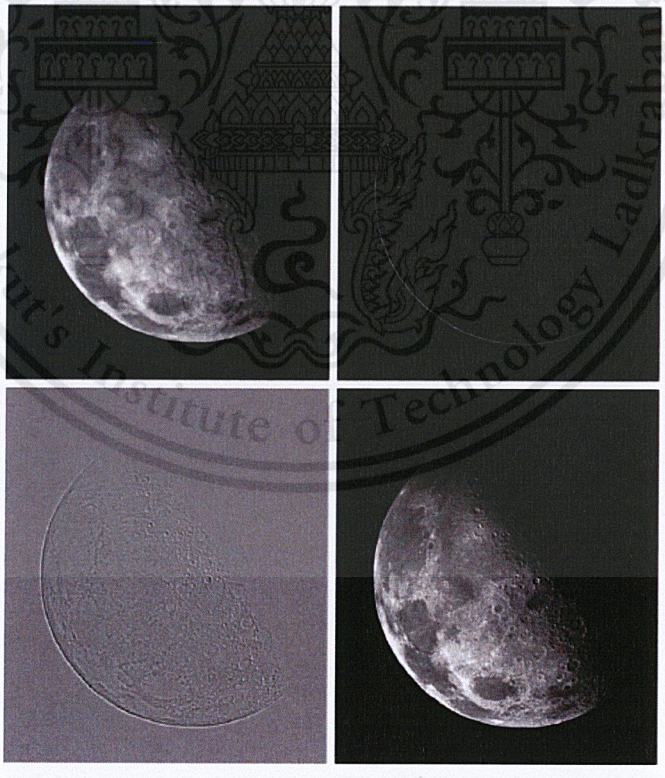


Figure 3.4: Example of image sharpening(Original image courtesy of NASA) [8]

3.11 Otsu's Method

Otsu's method is an automatically perform clustering-based image thresholding or the reduction of a graylevel image to a binary image.The algorithm will assume that image contains 2 classes of pixels which are background and foreground pixels.It will calculates the optimal threshold to separate the foreground from background.

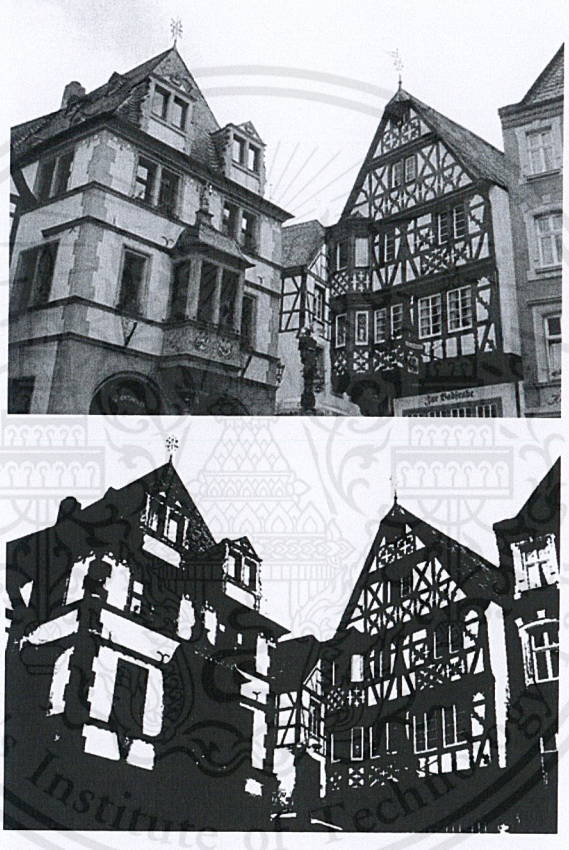


Figure 3.5: Otsu's method [9]

3.12 Morphological Image Processing

Morphological image processing is the a method of extracting borders in the image based on a shape. The value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with the size and shape of neighbors.

The basic morphological operation are erosion and dilation.

- Dilation is an adding of pixels to the boundaries of objects in an image. So, this mean it is the increasing of white area in a binary image.
- Erosion is an removing of pixels to the boundaries of objects in an image. So, this mean it is the increasing of black area in a binary image.

Figure 3.6: Dilation of binary images [10]

Figure 3.7: Erosion of binary images [10]

3.13 Histograms and Bayes Classification

This algorithm will find the percentage of the selected feature of each image by using a histogram to calculate. First we will find the probability of the information in each histogram to all information in all histogram which can be use as this expression

$P(A) = N_a/(N_a + N_b)$ for finding the probability of A histogram or area under graph A

$P(B) = N_b/(N_a + N_b)$ for finding the probability of B histogram or area under graph B

After we get the proportion of each histogram, we compute the density of the probability distribution of A along with variable x as $P(x|A)$ or P(x ∩ A)/P(A) and density of the probability distribution of B along with variable x as $P(x|B)$ or P(x ∩ B)/P(B).This mean that we are going to find the frequency among the information itself.

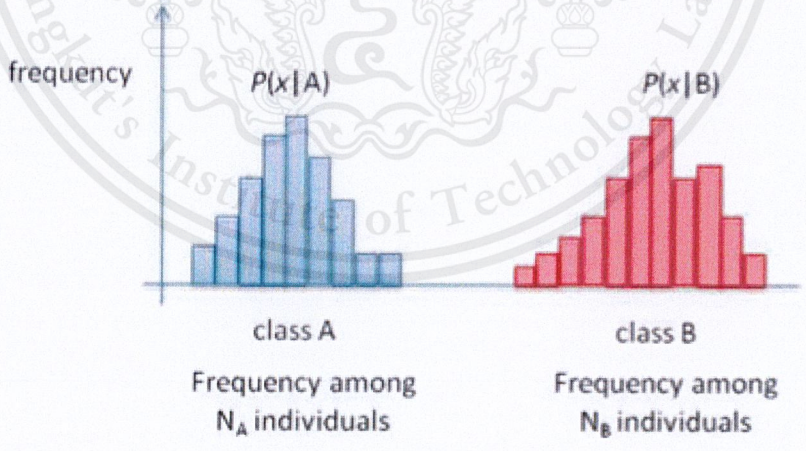


Figure 3.8: density of probability distribution of A and B [11]

But in image recognition, we are going to find the frequency among all every information. So, we find the proportion between probability of A and B to all information which are $P(A|x)$ and $P(B|x)$.

Then, we are plotting the $P(A|x)$ and $P(B|x)$ and use Bayes expression to find the intersected area of graph A and graph B.

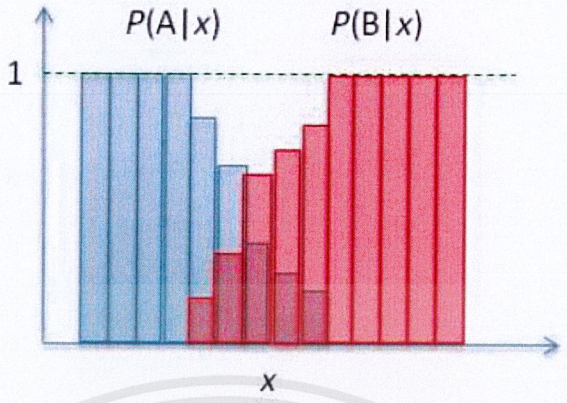


Figure 3.9: Finding intersect area under graph [11]

3.14 Minimum Bounding Rectangle (MBR)

MBR is a mathematics expression of the maximum extents of a 2 dimensional object or set of object within its 2D coordinate system.

MBRs are usually used as an indication of the general position of a geographic feature or dataset, for either display, first-approximation spatial query, or spatial indexing purposes.

Chapter 4

Requirement analysis

4.1 Requirement for Application

4.1.1 Functional Requirement

- Users can take a picture of an ID card.
- Users can decide to use the taken picture or take a new picture again.
- Users can crop the taken picture for better result.
- Users can import the image from the mobile photo library.
- Users can store the interpreted data.
- System shows the image of an ID card taken by the user.
- System interprets the pictures of information on the ID card into texts.
- System shows the user the information on the card after the process.

4.1.2 Non-functional Requirement

- User interface is easy to use.

- System supports English language.
- The system developed by Java language.
- The system uses OpenCV as a library for an image processing techniques.
- The system uses SQLite language for managing the database.

4.2 Use Case Diagram

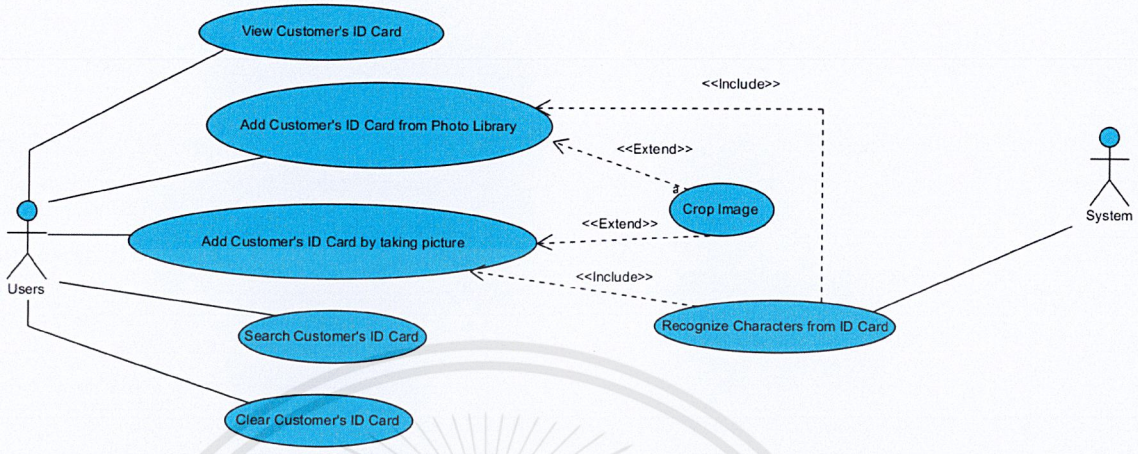


Figure 4.1: Use case diagram

4.3 Use Case Description

- **Take a picture of an ID card**

Actor: User

Pre-condition: -

Goal: To get a photo for the OCR processing.

Typical course of events:

User	System
1. User presses the take photo button.	
	2. System show camera mode page.
3. User moves the mobile phone to get the photo they want.	
4. User presses the take photo button.	
	5. System captures the picture.

Alternative Flow: -

- **Select the picture of an ID card**

Actor: User

Pre-condition: -

Goal: To get a photo for the OCR processing.

Typical course of events:

User	System
1. User presses the select the picture button.	
	2. System shows gallery mode page.
3. User selects the picture he wants.	
	4. System uses that picture.

Alternative Flow: -

- **Decide to Use Taken Picture**

  Actor: User

  Pre-condition: Take a picture of an ID card.

  Goal: To decide to use the taken image or not.

  Typical course of events:

User	System
2. User presses the accept button.	1. System receives the taken image.
	3. System saves the image.

  Alternative Flow: If user presses the cancel button, the system will let the user to take the picture again.

- **Crop taken ID card image**

  Actor: User

  Pre-condition: Accept the taken picture.

  Goal: To see the image that application will receive.

  Typical course of events:

User	System
	1. System receives the taken image.
	2. System shows the taken image.
3. User move and/or change size rectangle area to crop image.	
4. User press crop image button	
	5. System crop the image.

  Alternative Flow: If user presses the cancel button, the system will let the user to crop the picture again.

- **Interpreted the data**

  Actor: User

  Pre-condition: have picture an ID card.

  Goal: To decide to use the taken image or not.

  Typical course of events:

User	System
	1. System receives the image. 2. System interpreted data from the image.
	3. System show the interpreted data.
4. User edit than confirm the interpreted data.	

  Alternative Flow: If system can't interpreted data from the image, it will show error message.

- **Store the interpreted data**

  Actor: User

  Pre-condition: See the ID card information.

  Goal: To save the information from an ID card in database.

  Typical course of events:

User	System
1. User presses the save button	
	2. System stores the interpreted information to the database
	3. System responses that the information have been saved.

  Alternative Flow: -

- **See ID card information from database**

  Actor: User

  Pre-condition: Have store ID card information.

  Goal: To see the interpreted information from an ID card.

  Typical course of events:

User	System
1. User presses on the list of the stored ID card	
	2. System shows the interpreted information of the ID card that user select.

  Alternative Flow: -

4.4 Activity Diagram

4.4.1 Take a Picture of ID card

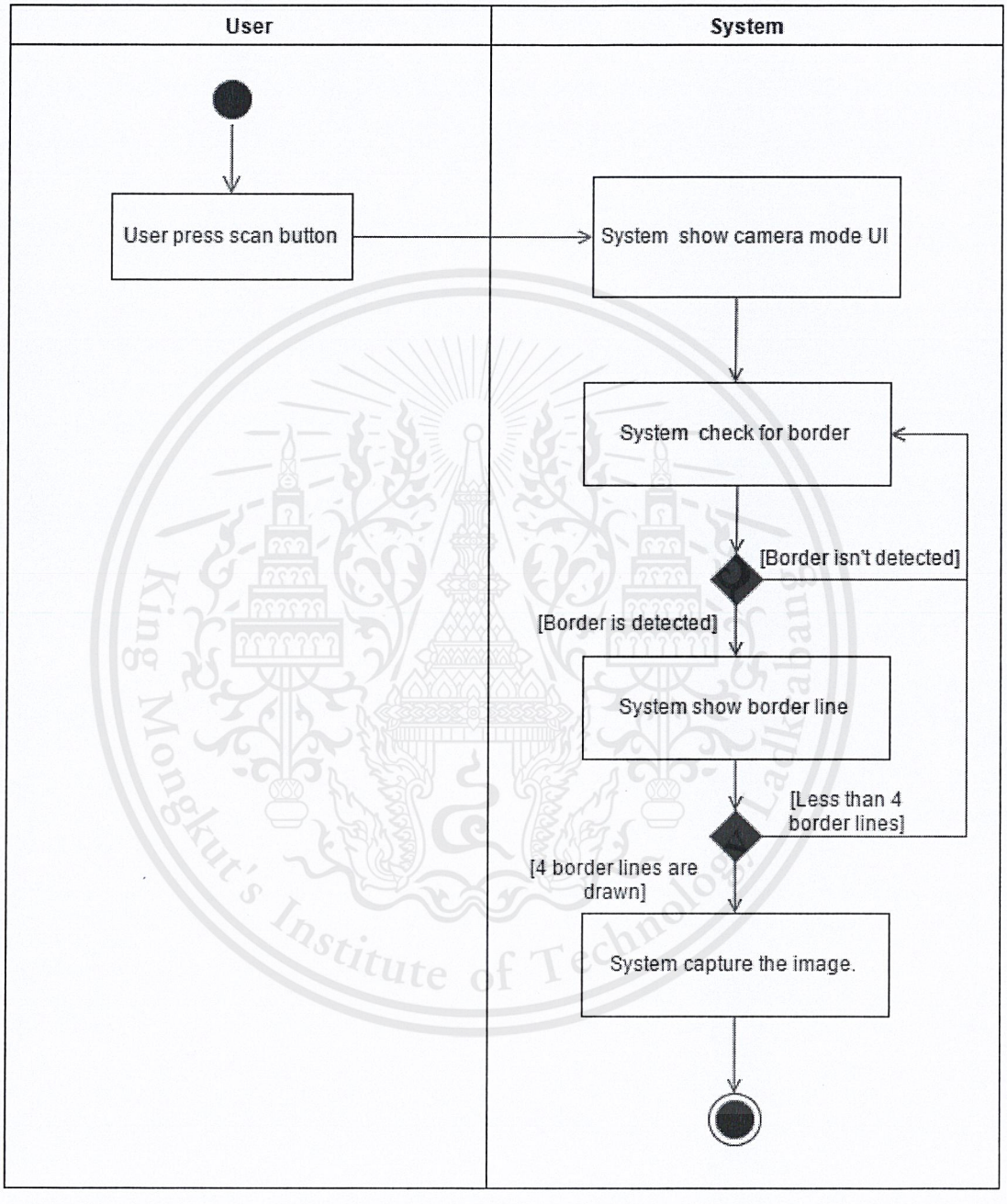


Figure 4.2: Activity diagram for take a picture of ID card

4.4.2 Decide to Use Taken Picture

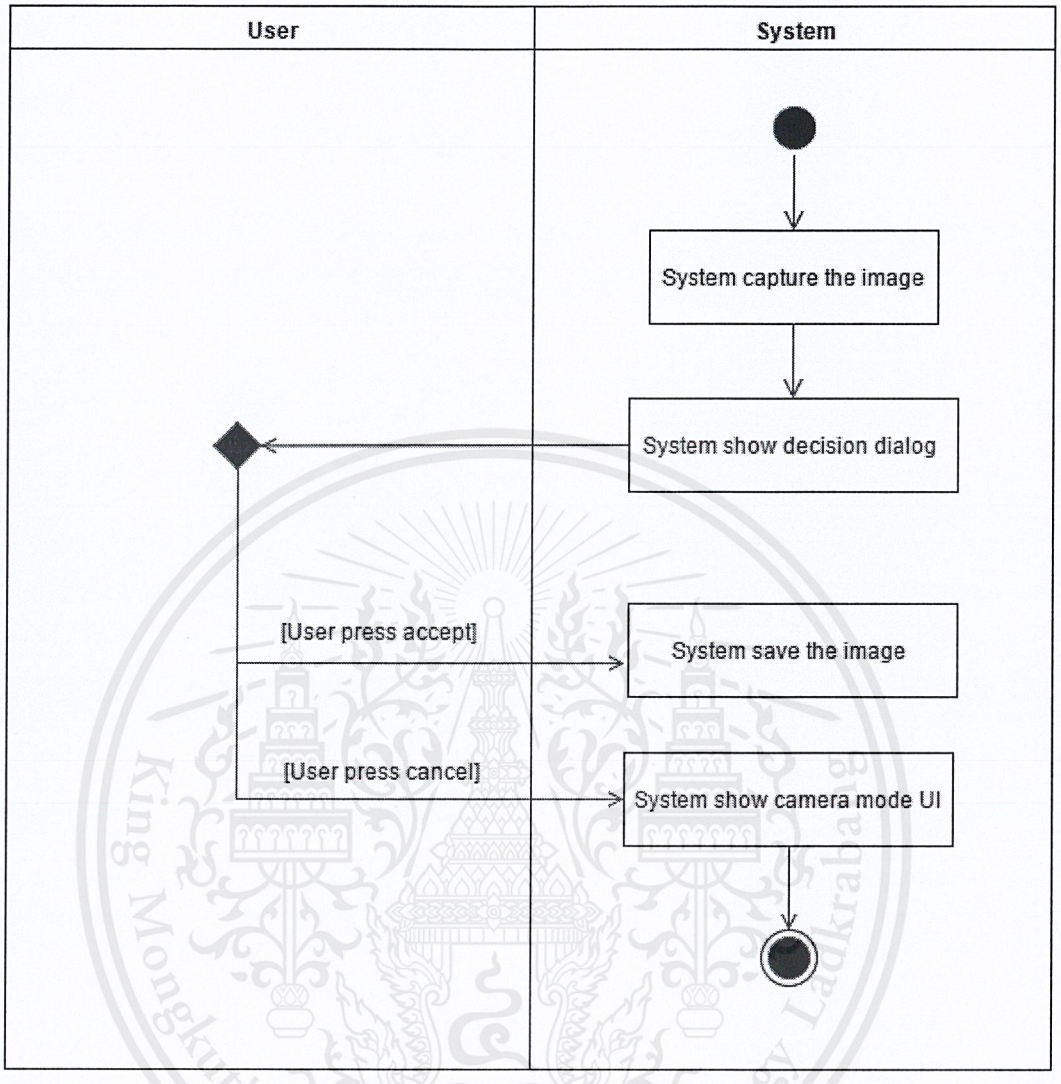


Figure 4.3: Activity diagram for decide to use taken picture

4.4.3 Store The Interpreted Data

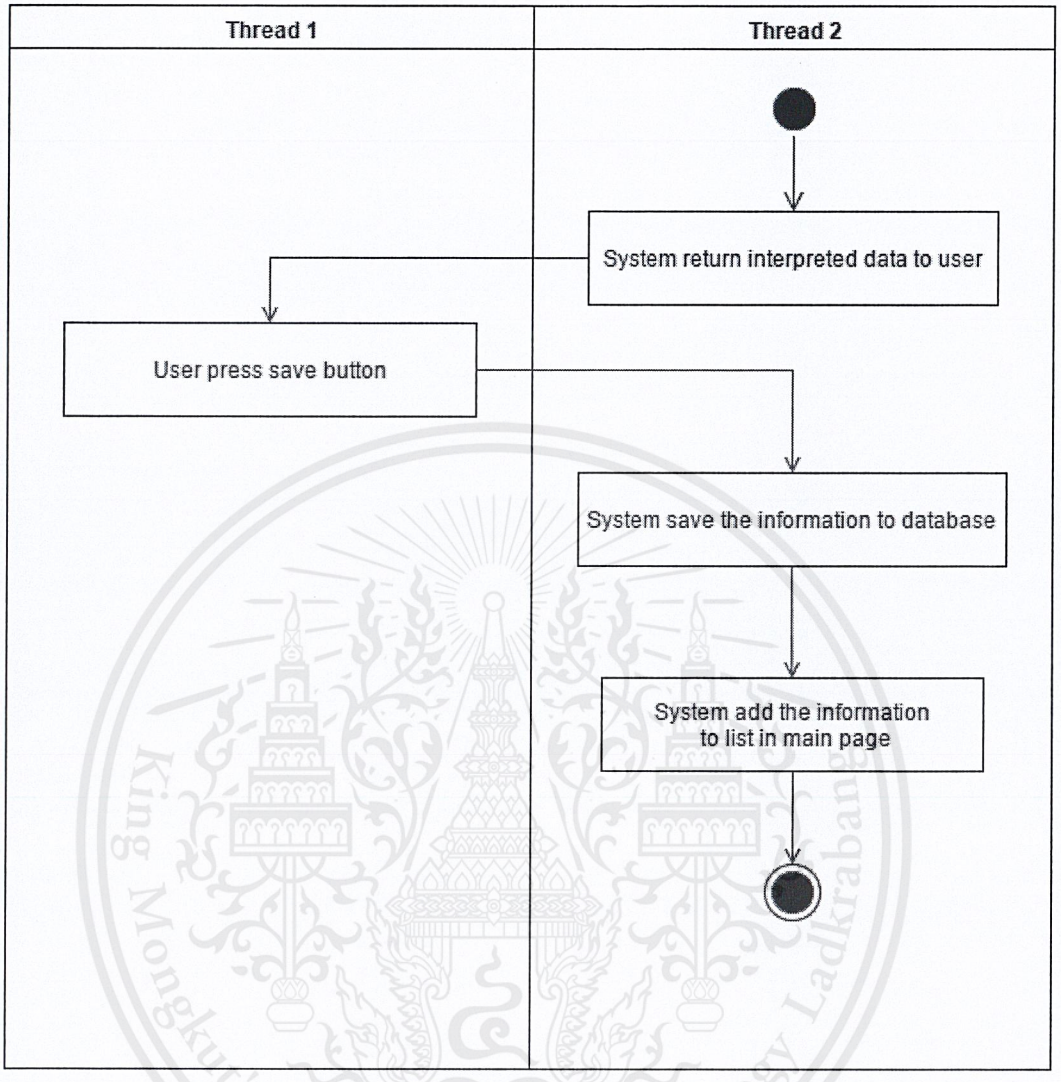


Figure 4.4: Activity diagram for store the interpreted data

See ID Card Information

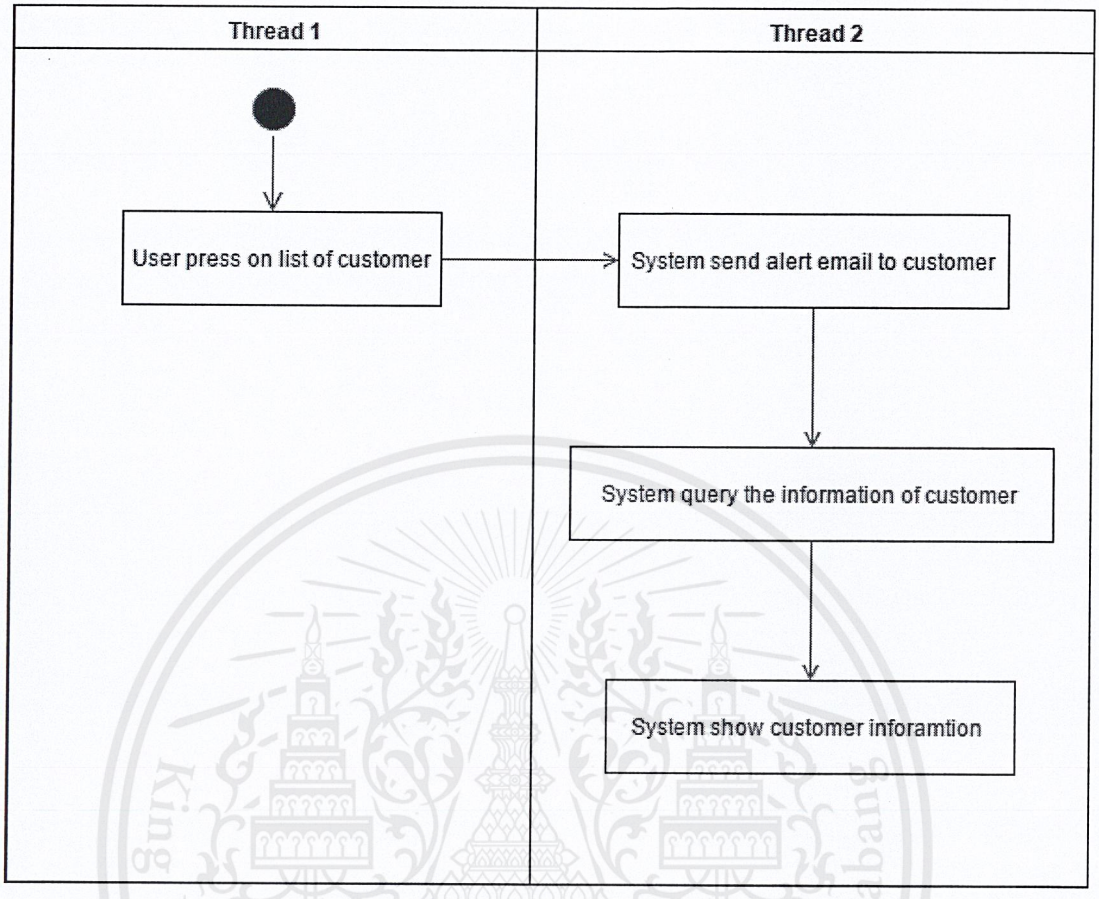


Figure 4.5: Activity diagram for see ID card information

Chapter 5

Software design

5.1 System Architecture

In our project, we separate the application into 3 parts which are the image processing method, mobile application and database.

5.1.1 Image Processing Method

Image Acquisition and prepossessing image

User is allowed to get the ID card image by using the mobile phone camera or importing the photo from the photo library.

For using mobile phone camera, the application will show the line to the user which will guide the user where should the ID card be, but to get more accuracy, we also make an area for the portrait and smart chip to let the user knows more exactly where the ID card should be. When the user thinks that the ID card is in the appropriate position, the user can touch the screen to take a picture.

If the user decides to import the picture from the photo library, the user can press the import button. The application will access the photo library in a mobile phone and let the user picks the image as an input image.

When the application have gotten the ID card image, the image will transform into a binary image using Otsu's method in order to let one knows which one is a foreground or a background.

Assign Region of Interest(ROI)

After we get the input ID card image, we divide the image into many areas which are ID number, name, surname in both Thai and English, date of birth, card holder portrait, address, date of issue, and date of expiry.

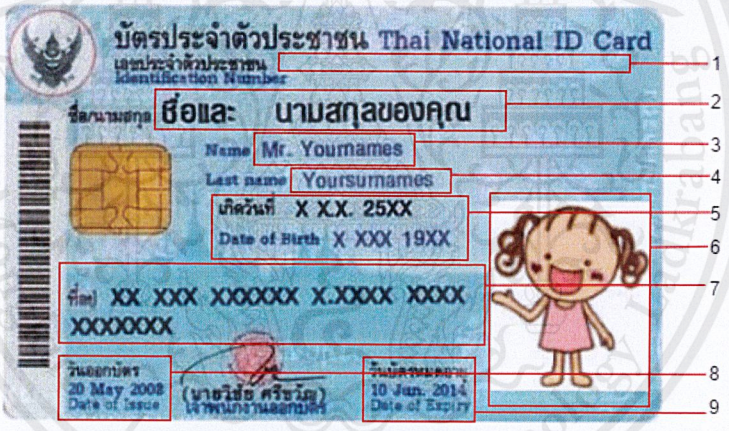


Figure 5.1: ID card border

Separate Line

For some areas like address, date of issue, and date of expiry contain more than one line of information. So, we need to divide the information in the area to one line, to let the system be able to separate the word and character easier later in the process.

Separate Word

We will transform each area of information into histograms to separate the words. Histogram will show us the density of black pixels in each row and column.

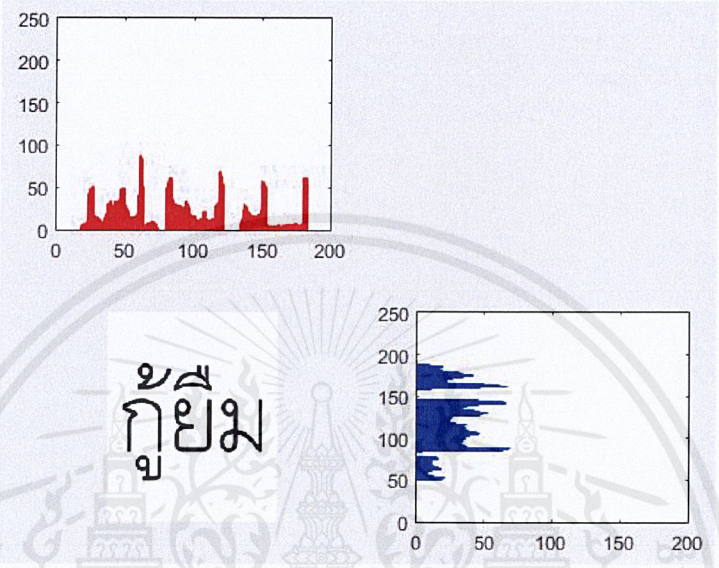


Figure 5.2: Example of histogram for word classification

From figure 5.2 ,we can see that the word have 2 axis of histograms, the blue one and the red one. The red histogram shows the vertical column pixel. The blue histogram shows the horizontal row pixel. The color in the histogram shows the area that contains the black pixels which mean that there is a character in those area. We will use the white area of the column and row in the histogram to separate the information into words.

Separate character

To separate the characters from words, we use a histogram of the word to detect where the characters are. When we know where each character is, we will collect the width and height of each character and calculate the average width and maximum height to use as the condition to separate the character again when we face the overlapped character problem.

Classify Character

We classify the types of Thai characters into 3 types which are top-layer, middle-layer, and bottom-layer character. Thai characters play an important role in a tonal language with 5 tones. These proposed layers consider the specific consonants vowels, and tone marks in a syllable according to the pronunciation. Top-layer character containstonal marks and some vowels that put above alphabet. Middle-layer character contains all alphabets and the vowels which put in the same level as the alphabet. Bottom-layer character contains vowels that put under the middle character.

For all none middle-layer character, top-layer and bottom-layer character, when we have already separated the character from words, we will find the histogram of that character and compare the histogram information with all the data set of all characters we have stored in the database.

For the middle-layer character, after we separated the character from the word, we will divide the areas of the image of the character into 4 areas and find all the histograms of all areas. All characters will have a specific histogram in each area. Then we will find the intersect area of each 2 areas by finding the lowest value between each histogram. We will use Markov chain to compare the intersect area and the area of the histogram of each character to find the proportion between one area to another area. We will find the intersect area under a graph between each histogram by using Histograms and Bayes classification. After we get the intersect area of all areas, we will find the proportion by comparing between the intersect area to the area under the graph. After that we will store the proportion in the database as the characteristic of each character.

We will use Sum of Absolute Difference (SAD) to classify the features of the character with the characteristic of Thai character that we stored in the database.

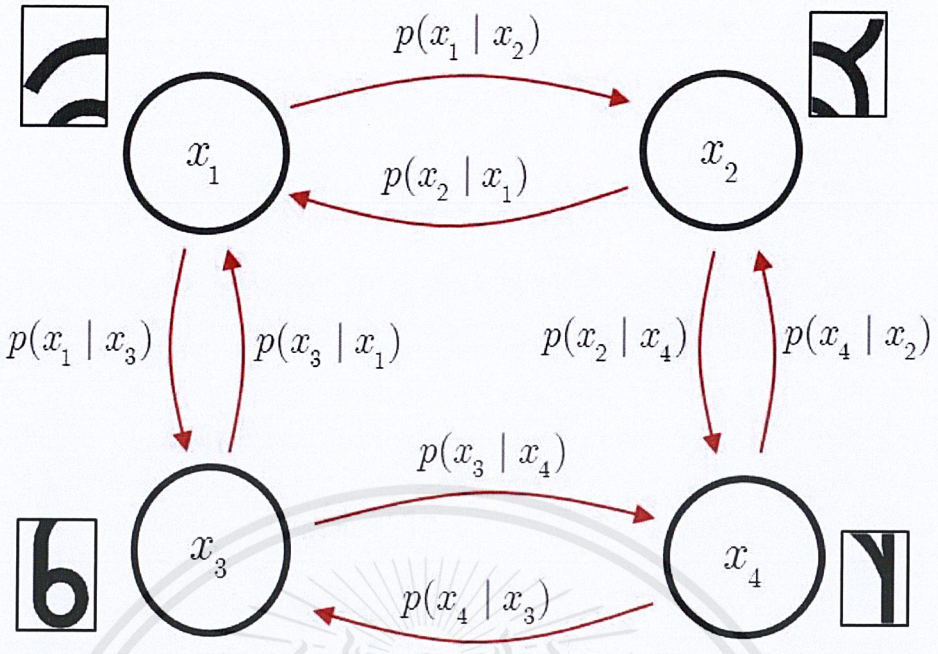


Figure 5.3: Using Markov chain to find proportion between each area

(Under the patent of Dr. Montri Phothisonothai)

5.1.2 Mobile Application

In mobile application, the main objectives are managing the information and getting the input image of an ID card. In the application, we also include the login system to verify user who take in the information of ID card.

Login page will require the user name and password for each user that wants to access the application for the security for all people who will get the taken ID card image. All users will need to register and verify themselves before accessing. When the user logins into the system, the application will check the user name and password in the database.

Scanning ID card page is used to get in the picture of the ID card by using the mobile phone camera. To add the ID card information, user needs to open the camera mode to take the picture of the ID card. Camera mode will show the user the area of the ID card. The borderline will show where the ID card should be.

To make the position of the ID card more accurate, system will show where the area of the portrait and smart chip should be by locating the borderline. Then the user is allowed to take the picture of an ID card oneself. After the application has gotten the image, system will show the image to the user and ask whether the user wants to use or not. If the user confirms ,the image will be prepared for the next step, otherwise, the application will ask the user to take the ID card picture again. After the system processes the information of the customer's national ID card, the system will require the customer's email to verify this information to send the email to remind the customer when will the user uses this information.

Result page is a page that shows the user the interpreted data. The interpreted data is a result from the image processing method of the input image. Result page will show ID number, first name, surname in both Thai and English, date of birth, card holder portrait, address, date of issue, and date of expiry. In this page, users are requested for a customer email to be input.

Scanned information page is a page that contains the information about ID card which interprets from ID card image that users scan from adding the ID card picture page after passing through the image processing process. In this page, the user can see the information of each ID card that the user has saved in the result page. The information that will show after the user requests for the information are ID number, first name and surname in both Thai and English, date of birth, card holder portrait, address, date of issue, and date of expiry. Every time, the user accesses the information of the customer, the system will automatically send the email to remind the customer that his information has been accessed by who, at what time, and for what reason.

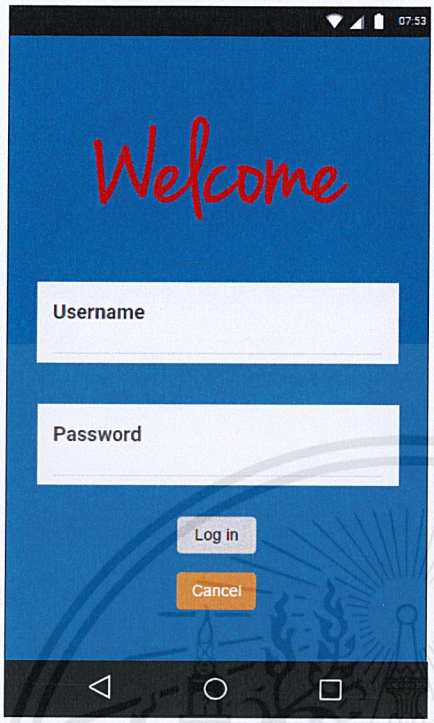


Figure 5.4: Login page

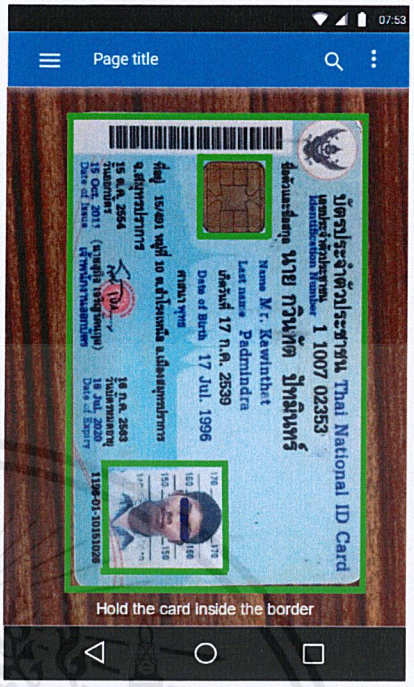


Figure 5.5: Scanning ID card page

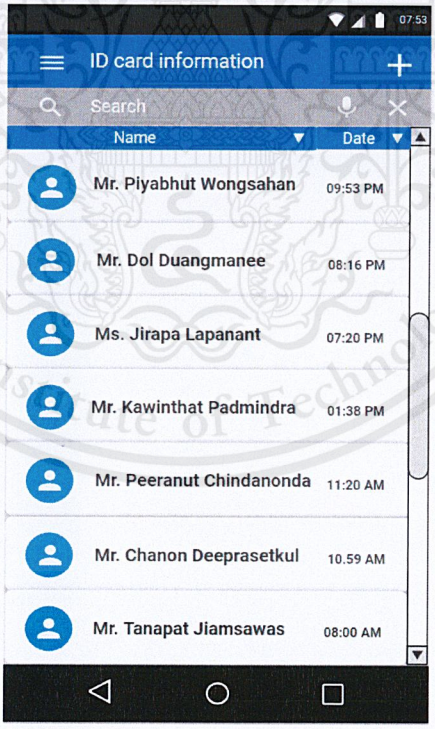


Figure 5.6: Scanned information page

5.1.3 Database

Our project consists of 3 tables of database which are user information,[4x40] matrix of each character, and customer ID card information. The matrix of each character database is a local database which will store in the mobile phone. Other two databases which are customer ID card information database and user information will be in a server database.

User information database

This database will contain the information that required for the login system for the user. Database will contain a user name and password that will be used for comparing with the one that user has put in the login page to access to the system.

Customer ID card information database

This database will contain all information that the user has scanned from the customer ID card including ID card number, first name, last name, address, date of birth, date of issue, and date of expiry. Whenever the user scans the customer ID card, the information will be stored in the same database as the other users do. We will use the ID number as a primary key in this database, this mean that the information that will receive can't have a same ID number.

Character trained set database

In this database, we will separate the trained set into 3 types which are Thai characters, English characters and numerical characters. Each database will contain the Markov values of each character as a characteristic of that character. We will use those database to compare with the input character to find the SAD

value and choose which character is the input character.

5.2 Class Diagram

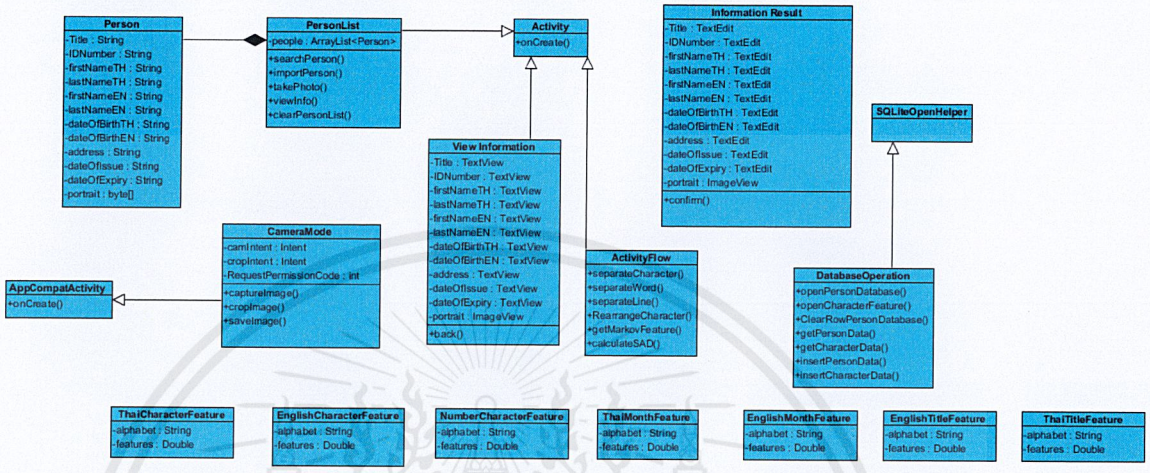


Figure 5.7: Class Diagram of the mobile application

Chapter 6

Development

This project is developed by using Android Studio as a main development framework for the mobile application, OpenCV as a library for image processing method, and Matlab as a platform for testing algorithms before applying them in Android Studio.

6.1 Image Processing Algorithm

6.1.1 Preprocessing Image

We assume that the input image is the perfect perspective and position. So, we use the ID card picture from the scanning machine as an experimental image.

For preprocessing step, we convert the ID card picture into a grayscale image and binary image. The algorithm that we use for converting to binary image is Otsu's method. Otsu's method is also use as a build-in function so we don't need to implement the method.

6.1.2 Assign ROI

After we got the binary image of an ID card, we separate the ROI by calculating the area with a ratio of the width and height. Each area will have it's own ratio of width and height, this means that one area will have the ratio of 4 points of the border.

The ratio that we use is calculated from the point where the area starts and the point where area ends. To calculate the ratio, we will compare the starting point of the area in width and height to the width and height of ID card.

We apply the ratio to every ROI in the ID card but there is a problem in the real test if users take a picture that is not in the perfect position.

6.1.3 Histogram

We interpret all ROIs into histograms which are x-axis and y-axis histogram.The histogram will interpret from the black pixel. So, we create the loop for detecting the black pixel in column for x-axis histogram and the loop for detecting the black pixel in row for y-axis histogram. We will get 2 line graphs of x-axis and y-axis.

6.1.4 Separate Word

After we get the line graph, we set the minimum amount of pixels that will indicate that there is a character in that column. We create the loop for checking the white column and black column. If we meet the black column, we will mark it as the starting point of the word. We loop until we meet the white column.

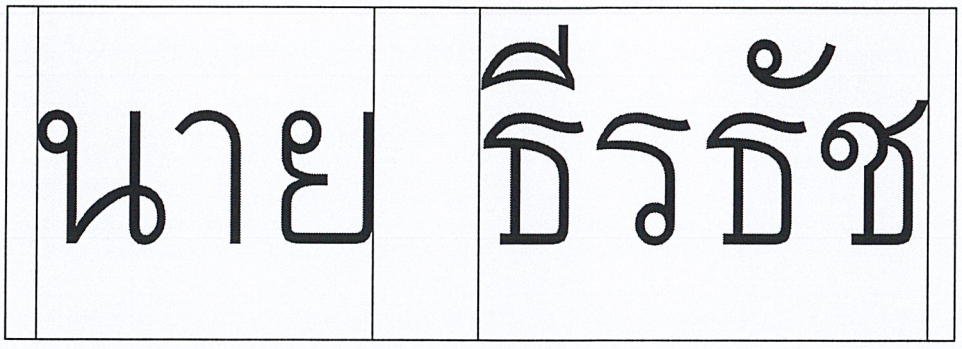


Figure 6.1: Second separating (Vertical)

6.1.5 Separate Character

After we get the words, we also interpret them into a histogram. On the first try, We use the separate word algorithm to cut the word into characters, but the problem is when the word has some characters that overlap with another. We fixed it by separating the horizontal line first, then separate the vertical line.

Figure 6.2: First try for separating character

To make the separating more efficient, we create a new way to separate the character. The algorithm will collect the value of all characters namely, mean width, and maximum height in order to use to calculate later.

At first, we use the x-axis histogram to separate the word in vertical line. While the word is being separated, we collect each width of character and find their Mean (Average value of width). We separate in a vertical line until the ending of the word.

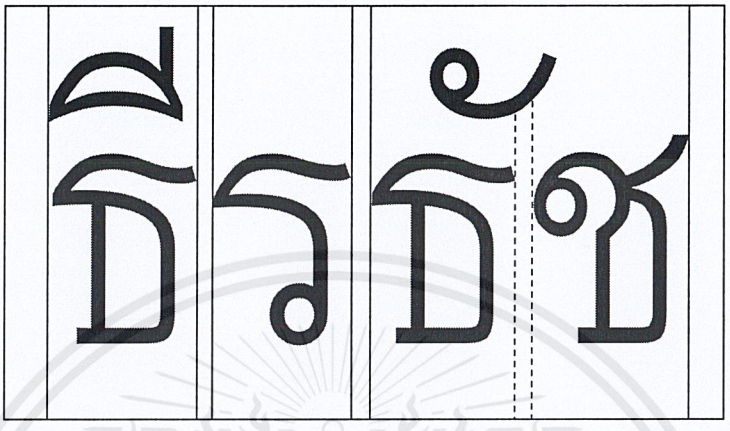


Figure 6.3: First separating (vertical)

Next, we use the y-axis histogram to separate the word in a horizontal line. While the word is being separated, we collect each height of the character and find their maximum height. We separate the word in horizontal line until the max height of the word.

Figure 6.4: Second separating (horizontal)

After we get the mean of width and maximum of height of the word, we separate in a vertical line again, however we will separate only the word that includes in this following conditions, the width of word is more than mean width, or the height is less than 80 percent of maximum height.

Figure 6.5: Third separating (vertical)

6.1.6 Separate Line

We did the experiment for every ROI and faces the problem that there are some ROIs that will contain more than one line of information. All of the ROIs that will contain more than one line of information like address, date of issue, and date of expiry need to extract the information in each line first to apply with separate word and character later.

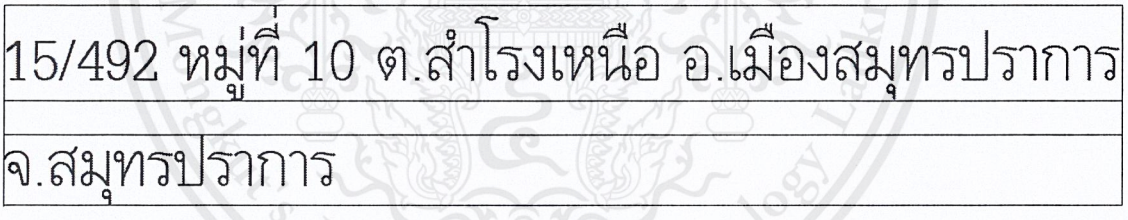


Figure 6.6: Line separating

6.1.7 Separated Character Sorting

About this algorithm, we process after the result of the separate character process. The result that was created from the separate character process will sort by top-left of the word to bottom-right which will disorder the character that we want to print out.

The algorithm that we use to sort the separated character is we will have a stack, mark, and output list. After we finish the character separation, we will get the list of the character of the word. At first, we assign the false value to the mark. After that we loop the character list.

If we found the top-layer character and the mark is false, we put that top-layer character in the stack but if we found the middle-layer character and the mark is false, we put that middle-layer character into the stack and change the mark value to true.

If we found the top-layer character and the mark is true, we pop out every character from the stack, put them into the output list, change the mark value to false and put that top-layer character into the stack, but if we found the middle-layer character and the mark is true, pop everything out of the stack, put them into the output list, and put that middle-layer into output list.

If we found the bottom-layer character and the mark is true, we pop out the top character of the stack , put into the output list, put the bottom-layer character into the output list, pop out everything in the stack into the output list, and change the mark value into false.

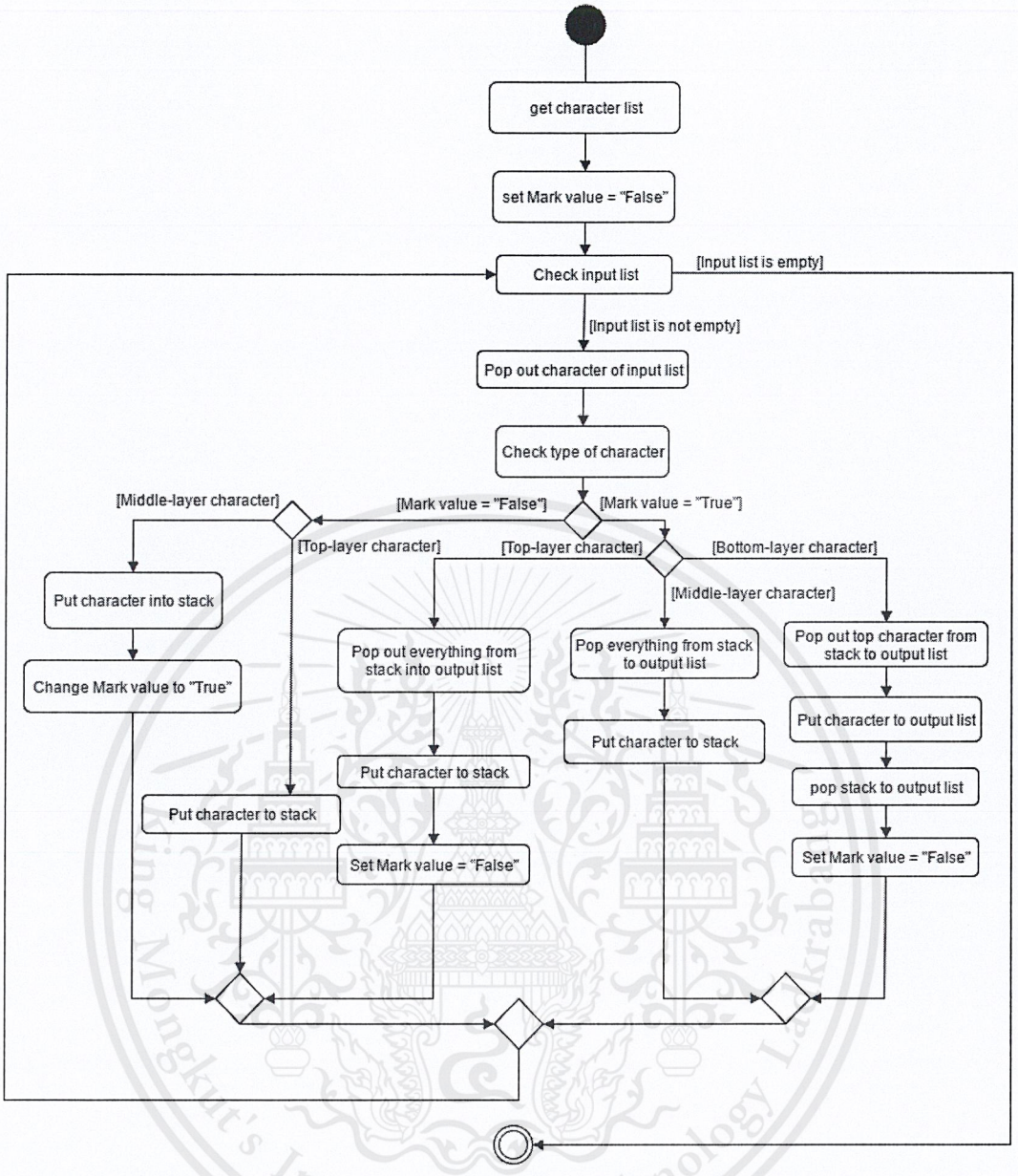


Figure 6.7: The proposed algorithm for Thai characters sorting

6.2 Android Studio

Android studio is a development environment for developing a mobile application on Android. We use it to develop our mobile application.

The purpose of the mobile application is to allow the user to manage the data of an ID card and give the security of the customer information. Therefore, the pages that we need to develop first are the login page, customer information page, and ID Card scanning page.

Log in Page

The purpose of this page is to authorize users who access the system to ensure that the customer ID card information is being secured.

Scanning Page

The purpose of this page is to let the user gets the image of an ID card by a mobile phone camera to use in the next processes.

At first, the user will get the ID card image by using the mobile phone camera. The application will show 4 corners of an ID card border to help the user to know where the ID card will be. If each border is detected, the line will be drawn between each corner. After all borders are drawn, the application will automatically capture the image of the ID card inside the border.

On the first try, we use Canny's edge detector as a technique to detect the border of the card. The result is not stable, sometimes, it isn't in the right position and angle.

So, we change the technique into finding contours. The result is more

accurate, but it needs a lot of memory to use which means that it causes phone camera to lag.

Because the process of both mentioned automatically capture image is not quite accurate and due to the ease of use in terms of user-friendliness, so we change the scanning page from the automatically capture into the manually capture with a guided border. Application will provide the line that guides the user to know where the ID card, portrait, and smart chip should be. When the user thinks the ID card is in the appropriate position, the user can touch on the screen to take a picture.

Image Processing

In this step, we implement the algorithm what we have developed in Matlab to interpret the data from the ID card image that received from scanning page. We develop the performance of the algorithm to be better for using with the mobile phone by using OpenCV.

In developing OpenCV, we focus on using the build-in function to reduce the process time in mobile devices when compare with the desktop with a higher performance.

At first, we develop the separate word and character part by testing the algorithm with the scanned image of the name and surname to see that our algorithm works or not.

After we are able to separate the character correctly, we try to test to all ROIs and found that some ROIs have more than one line of information. So, we develop the separate line part which will help us to divide each line of information.

6.3 Database

6.3.1 User Information Database

This database, we collect all users' id and password. This two information will use for an authentication in the log in page and register page. We develop the database along with the log in and register page.

6.3.2 Customer ID Card Information Database

We develop this database by putting the information of the customer ID card that we received from the image processing method. By collecting all images according to the information which we want and interpret them into computer data. The computer data that we already classified and interpreted will be collected into the database.

6.3.3 Character Trained Set Database

First we create images of every character that are able to be in the national ID card namely, Thai characters, Thai exceptional case character, English capital and small letter, number, and some other symbol. Some Thai exceptional case characters have some problems that we need to create by using Photoshop. Then we apply our character classification process to all of the images we prepared. We will get the Markov values of all characters and store them in the database.

6.4 Experiment

Markov Chain Character Value Experiment

In this experiment, we will create the table of all values of each area of all characters namely $P(x1|x2)$, and so on. All eight values of each character will show the characteristics of that character. We will use these values to calculate the SAD values for comparing the character.

At first, some characters' Markov value are infinite due to some areas of the character is empty when we put in the function $P(x1|x2)$ which will divide by 0. For the solution, we fix it by changing the values of all empty areas to the lowest value to make $P(x1|x2)$ not becoming an infinite number.

SAD Value Experiment

After we get the character of the input information, we find the Markov values of the input character and then compare each Markov value for each character in the trained set. The character in the trained set with the least SAD value will be selected as the result of that input character.

ROI Separation Experiment

In ROI separation experiment, we will get the picture of the national ID card , but there is a problem of watermarks on the national ID card which make us unable to use the camera to get the image. So, we test this process by importing a picture from the photo library. The result that should come out is all ROIs that contain all information that we need to extract.

Word Separation Experiment

We take an experiment of the word separation by creating the word and use it as an input of this process, we try to create not too small character so that our process won't cause a problem of a small size character and make the result goes wrong.

Character Separation Experiment

For this experiment, we want to see that the separated character is correctly separated or not. Moreover, the order of the character in the word must also be correctly ordered.

Character Classify Experiment

In character separation experiment, we will test our classification process accuracy to see the result whether it matches the input character or not.

At first, English and numerical character is 100 percent correct, but there is a problem on Thai character classification. In Thai character, if the font size is lower than around 30x30 pixels, the accuracy of the classification will be less than 50 percent, but if we use the test sets which the font size is more than 100x100 pixels, the accuracy will become 100 percents.

So, we fix the accuracy problem by creating a new trained set for Thai character that is around 30-50 pixels to let our application be able to classify the small Thai character.

char	p12	p21	p13	p31	p34	p43	p24	p42	type
ก	0.817313	0.619981	0.815415	0.415536	0.831818	0.534519	0.5	0.478261	M
ข	0.70804	0.67404	0.654359	0.143632	0.738433	0.780502	0.211483	0.969461	M
ฃ	0.729869	0.661755	0.883774	0.108227	0.715796	0.82966	0.115041	0.987238	M
ค	0.902716	0.769779	0.713712	0.67134	0.810072	0.512795	0.612466	0.351477	M
ฅ	0.818728	0.760127	0.751487	0.632453	0.915012	0.485599	0.59642	0.349175	M
ฆ	0.797258	0.494597	0.718067	0.64236	0.706097	0.819146	0.873286	0.702574	M
ง	0.031294	1	0.060811	0.818182	0.816403	0.633971	0.441799	0.814803	M
จ	0.740934	0.828634	0.921643	0.357267	0.674541	0.908872	0.24356	0.946789	M
ฉ	0.789474	0.548506	0.61651	0.727043	0.471084	0.943833	0.752303	0.888002	M
ช	0.807443	0.792932	0.621013	0.34137	0.732723	0.500502	0.448946	0.547847	M
ซ	0.815936	0.865889	0.567076	0.362308	0.727972	0.561173	0.434976	0.556952	M
ฌ	0.664212	0.601005	0.56962	0.651207	0.606257	0.994446	0.628929	0.816514	M
ญ	0.59534	0.467772	0.676041	0.60676	0.60168	0.952916	0.48655	0.674593	M
ฎ	0.689486	0.622523	0.620402	0.823963	0.567991	0.838905	0.604536	0.607	M
ฏ	0.688954	0.626337	0.596311	0.815318	0.567702	0.845819	0.630771	0.624872	M
ฐ	0.764614	0.559904	0.843337	0.395054	0.612264	0.711957	0.293261	0.533072	M
ฑ	0.884294	0.694894	0.795021	0.316952	0.94612	0.507945	0.705843	0.746938	M
ฒ	0.653607	0.465994	0.800701	0.755298	0.608021	0.955533	0.656687	0.780012	M
ณ	0.527002	0.335076	0.691467	0.552491	0.422708	0.984482	0.392924	0.728204	M
ด	0.870923	0.868081	0.794743	0.75524	0.823259	0.55226	0.711376	0.500528	M
ต	0.808856	0.843217	0.805996	0.742664	0.926995	0.643034	0.702801	0.551566	M
ถ	0.815994	0.619395	0.695127	0.708339	0.79303	0.555722	0.714142	0.372783	M
ท	0.716131	0.71562	0.817201	0.567452	0.942761	0.433333	0.621212	0.410913	M
ธ	0.811408	0.638254	0.782629	0.515503	0.634543	0.546066	0.839141	0.862378	M
น	0.90924	0.481196	0.686813	0.550763	0.397598	0.97432	0.585846	0.947454	M
บ	0.5	0.232759	0.708508	0.527486	0.558232	0.982823	0.643216	0.708089	M
ป	0.197578	0.192803	0.410398	0.623165	0.643794	0.981104	0.627724	0.614772	M
ผ	0.544558	0.287158	0.853225	0.855674	0.520476	0.982766	0.799868	0.794146	M
ฝ	0.200097	0.218811	0.419673	0.829814	0.589585	0.994888	0.758319	0.707683	M
พ	0.695213	0.794248	0.748052	0.619635	0.773639	0.535236	0.799857	0.763226	M
ฟ	0.420011	0.670695	0.289573	0.548872	0.565085	0.730493	0.709652	0.772856	M
ภ	0.73474	0.666472	0.587028	0.680935	0.827686	0.531084	0.5	0.250882	M
ม	0.515625	0.244548	0.723288	0.867876	0.501278	0.996443	0.753962	0.592388	M
ย	0.574162	0.234379	0.83648	0.742531	0.601455	0.981145	0.792808	0.594733	M
ร	0.624116	0.640446	0.518562	0.11082	0.613606	0.689916	0.092612	0.5	M
ฤ	0.752447	0.572162	0.818134	0.484226	0.879273	0.673604	0.372526	0.366655	M
ล	0.760249	0.961386	0.623289	0.758832	0.848438	0.568327	0.68135	0.474061	M
ฦ	0.69176	0.611346	0.673832	0.443269	0.879814	0.633174	0.204919	0.198121	M
ว	0.602033	0.881108	1	0	0.574288	0.826991	0	1	M
ศ	0.749385	0.712825	0.589375	0.682025	0.794571	0.527004	0.59051	0.321942	M
ษ	0.736276	0.6346	0.732664	0.532016	0.726247	0.699953	0.657946	0.752686	M
ส	0.679954	0.882318	0.568213	0.737068	0.789865	0.613307	0.631583	0.490574	M
ห	0.595865	0.737947	0.719649	0.553085	0.814357	0.461674	0.527664	0.482041	M
ฬ	0.366716	0.830506	0.30737	0.504814	0.878795	0.623127	0.735224	0.718872	M
อ	0.812052	0.705759	0.671497	0.825759	0.73345	0.825394	0.71881	0.5717	M
ฮ	0.69409	0.766828	0.577873	0.751243	0.804299	0.612003	0.824117	0.532916	M

Figure 6.8: Thai alphabets representation by means of Markov value

char	p12	p21	p13	p31	p34	p43	p24	p42	type
A	0.444268	0.5	0.395832	0.731142	0.419592	0.723292	0.85789	0.901058	M
a	0.693119	0.743832	0.675185	0.754478	0.752707	0.799047	0.683291	0.69662	M
B	0.726499	0.651657	0.897543	0.870627	0.81975	0.860438	0.688729	0.668488	M
b	0.836818	0.402461	0.884818	0.905712	0.456476	0.852992	0.710603	0.623894	M
C	0.645436	0.633013	0.901915	0.917381	0.64436	0.51909	0.820316	0.637192	M
c	0.708842	0.630635	0.926949	0.901487	0.667958	0.548056	0.839667	0.630243	M
D	0.671378	0.579941	0.817426	0.817306	0.876901	0.892899	0.66239	0.582701	M
d	0.376879	0.814213	0.439252	0.864781	0.835911	0.83118	0.634284	0.692091	M
E	0.767078	0.683058	0.896258	0.869124	0.833876	0.811292	0.768924	0.686955	M
e	0.870679	0.861227	0.900919	0.760557	0.863043	0.560723	0.888657	0.676493	M
F	0.788238	0.701299	0.897103	0.871934	0.98118	0.29429	0.845458	0.232126	M
f	0.54182	0.328393	0.771997	0.88016	0.5	0.155088	1	0.164892	M
G	0.709891	0.579713	0.852193	0.899426	0.512564	0.907196	0.586311	0.802924	M
g	0.653767	0.702162	0.661415	0.675422	0.782294	0.752825	0.702018	0.710536	M
H	0.746409	0.73956	0.941686	0.898022	0.942884	0.894841	0.715731	0.705754	M
h	0.917405	0.3825	0.878859	0.784286	0.298217	0.632219	0.638262	0.632193	M
I	0.591274	0.71516	0.835144	0.820521	0.877173	0.865692	0.582538	0.707764	M
i	0.959885	0.99851	0.968208	0.99851	0.969444	1	0.961111	1	M
J	0.050524	1	0.038866	0.493868	0.931549	0.712487	0.528794	0.629979	M
j	0.9625	1	0.998555	0.997114	1	1	0.961111	1	M
K	0.872291	0.560667	0.874328	0.825026	0.671565	0.704047	0.808978	0.577698	M
k	0.506109	0.307758	0.66008	0.928347	0.423685	0.814967	0.81668	0.679205	M
L	1	0.026715	0.842308	0.842308	0.019656	0.5	0.799699	0.543458	M
l	0.443478	0.426471	0.855392	0.855392	0.787299	1	0.561851	0.686275	M
M	0.735355	0.757154	0.74051	0.794538	0.743946	0.798889	0.697738	0.71902	M
m	0.746769	0.630898	0.793958	0.651115	0.686343	0.718815	0.620438	0.669403	M
N	0.712904	0.500567	0.885593	0.603309	0.630587	0.814964	0.474302	0.631791	M
n	0.798105	0.609762	0.803374	0.687693	0.656996	0.736105	0.666213	0.666213	M
O	0.776562	0.789915	0.907808	0.871793	0.912175	0.86988	0.78348	0.791395	M
o	0.80607	0.808318	0.912621	0.911942	0.911469	0.907414	0.807997	0.807245	M
P	0.659638	0.633177	0.776194	0.911998	0.877618	0.268708	0.898651	0.224782	M
p	0.70406	0.668322	0.792126	0.896435	0.877152	0.50092	0.888426	0.425565	M
Q	0.775003	0.753586	0.898752	0.865573	0.736179	0.868098	0.639132	0.760924	M
q	0.669027	0.703278	0.873433	0.500738	0.778012	0.899237	0.424649	0.899955	M
R	0.67947	0.596637	0.86382	0.888428	0.902855	0.823649	0.816711	0.636111	M
r	0.884299	0.380975	0.857533	0.849064	0.5	0.228374	1	0.198739	M
S	0.791233	0.585638	0.886131	0.605198	0.67145	0.922664	0.516205	0.768736	M
s	0.803559	0.615493	0.840276	0.605376	0.676828	0.883162	0.571128	0.792316	M
T	0.876295	0.876295	0.826392	0.422403	0.823825	0.424495	0.49599	0.5	M
t	1	0.273444	0.948441	0.726954	0.280409	0.583042	0.611897	0.453898	M
U	0.623098	0.630292	0.742494	0.641089	0.749141	0.645265	0.712147	0.718628	M
u	0.60397	0.625592	0.638378	0.698817	0.641122	0.861849	0.618174	0.786306	M
V	0.732175	0.725062	0.564743	0.40309	0.560398	0.413813	0.487221	0.499164	M
v	0.841058	0.795541	0.626671	0.41965	0.609193	0.434329	0.49238	0.495854	M
W	0.737041	0.721779	0.624716	0.505947	0.621462	0.505577	0.81827	0.804932	M
w	0.700861	0.694802	0.626914	0.489004	0.621714	0.487245	0.759181	0.756183	M
X	0.906822	0.917401	0.826589	0.834647	0.850335	0.851468	0.949583	0.952655	M
x	0.871998	0.85819	0.824554	0.837351	0.826824	0.8439	0.928196	0.918115	M
Y	0.955173	0.967834	0.566867	0.352011	0.578607	0.327534	0.499806	0.461656	M
y	0.946767	0.921609	0.605445	0.510602	0.609624	0.229408	0.545948	0.237134	M
Z	0.561882	0.896267	0.406952	0.673263	0.592415	0.371681	0.903144	0.546325	M
z	0.641824	0.922785	0.484234	0.705987	0.639609	0.438156	0.922205	0.622994	M

Figure 6.9: English alphabets representation by means of Markov value

char	p12	p21	p13	p31	p34	p43	p24	p42	type
0	0.734912	0.748061	0.932858	0.910301	0.933263	0.908273	0.728854	0.73992	M
1	0.32761	0.594509	0.5	0.199452	0.819905	0.987765	0.182464	1	M
2	0.51878	0.841253	0.434241	0.740227	0.660791	0.614145	0.809186	0.715426	M
3	0.527181	0.940904	0.583558	0.597133	0.929082	0.923802	0.528641	0.916819	M
4	0.313167	0.671235	0.288317	0.411248	0.671167	0.908559	0.466359	0.948654	M
5	0.937081	0.511943	0.953117	0.538639	0.472879	0.607789	0.59444	0.73859	M
6	0.888991	0.708554	0.901975	0.757787	0.70224	0.719044	0.707564	0.687319	M
7	0.489607	0.828087	0.291143	0.22891	0.7619	0.237388	0.142069	0.09522	M
8	0.823659	0.831975	0.89253	0.878817	0.878625	0.880153	0.765586	0.786747	M
9	0.685844	0.713828	0.72306	0.707353	0.769161	0.912363	0.701201	0.884911	M

Figure 6.10: Numerical characters representation by means of Markov value

char	p12	p21	p13	p31	p34	p43	p24	p42	type
◌ฺ	0.826994	0.837614	0.860119	0.309256	0.806306	0.945589	0.302703	1	B
◌ฺ	0.861739	0.556742	0.83678	0.636517	0.603908	0.97413	0.598652	0.820162	B
ฐb	0.174285	0.954207	0.130733	0.944089	0.766146	0.773001	0.923463	0.706385	B
ฯ	0.707781	0.659209	1	0	0.961672	0.488064	0	1	M
ฯ	0.648597	0.521728	1	0	0.928362	0.622754	0	1	M
เ	1	0	0.799081	0.932634	0	1	0.916748	0.560815	M
ไ	0.434794	0.502139	0.976601	0.116979	0.597223	0.835228	0.037081	0.5	M
ใ	0.466283	0.393049	1	0	0.600161	0.959039	0	1	M
ใ	0.438456	0.514414	1	0	0.63497	0.866423	0	1	M
า	0.447941	0.949418	1	0	0.873386	0.7	0	1	M
◌ั	0.715254	0.670599	0.672627	0.928494	0.40525	0.83372	0.620813	0.867473	T
◌่	0.918033	1	1	0.998512	1	1	0.916667	1	T
◌้	0.926721	0.269231	0.674328	0.814397	0.254478	0.58664	0.946084	0.52464	T
◌๊	0.751759	0.634194	0.611635	0.820774	0.597818	0.745947	0.896298	0.703077	T
◌๋	0.676539	0.789091	0.681781	0.765455	0.743444	0.817615	0.719348	0.821862	T
◌์	0.532644	0.629071	0.40235	0.90182	0.549648	0.347932	0.927611	0.309402	T
◌ๆ	0.850997	0.886944	0.835626	0.809578	0.856274	0.838643	0.824408	0.868617	T
◌ั	0.83663	0.858174	0.641058	0.903433	0.658837	0.896158	0.895694	0.886766	T
◌ั	0.432164	0.823158	0.376314	0.847895	0.670338	0.815557	0.81717	0.840458	T
◌ั	0.242206	0.95898	0.219204	0.781227	0.759658	0.704779	0.841549	0.867379	T
◌ั	0.330705	0.816842	0.376314	0.847895	0.755508	0.708822	0.81717	0.840458	T
◌ั	0.891485	0.413492	0.820832	0.944775	0.395914	0.613414	0.976897	0.60993	T

Figure 6.11: other Thai alphabets representation by means of Markov value

6.5 Result

For testing our application, we use three types of pictures to be applied with our experiment as the test sest including printed pictures, scanned pictures, and pictures from the mobile camera. All of these pictures will be used as input images of our application.

The result of each type of pictures will be measured as how our application show the corrected character of the information that contains in the national ID card. The accuracy of each type of pictures depend on the resolution of the picture, amount of noises in the picture, and some exceptional cases that contain in the information.

6.5.1 Printed Picture

Printed picture is the picture that we create by ourselves for comparing the position of each ROI with the real national ID card, and creates it using Photoshop. This mean that the position of the information will be the same as where the information is on the real card. The advantages of this type of picture are the noise on national ID card is lowered, the resolution is the highest among other types of pictures, and be able to test any cases we want without requiring the real card.

6.5.2 Scanned Picture

Scanned picture is the picture of the national ID card that uses a scanner to take in the national ID card picture. This type of picture will have a higher resolution than the picture from the mobile camera, fewer noises on picture and no light reflection, but sometimes watermark can appear which make our classification wrong.

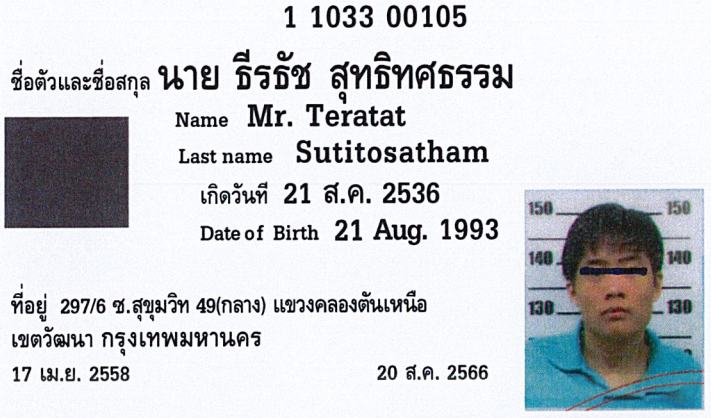


Figure 6.12: Example of printed picture

Figure 6.13: Example of scanned picture

6.5.3 Picture from Mobile Camera

This type of picture is taken from the mobile phone camera. This type of picture should be our real input of the application, but there are a lot of problems that occur in this type including watermarks, light reflections, low resolutions, and some user errors which make the accuracy low.

The order of the accuracy from the highest to lowest will be printed picture , scanned picture and picture from the mobile camera respectively.

ROI \ Type of input	Printed Picture	Scanned Picture	Picture from Mobile Camera	Average for each ROI
Title	100.00%	100.00%	100.00%	100.00%
ID number	100.00%	100.00%	58.97%	86.32%
First name (TH)	75.00%	66.68%	31.58%	57.75%
Last name (TH)	77.41%	68.18%	18.51%	54.70%
First name (TH)	92.00%	81.25%	51.85%	75.03%
Last name (TH)	78.94%	79.66%	66.67%	75.09%
Date of Birth	100.00%	85.71%	9.52%	65.08%
Date of Birth	95.24%	92.86%	90.47%	92.86%
Address	70.44%	46.02%	9.27%	41.91%
Date of Issue	100.00%	50.00%	28.57%	59.52%
Date of Expiry	100.00%	85.71%	23.81%	69.84%
Average for each Input Type	89.91%	77.82%	44.47%	70.74%

Table 6.1: The accuracy of each ROI for each type of picture

Chapter 7

Conclusion

7.1 Conclusion

We have applied the histogram-based algorithm to separate the character from the information and apply Markov chain as the classification process of the character. The Markov chain works on classification process, but a histogram-based algorithm have some problems with the separation process due to the over-lapping characters. In our project, the major problem occurs in the separation process which mean that we need to improve the histogram-based algorithm or use a new algorithm to separate.

7.2 Limitations and Suggestions

Most of our problems occur on an image acquisition process which makes our input image not good enough to extract the information in the national ID card. These problems lower our application accuracy and performance which some problems we can solve only just for making our application be able to execute. The suggestions for the limitations to our application are as follow:

- **Low Resolution:** The resolution of the mobile camera is varied according to the API level of the Android operating system. Therefore, to develop further, we recommend to use the Android native camera instead of using Camera build-in class since it will be deprecated in a more upgraded level of API.

- **Watermarks:** The positions of the watermarks in a Thai national ID Card are random and are varied for each person. Therefore, there is no certain ways to eliminate all watermarks perfectly, however, if the watermarks are certainly positioned. It could be solved by using Morphological filtering, thresholding, and the difference between images.

- **Light Reflection:** Might enable some filters from camera that reduce the brightness of the image, but we have search for an algorithm and we found histogram equalization which could be able to solve this problem.

- **Skewing Photo of ID Card:** There are papers regarding on unskewing the skewness by calculating the angle of the skewness and turn it to back to normal.

7.2.1 Watermarks

Watermark is an identifying image which will appear when the light reflect with the national ID card. The reason why the watermark becomes a problem is that when we use the taken photo by the mobile phone camera and use it as an input image, the reflection will make the watermark appears randomly which will block the information that we want to get.

7.2.2 Thai Fonts

Thai font is one of our main problem. The font that uses in Thai national ID card is DilleniaUPC which has two versions which are DilleniaUPC 2.0 and

Figure 7.1: Watermark and light reflection on national ID card

DilleniaUPC 5.0. The difference between these two versions is the typing pattern of the word which is the combination of the top-layer character and middle-layer character. This cause the problem when we create a trained set of an exceptional character for classifying what character is. DilleniaUPC 5.0 is used in the operating system above WindowXP, on the other hand, the printed national ID card is still in a version of DilleniaUPC 2.0 which will make some cases different from how we can type in the trained set.

So, we fix this problem by creating the trained set of the exceptional cases in Photoshop to make our trained set be the most similar to what will be appeared in the national ID card.

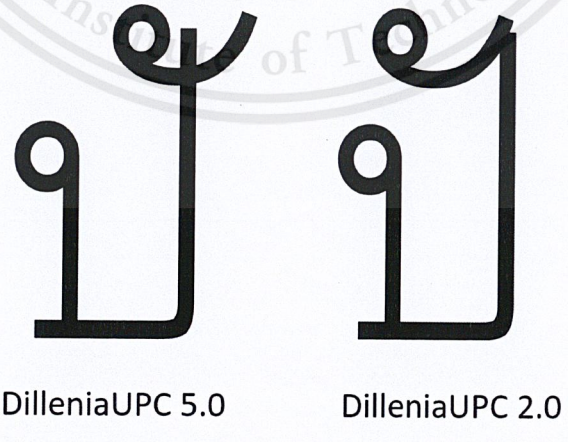


Figure 7.2: Difference between DilleniaUPC2.0 and DilleniaUPC5.0

7.2.3 Distorted Character

Some national ID cards are not perfect. Some areas in the ID card have scratches, blurs or disappeared characters. This damaged area will make some information in the input image be distorted which make the result of the classification process wrong.

Figure 7.3: Distorted information on national ID card

7.2.4 Light Reflection

This problem is similar to the watermark problem. When we take the picture, if the national ID card is opposite with the light source, there will be a light reflection that will block the information that we need to extract.

7.2.5 Too Small Character

For this problem, after we separate characters from words, some ROIs especially address contain very small characters. When we get ROIs which contain too small character, there will be problems with the separation process and classification process. In separation process, if the character is too small, the separation could be wrong because the two characters are too near to each other and

there are no space between them which will make our algorithm interprets that as one character. Follow with the classification process, our application can't classify what character is due to no trained set in our database.

References

[1] Supachai Yangwongsan and Buntida Suvacharakulton, Realization of a High Performance Bilingual OCR System for Thai-English Printed Documents. *Natural Language Processing and Knowledge Engineering (NLP-KE)*, pp.14-16, 2010.

[2] C. Tanprasert and S. Sae-Tang, Thai type style recognition. *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE*, pp.336 - 339, 1999.

[3] Sarin Watcharabutsarakham, Spell Checker for Thai Document. *TENCON 2005 - 2005 IEEE Region 10 Conference*, pp.1-4, 2005.

[4] Supachai Tangwongsan and Buntida Suvacharakulton, OCR with Word Prediction Technique for Bilingual Documents. *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International* , pp.343 - 347, 2012.

[5] Xi-Ping Luo, Li-Xin Zhen; Gang Peng, Jun Li and Bai-Hua Xiao, Camera based mixed-lingual card reader for mobile device. *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pp.665 - 669, 2005.

[6] Teera Siriteerakul, Mixed Thai-English Character Classification Based on Histogram of Oriented Gradient Feature *2013 12th International Conference on Document Analysis and Recognition*, pp.847 - 851, 2013.

[7] http://blogs.mathworks.com/steve/2012/12/04/image-effects-part-4/?s_tid=Blog_Steve_Archive

[8] https://www.slideshare.net/gichelleamon/6-spatial-filtering-p2

[9] https://en.wikipedia.org/wiki/Otsu%27s_method

[10] http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[11] Histograms and Bayes Classifiers by Ral Rojas, Freie Universit at Berlin

Appendix A

User's Guide

In the main Screen, if the user touches the add icon, the system will let the user selects an image from gallery. If the user touches the camera icon, the system will go to the mobile camera. If the user touches the arrow icon, the system will clear all the saved id card information. Users also can touch on the name to access that person that has been stored in the id card information database.

After the user touches the add icon, the system will prompt the user to select an image from photo gallery. After that the user can select the input image they want. After the user has selected the input image, if the the input is acceptable, the system will process and interpret the information, otherwise, the system will prompt the error messages.

After the user touches the camera icon, the system request the phone's camera and allow the user to take a photo of an ID Card. After the user has taken the photo, the system will go to crop image screen.

After the user has cropped the image, the system will show the preview of the input image. if the input is acceptable, the system will process and interpret the information, otherwise, the system will prompt the error messages.

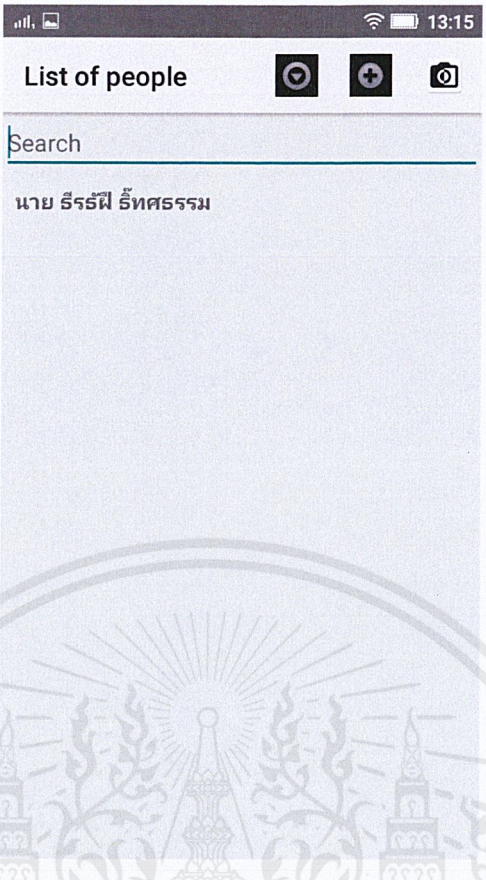


Figure A.1: Home Screen

In the interpreted screen, the user can edit the misinterpreted information by touching on it. After finishing, the user will touch on the confirm button. The system will return to the home screen with a newly obtained ID Card information of that person.

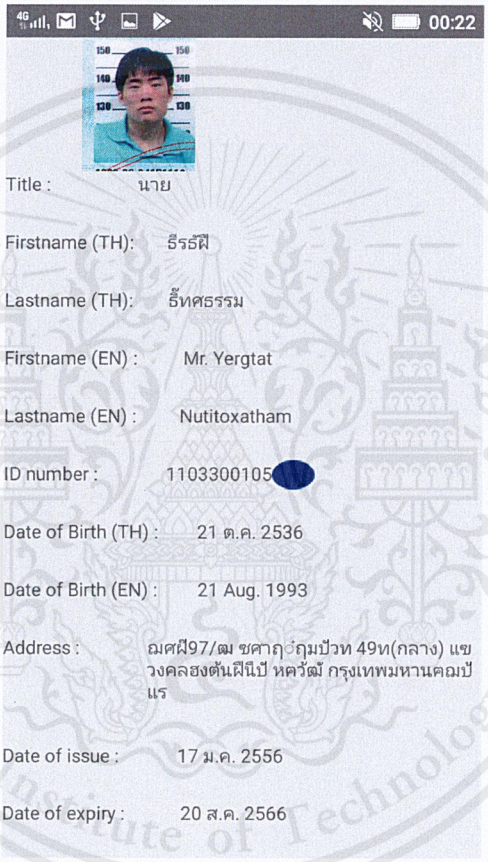


Figure A.2: View The Stored ID Card Information

Figure A.3: Select an Image from The Gallery

Figure A.4: Crop Image Screen

Figure A.5: Preview of The Input Image

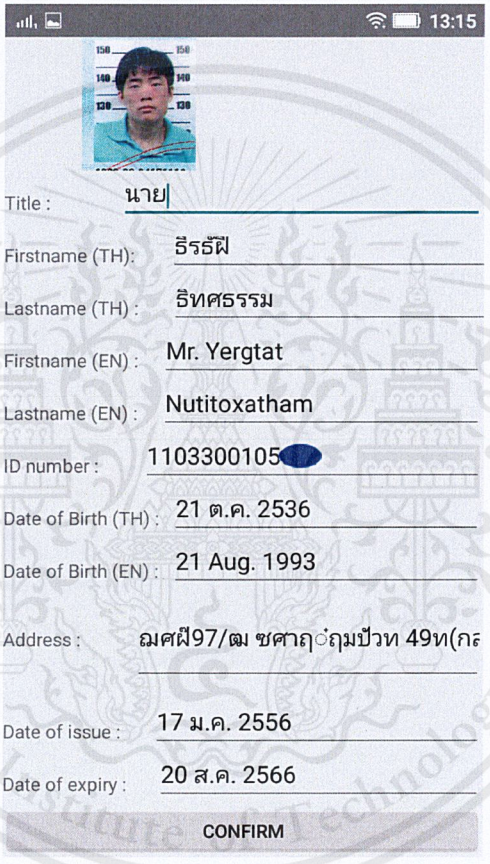


Figure A.6: Interpreted Screen

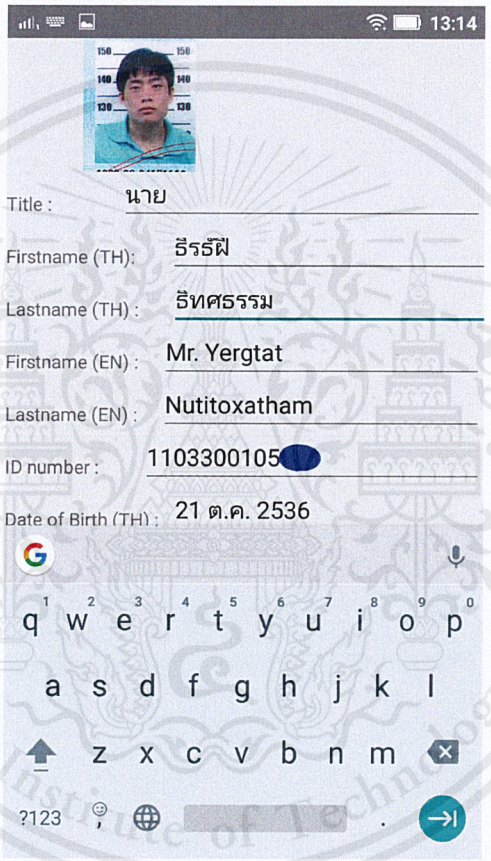


Figure A.7: Edit misinterpreted Information

Abbreviations

ID	Identification
OCR	Optical Character Recognition
ROI	Region of Interest
SAD	Sum of Absolute Difference