A Novel Approach of Finger Print Recognition Using (Ridge) Minutia Method and Multilayer Neural Network Classifier

Devendra Singh Kaushal¹, Yunus Khan² & Dr. Sunita Varma³

¹Department of CSE
Jawaharlal Institute of Technology Borawan Khargone M.P. India
devendrasinghkaushal@gmail.com

²Department of CSE
Jawaharlal Institute of Technology Borawan Khargone M.P. India
callyunuskhan@gmail.com

³Department of CTA
Shree G.S. Institute of Technology and Science Indore M.P. India
Sunita.varma19@gmail.com

Abstract: Finger print recognition has always been a challenging field for the researchers. Fingerprint recognition is a biometric identification technology that distinguishes whether two fingerprints are the same fingerprint. There has been an astounding progress in the development of the systems for fingerprint recognition. Finger print recognition is the process of identification and classification of fingerprint using some classification and feature extraction method. The finger print recognition process can have several stages like preprocessing, training, testing, recognition and post processing. The recognition generally consists of feature extraction and classification. The choice of feature extraction and classification scheme affects the performance of recognition largely. In this paper, we implement finger print recognition scheme with multilayer neural network and minutia method for the feature extraction. This approach we extracted 80 features of every image using ridge scheme of minutia method of feature extraction and multilayer neural network for the classification and recognition of fingerprint. In this paper we describe how to implement each functional which we explained in functional specification. This design is in high level software design, it includes architectural design, data design interface design and procedural design.

Keywords: Multi-Layer neural network, Feature extraction, Ridge.

1. Introduction

Fingerprint is one of the popular ways in human being identification. Fingerprint recognition is a biometric identification technology that distinguishes whether two fingerprints are the same fingerprint. The theory of fingerprint recognition is finding out minutiae (bifurcation and ridge) of two fingerprints, comparing them depend on their direction, local position and type. This project is Fingerprint recognition system. It is software that will work the same fingerprint recognition theory in computer. This system will compare two fingerprints image to make sure whether they came from the same finger. Fingerprint recognition system will implement fingerprint recognition theory base on image pre-processing technique and image recognition technique.

This document has been written to indicate the software design of the fingerprint recognition system. It designs how to implement each functional which I explained in functional specification. This design is in high level software design, it includes architectural design, data design interface design and procedural design. At the beginning of this document, I will do the low level design.
1. Low level design

The low level design explains what the project will do and what the project looks like. The contents of this part show the domain layer of the project. It includes use case diagram, brief use case and system sequence diagram and domain model.

1.1. Use case diagram

![Figure 1.1 Use Case Diagram](image)

1.2. Brief use case

**Use case: load image**

Actor: user

Description: This use case begins with a user wants to load a fingerprint image in system. User can load the fingerprint image from an external image file or capture a fingerprint image from a fingerprint scanner.

**Use case: Image pre-processing**

Actor: user

Description: This use case begins with a user wants to pre-process a fingerprint image. This part includes all operation of image pre-processing: normalization, orientation estimation, edge detection, ridge detection, and thinning and minutiae extraction. After loaded image, user has to click these buttons one by one to finish the pre-processing.

**Use case: Database management**

Actor: user

Description: This use case begins with a user wants to enroll a fingerprint/match a fingerprint to database/update fingerprint information/delete a fingerprint. For enroll, system will add fingerprint information to database. For match in dataset, system will select a fingerprint in database for user. For update, system will select fingerprint information which user preference, and then allow user to change its information, but user cannot change its minutiae information. In delete fingerprint information, system will select fingerprint information which user preference, and then allow user to delete it.

1.3. System sequence diagram

![Figure 1.2 System sequence diagram for image pre-processing use case](image)
Figure 1.4 System sequence diagram for Database management use case
1.3.1. Domain model

2. High level design
This part shows the idea of how I design to implement this project.

2.1. Architectural Design
The main idea in this part is that comprising the system with a number of components, and the components are divided into six: User Interface module, Image loading module, Image pre-processing module, Image recognition module database management module and database module. Each module can be comprised into several sub-modules. In this part, I will describe them in detail. The main frame work of entire system architectural is showed in figure 2.1.

Figure 1.5 Domain model diagram

Figure 2.1 Main Frame of System architectural
Figure 2.1 shows the relationship of each module. User can do the operation in each module directly. The operation result in image recognition module, image pre-processing module and database management module will return back and display to user. Data has
been transformed between user and database in database management module. After image pre-processing, user can do the image recognition or enroll fingerprint information in to database through database management module. In Image recognition module, system gets the image data from image pre-processing module and gets the template image data from database. This is the main relationship between each module. Each module will be described in the following parts.

2.1.1 User Interface module
User interface module contains the main operation of this system. Through user interface, data can access Image loading module, Image pre-processing module, Image recognition module and database management module. These four modules’ result will display to user in user interface module.

2.1.2 Image loading module
User can import the external image data in this module. In this section, it allows user to load the image from external image file and also allows user to capture the image file from fingerprints scanner. The loading image is bmp type image.

2.1.3 Image pre-processing module
In this module, user can do pre-processing for an image. The goal of this module is to make the input fingerprint image suitable for recognition. In this module system get the original image data from image loading module. This module can be divided into six sub-modules: Normalization, Orientation estimation, Edge detection, Ridge detection, thinning and Minutiae extraction. Each sub-module instead each steps in image pre-processing. Figure 2.2 shows the relationship between each sub-module. System will run these sub-modules one by one. After this module, an image is ready for recognition. Description of each sub-module is showed following:

- **Normalization**: In this sub-module, fingerprint image will become a gray image. The result is a gray image with an excepted average gray level and an excepted variance.
- **Orientation estimation**: This sub-module is calculating the orientation file of a fingerprint image.
- **Edge detection**: The goal of this sub-module is to keep the useful data, and throw the noisy point of an image.
- **Ridge detection**: The goal of this sub-module is to separate the background and foreground. The data of this module came from edge detection module.
- **Thinning**: In this module, data came from ridge detection module. All ridges will be thinned into a 1 pixel weight line.
- **Minutiae extraction**: This is the last sub-module in Image pre-processing module. In this module, system extracts minutiae from a thinned image. After minutiae extraction, system will check the minutiae and remove the false minutiae to keep the matching algorithm accurate. In order to show these two processes, I will show two results in user interface, one is minutiae extraction and another one is after false minutiae extraction.

**Figure 2.2 Sub-modules in Image pre-processing module**

**Image recognition module**
The goal of this module is matching two fingerprints. The data of this module came from Image pre-processing module. It also gets the template data from database. This module also can be divided into two sub-modules: Core extraction and Minutiae matching. Core extraction module is the first sub-module in Image recognition module, in this sub-module, system will find out fingerprint’s core and delta; it is for finding out the local position of the minutiae. In Core extraction
module, each fingerprint has been rotated into a format position (fingerprint is perpendicular in the image). The last sub-module is minutiae matching module. This sub-module implements the recognition function. The relationship between each sub-module in image recognition module shows in figure 2.3.

2.1.6 Database module
This module is physical database, it records the minutiae and fingerprint information.

Data Design
Data structure design
Array is the basic data structure in this system. Almost data is storage into an array. In image pre-processing, the main data is image data; there are two structures to hold an image data in image pre-processing: Pixel and Image. Pixel is for explain the image, and the image structure is to hold the data in image pre-processing. These two kinds of structures design as table 3.1 and table 2.2.

<table>
<thead>
<tr>
<th>Pixel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Data type</td>
</tr>
<tr>
<td>X coordinate</td>
<td>Integer</td>
</tr>
<tr>
<td>Y coordinate</td>
<td>Integer</td>
</tr>
<tr>
<td>Gray value</td>
<td>Integer</td>
</tr>
<tr>
<td>X gradient</td>
<td>Float</td>
</tr>
<tr>
<td>Y gradient</td>
<td>Float</td>
</tr>
<tr>
<td>Orientation</td>
<td>Float</td>
</tr>
</tbody>
</table>

Table 2.1 Pixel data structure

<table>
<thead>
<tr>
<th>Image</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Data type</td>
</tr>
<tr>
<td>Height</td>
<td>Integer</td>
</tr>
<tr>
<td>Weight</td>
<td>Integer</td>
</tr>
<tr>
<td>Pixel</td>
<td>Pixel</td>
</tr>
</tbody>
</table>

Figure 2.3 Sub-modules in Image recognition module

2.1.5 Database management module
In this module, the main goal is allowing user to do the operation in database. For example enroll fingerprints information into database. Note that the image information data include minutiae information. All data in database is after image pre-processing. User cannot enroll a fingerprint which is un-processing.
Table 2.2 Image data structure

In image recognition, the main data is to explain for fingerprint data information and minutia is for fingerprint information. There are two data structures in image recognition: Fingerprint and minutia. Fingerprint show in table 2.3 and table 2.4.

<table>
<thead>
<tr>
<th>Minutia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Minutia id</td>
</tr>
<tr>
<td>Minutia type</td>
</tr>
<tr>
<td>X coordinate</td>
</tr>
<tr>
<td>Y coordinate</td>
</tr>
<tr>
<td>Angle</td>
</tr>
</tbody>
</table>

Table 2.3 Minutia data structure

<table>
<thead>
<tr>
<th>Fingerprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Fingerprint image</td>
</tr>
<tr>
<td>Core position x</td>
</tr>
<tr>
<td>Core position y</td>
</tr>
<tr>
<td>Delta</td>
</tr>
<tr>
<td>Number of minutiae</td>
</tr>
<tr>
<td>Minutiae</td>
</tr>
</tbody>
</table>

Table 2.4 Fingerprint data structure

2.2.2 Database design

Each image has been recorded with an id. In database, in this application, data will be storage into database, the main table is design for record fingerprint and minutiae data. The main data is image data, fingerprint data and minutiae data. Each tables design as table 2.5 and table 2.6.

twice, one is original one and one is pre-processed one.

<table>
<thead>
<tr>
<th>TABLE_FINGERPRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field name</td>
</tr>
<tr>
<td>FINGERPRINT_ID</td>
</tr>
</tbody>
</table>
A NOVEL APPROACH OF FINGER PRINT RECOGNITION USING (RIDGE) MINUTIA METHOD AND MULTILAYER NEURAL NETWORK CLASSIFIER

Devendra Singh Kaushal, Yunus Khan & Dr. Sunita Varma

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIG_IMAGE_ID</td>
<td>Integer</td>
<td>An id for find out the original image</td>
</tr>
<tr>
<td>PROCESSED_IMAGE_ID</td>
<td>Integer</td>
<td>An id for find out the processed image</td>
</tr>
<tr>
<td>DELTA</td>
<td>Double</td>
<td>A delta angle of a fingerprint image</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>String</td>
<td>A description of a fingerprint</td>
</tr>
</tbody>
</table>

Table 2.5 fingerprint table in database

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINUTIA_ID</td>
<td>Integer</td>
<td>An id for identify a minutia. Primary key</td>
</tr>
<tr>
<td>FINGERPRINT_ID</td>
<td>Integer</td>
<td>An id for shows fingerprint which this minutia belongs</td>
</tr>
<tr>
<td>POSITION_X</td>
<td>Integer</td>
<td>X coordinate in local position</td>
</tr>
<tr>
<td>POSITION_Y</td>
<td>Integer</td>
<td>Y coordinate in local position</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>Double</td>
<td>A angle of a minutia</td>
</tr>
</tbody>
</table>

Table 2.6 minutia table in database

3.3. Interface Design
The goal of GUI design is easy to use for user. Figure 2.4 is the main match user interface.
In my application’s main frame, User click the capture/load button can load the image by two different ways. Before loading the image, user need to choose which image (input image or template image) should be loaded. After image was loaded, user can do the image pre-processing next. Each step in pre-processing allocation a button. User can click the button one by one and see the result of each step. Input image and template image will do the pre-processing at the same time, user does not need to click every button twice. After pre-processing, user can do match through clicking the match button. The result will display in Match result panel. User can go to the database management through clicking the system menu in top of this window. Figure 3.5 is the enroll window.

In this window, user can enroll a fingerprint or match a fingerprint in database. User does not need to do pre-processing step by step (Note that before enroll/match, image must do image pre-processing). User can choose “Auto processing” to let system do pre-processing automatically. User can add a description of fingerprint before enroll it. After user click “Match in Database” button, if there is any same fingerprint in database, system will pop a result window (Figure 2.6).
Figure 2.6 is not only to show the match result, but also can do database management in this window. In update, user only is allowed update description. When user clicks the cancel button, system will go back to the main user interface (Figure 2.4).

3.4. Procedural Design
This part is for representing procedural detail that facilitates translation to code.

2.4.1 Class design
I mention develop fingerprint recognition system with Java. Almost module/data structure will be implementing as a class. Figure 2.7 is the system class diagram. This diagram shows the relationship between each class. In this diagram, some elements were not showed (e.g. GUI components).
Figure 2.7 Main class design diagram

This is the main class design diagram, some class maybe not consider in it, some class elements probably will be changed in coding.

2.4.2 Programming design

The idea of this design document is to separate this system into several modules, and analysis how to implement each module. This part is to design procedural of each module, in order to show the flow control of each module. Note that the pseudo code of this part only shows the logical organized of each module. Pseudo code will not show the algorithm in detail.

1. Main application Pseudo code

Begin

Input fingerprint
Fingerprint pre-processing
If enroll
    Add fingerprint to database
If match in database
    Select fingerprint in database.
If match with another fingerprint
    Input fingerprint_2
    Fingerprint_2 pre-processing
    Match fingerprint with fingerprint_2
End

2. Image pre-processing module Pseudo code

Begin

Input image
Image normalization
Image orientation estimation

End
Calculate b’s orientation base on sum of all pixel’s gradient in x axis and sum of all pixel’s gradient in x axis.

Return orientation filed

**Pseudo code of image enhancement**

Begin

Input normalized image

Input orientation file

F = ridge frequency // base on orientation file

A = ridge angle direction// base on orientation file

Calculate the new pixel gray value in normalized image

Set the new gray value in normalized image

Return enhanced image

End

**Pseudo code of ridge detection**

Begin

Input enhanced image

Calculate a gray level threshold

I = get pixel gray value in image

If I > gray level threshold

Set pixel gray value is 1

Else

Set pixel gray value is 0

Return image

End

**Pseudo code of thinning**

Begin

Input pre-processing image

Thinning image using Zhang-Suen Thinning algorithm

Return thinned image

End

**Pseudo code of minutiae extraction**

Begin

Count = 0

Calculate the value change of each pixel p and its 8 neighbors

Begin

If 1 change to 0

Count ++

Else

If 0 change to 1

Count ++

End

End

**3. Image recognition module Pseudo code**

Begin

Input pre-processing image_1

Input pre-processing image_2

Find core and delta in image_1

Find core and delta in image_2

End
2. Conclusion

In this paper, we implement finger print recognition scheme with multilayer neural network and minutia method for the feature extraction. This approach we extracted 80 features of every image using ridge scheme of minutia method of feature extraction and multilayer neural network for the classification and recognition of finger print. In this paper we describe how to implement each functional which we explained in functional specification. This design is in high level software design, it includes architectural design, data design interface design and procedural design.

References:

A NOVEL APPROACH OF FINGER PRINT RECOGNITION USING (RIDGE) MINUTIA METHOD AND MULTILAYER NEURAL NETWORK CLASSIFIER

Devendra Singh Kaushal, Yunus Khan & Dr. Sunita Varma