

# An Online Integrated Fingerprint Image System

Ching-Yu Huang, Limin Liu, and Yui-Liang Chen

**Abstract**—Fingerprint analysis is one of the most commonly-used biometrics technologies, primarily used in forensic science to identify people. Due to various reasons, most fingerprint related studies are conducted in an isolated environment which makes it hard for researchers to collaborate across institutes. Web-based client/server applications have become an important component for international collaboration because such an application allows users from different locations to execute the same program on the same data. Such systems can be quickly developed and are easy to use. In this paper, a 3-tier integrated online fingerprint system was developed to manage and process fingerprint images in real-time. This system allows users all over the world to access the public NIST-14 fingerprint dataset and collaborate with other researchers to conduct the research task. The front-end of the system is an advanced web-based interface that allows users to view fingerprint images in different resolution, singular points, enter, and update the location of singular points. This system can also preserve the entered information in the back-end database.

**Index Terms**—Classification, database, fingerprint, online image processing.

## I. INTRODUCTION

A fingerprint consists of two special direction-oriented parts: ridges and valleys. On latent or digitized fingerprint images, ridge is usually represented in dark and valley in light. These direction-oriented patterns contain various fingerprint features, including global features called singular points and local features called minutiae [1]. A fingerprint databank could contain millions or billions of fingerprints, and the size is still growing at a steady rate as the population increases. Nowadays, fingerprint images are processed and recognized by computer programs (or AFIS, Automated Fingerprint Identification System) but some features, e.g., the location of singular points, still require verification from human experts since the locations of singular points derived from computational definitions are different. This research will focus on developing a framework to utilize the latest artificial intelligence technology on identifying features and partial fingerprint classification.

In order to allow users to have the ability to interact with the images displayed on the browser, Graphic User Interface (GUI) tools must be developed that can not only catch click events but also take inputs, such as drawings by a mouse or

keyboard. The client side application uses the GUI tool to display the results provided by algorithms stored at the server side in real time. However, to process a large number of images consumes a lot of computer resources including CPU time, memory, and storage space. It is a challenging task to integrate the front-end and back-end efficiently on a multiuser environment to provide a real-time working environment.

TensorFlow [2] developed by Google is an open-source software library. It can be integrated within applications to classify images based on deep learning neural networks. Shreyn [3] used Convolutional Neural Networks (CNNs) model for the fingerprint image classification and showed that effective image preprocessing can greatly reduce the dimensionality of the problem. Jeon and Rhee [4] also proposed the use of a CNN model combined with an ensemble model and a batch normalization technique after improving the quality of fingerprint images. Perrier [5] reported that Google announced the release of Tensorflow.js, a Javascript implementation of Tensorflow. With Tensorflow.js, one can train deep neural networks in browser at client-site. In order to interact with the dataset and the database, the TensorFlow should be still run at the application server in our proposal. Toth *et al.* [6], presented a platform that can help researchers to develop functional components for image processing algorithms for specific tasks by providing training data sets. Senthilkumar *et al.* [7] proposed a new system that can be achieved by adopting web service based image processing system. ReddyBorraa *et al.* [8] proposed technique uses Wave atom transform for denoising, morphological operation for image upgradation and Adaptive genetic neural system for image classification.

The goal of this project is to study the mechanisms for developing an online system with advanced web based tools at the front-end that can perform functions to visualize the images, transform (translation, rotation, and zoom), adjust colors, apply filters on images, apply pattern recognition, analyze and interact with the images, in real-time. This system must also be able to extract features and differentiate normal and abnormal areas of the image, while at the same time allowing users to retrieve, view, highlight, and mark the images. In addition to altering the images, users will be able to view information and write notes during the works. These advanced online tools and algorithms provide a centralized application for people who are working at different locations and working in different fields to collaborate with other coworkers.

The rest of the paper is organized as follows. Section II shows the NIST-14 fingerprint dataset. Section III describes the architecture of the proposed system. Section IV lists some of the features implemented in the system and followed by the conclusions in Section V.

Manuscript received May 1, 2018; revised August 20, 2018.

Ching-Yu Huang is with the Kean University, Union, NJ 07083, USA (e-mail: chuang@kean.edu).

Limin Liu and Yui-Liang Chen are with Shih Hsin University, Taipei, Taiwan (e-mail: liminliu@mail.shu.edu.tw, ychen@cc.shu.edu.tw).

## II. NIST-14 DATASET INFORMATION

In this paper, the NIST Special Database 14 (NIST-14) fingerprint dataset is used to design the system [9]. The dataset collects 2700 grayscale fingerprint images named from s0024301 to s0027000 from 270 individuals. All 10 fingers' images of these people are collected in the dataset.

### A. Singular Points

Each fingerprint has two important features – singular points and minutiae. Based on the global pattern, fingerprints are classified into 19 types as shown in Fig. 1. Singular points of fingerprints are classified into two types – core and delta as shown in Fig. 2 (a) and (b), respectively, and the centers of core and delta are marked in red rectangles. They are considered the global pattern because their geometric locations and numbers determine the fingerprint types.

Arch (aa)
Tented Arch (tt)
Ulnar Loop Ridge Counts (01 - 49)
Radial Loop Ridge Counts (51 - 99)
Plain Whorl Inner Ridge Tracing (pi)
Plain Whorl Outer Ridge Tracing (po)
Plain Whorl Meeting Ridge Tracing (pm)
Central Pocket Whorl Inner Ridge Tracing (ci)
Central Pocket Whorl Outer Ridge Tracing (co)
Central Pocket Whorl Meeting Ridge Tracing (cm)
Double Loop Whorl Inner Ridge Tracing (di)
Double Loop Whorl Outer Ridge Tracing (do)
Double Loop Whorl Meeting Ridge Tracing (dm)
Accidental Whorl Inner Ridge Tracing (xi)
Accidental Whorl Outer Ridge Tracing (xo)
Accidental Whorl Meeting Ridge Tracing (xm)
Approximate Class (ac) followed by a valid class
Amputation or Missing (xx)
Scar or Mutilation (sr)

Fig. 1. The fingerprints classification name and class code indicated inside the parentheses.



Fig. 2. The singular points. (a) core and (b) delta positions are highlighted in red.

It is very hard to precisely detect a singular point at pixel level. This is because there are many definitions exist and there do have difference among these definitions. Therefore, most of the singular point detection algorithms use the block direction method and represent a singular point in a block [10]. Fig. 3 shows the Henry's definition of the delta points presented over a hundred year ago [11].



Fig. 3. Henry's definition of 'the delta'.

Fig. 4 shows the FBI's definition [12]. Note that, we list only some of the definition of the Henry and FBI's full definition of singular point. However, one still can find

singular patterns that are not listed in those documentations.

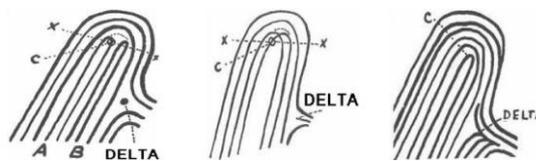


Fig. 4. FBI's definition of 'the delta'.

### B. Minutiae

Minutiae is considered as the local pattern. It has been classified into the following sub-types: bifurcation, ridge ending, island, and pore as shown in Fig. 5. They are formed when the ridge split, isolated, or emerged. These local features are easily affected by the quality of the image so they are not very reliable.

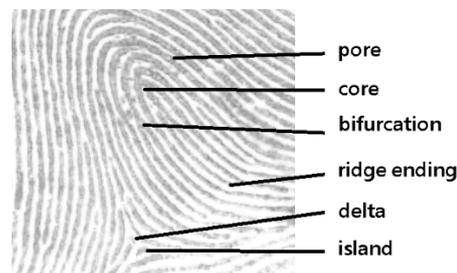


Fig. 5. Minutiae is considered local pattern indicated on the images.

```

/*****
File Name: IHead.h
Package: NIST Internal Image Header
Author: Michael D. Garris
Date: 2/08/90
*****/
/* Defines used by the ihead structure */
#define IHDR_SIZE 288 /* len of hdr record (always even bytes) */
#define SHORT_CHARS 8 /* # of ASCII chars to represent a short */
#define BUFSIZE 80 /* default buffer size */
#define DATELEN 26 /* character length of data string */

typedef struct ihead{
char id[BUFSIZE]; /* identification/comment field */
char created[DATELEN]; /* date created */
char width[SHORT_CHARS]; /* pixel width of image */
char height[SHORT_CHARS]; /* pixel height of image */
char depth[SHORT_CHARS]; /* bits per pixel */
char density[SHORT_CHARS]; /* pixels per inch */
char compress[SHORT_CHARS]; /* compression code */
char complen[SHORT_CHARS]; /* compressed data length */
char align[SHORT_CHARS]; /* scanline multiple: 8|16|32 */
char unitssize[SHORT_CHARS]; /* bit size of image memory units */
char byte_order; /* 0->highlow | 1->lowhigh */
char sigbit; /* 0->sigbit first | 1->sigbit last */
char pix_offset[SHORT_CHARS]; /* pixel column offset */
char whitepix[SHORT_CHARS]; /* intensity of white pixel */
char igned; /* 0->unsigned data | 1->signed data */
char row_maj; /* 0->row maj | 1->column maj */
char top_bot; /* 0->top2bottom | 1->bottom2top */
char lr_rl; /* 0->left2right | 1->right2left */
char parent[BUFSIZE]; /* parent image file */
char par_x[SHORT_CHARS]; /* from x pixel in parent */
char par_y[SHORT_CHARS]; /* from y pixel in parent */
}IHEAD;
    
```

Fig. 6. The image header structure of NIST-14 dataset.

### C. Image Information in NIST-14

Each NIST-14 fingerprint image contains a header and the header structure is shown in Fig. 4. The header values of the NIST-14 s0022700 attributes are shown in Fig. 6. The resolution of the NIST-14 fingerprints is 500 pixels per inch (ppi) and the raw image format is stored in an 8-bit grayscale Wavelet Scalar Quantization (WSQ) format with size 832x768 pixels. In order to display such fingerprint images on the browser, all the 2700 files were converted from WSQ

into JPG format. This paper utilized the following attributes from the header:

- Identity: It includes the file name, gender female (f) or male (m), scanned by ink (i) or live (l), and the type of this particular fingerprint. Fingerprints are classified into 19 types (classes) and class codes are shown in Fig. 1. This paper used code (ul) to represent the type of Ulnar Loop Ridge Counts (01-49), and code (rl) to represent the type of Radial Loop Ridge Counts (51-99). Please note that some fingerprints might have multiple codes because they were difficult to be classified into one single type.
- Parent: It includes the image original source, subject id, which finger and the original size. The example shown in Fig. 7 indicates that the original size of fingerprint s0027000 is 4096x1536 and it was extracted from tape227, and the subject id: t0419047, and it is the 10<sup>th</sup> finger (left little) indicated on the layout shown in Fig. 8.

```

Identity      : s0027000.wsq m i pi
Header Size  : 288 (bytes)
Date Created  : Fri Apr 23 00:27:31 1993
Width        : 832 (pixels)
Height       : 768 (pixels)
Bits per Pixel : 8
Resolution   : 500 (ppi)
Compression  : 7 (code)
Compress Length : 36968 (bytes)
Scan Alignment : 8 (bits)
Image Data Unit : 8 (bits)
Byte Order   : High-Low
MSBit       : First
Column Offset : 0 (pixels)
White Pixel  : 255
Data Units   : Unsigned
Scan Order   : Row Major,
              Top to Bottom,
              Left to Right
Parent       : tape227.t0419047.10 4096x1536
X Origin     : 0 (pixels)
Y Origin     : 0 (pixels)
    
```

Fig. 7. The header values of the fingerprint s0027000.

1. R. Thumb	2. R. Index	3. R. Middle	4. R. Ring	5. R. Little
6. L. Thumb	7. L. Index	8. L. Middle	9. L. Ring	10. L. Little

Fig. 8. Layout of fingerprint card and numbers.

The system developed by this paper utilized the header attributes Identity and Parent to quickly track and retrieve all 10 finger images for the same person in the NIST-14 dataset.

### III. ARCHITECTURE

The system was designed in a 3-tier architecture – front-end GUI on the browser, middle-end application Apache web server and back-end MySQL database server as shown in Fig. 9. Both servers are based on Fedora Linux system. The challenging task is to develop programs interacting between front-end and middle-end in real-time.

#### A. Front-End Functions

HyperText Markup Language (HTML) and JavaScript jQuery are adopted to develop the front-end interface and associated functions. Several web applications and platforms

[13]-[15] were implemented for fingerprinting information and recognition. In order to interact with the images, it is necessary to have a listener (handler) to catch and process mouse event including mouse move, click down and up. Since HTML is defined only for formatting the display, the event handlers have to be done by JavaScript. The system interface layout is shown in Fig. 10. The fingerprint information is also shown on the same web page.

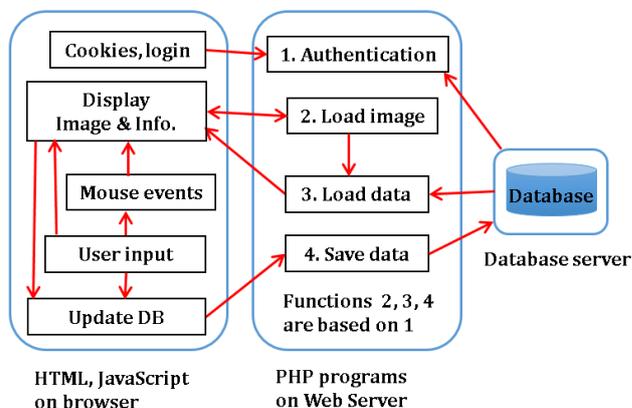


Fig. 9. The 3-tier system architecture and the interactions among browser, web server and database server.

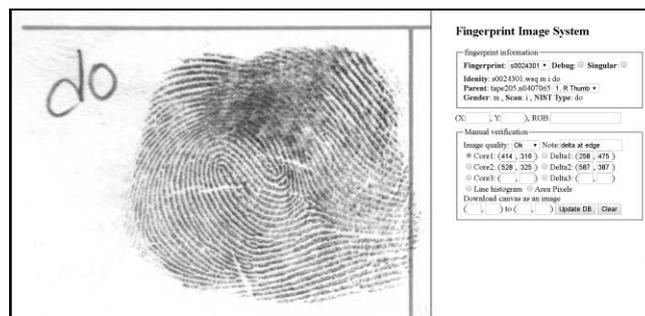


Fig. 10. The main GUI of the fingerprint system.

The system has implemented an authentication mechanism that requires users to login before using any feature. The user information is stored on the back-end database server. The front-end interface layout is divided into 2 parts – left panel is a HTML canvas to display the complete fingerprint image and right panel is the information section. Users can see the fingerprint image by selecting a fingerprint or the finger. The content of Identity, Parent, Gender, Scan and Type will be displayed immediately when a fingerprint is selected.

There are two checkboxes on the main GUI: Debug and Singular. If Debug is checked, the debugging information will be displayed in the browser.

If the Singular checkbox is checked, the system will display the singular points on the fingerprint if their positions are stored in the database. The core is displayed in a smaller circle and delta in a triangle shape. The singular points are labeled in sequential numbers as shown in Fig. 11. The numbers of these radio boxes are associated with the number on displayed image as shown in Fig. 11.

The grayscale intensity value of a pixel and mouse current position on the image will be displayed under the fingerprint information panel when the user moves the mouse. The system can handle at most 3 cores and 3 deltas. The position will be copied to the corresponding checked radio box of core or delta point when the mouse is clicked. A circle and triangle

in blue color centered at the clicked position represent the core and delta, respectively. The corresponding number of singular points will be displayed above the shape as shown in Fig. 11.

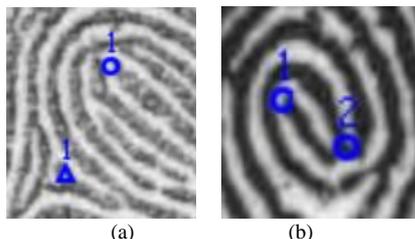


Fig. 11. (a) A core is displayed in a circle and delta is in a triangle. Both are labeled 1. (b) Two cores are displayed with labeled 1 and 2.

From the dropdown list, the user can select one from three different types of image quality: “Good”, “Ok” and “Bad”. The default quality is “Null” which means the image has not been reviewed by human experts. Users can also type some notes in the “Note” input box. The positions and types of the singular points will be saved into the database when the user clicks “Update DB” button. The “Clear” button will reset these textboxes into empty.

The front-end interface is implemented in JavaScript jQuery. The mouse events are caught and passed to the programs on web server by Ajax. The graphical functions are very helpful for users to verify the number of singular points, and their types and positions. Users can easily update the database if the positions and type are not correct.

### B. Middle-End Web Server

The authentication mechanism of this system is implemented in PHP programs on the web server. The authentication process is done by programs that will retrieve the user information from the database server. If a user logs in successfully, the user id will be kept in a cookie on the browser. All programs on the web server will first verify the cookie when they are called to execute the functions for view the histograms and pixel values.

```
function displaySingularbyFP(myfplist) {
    var datastring='myfplist='+myfplist;
    var type,seq,x,y;
    var rec;
    $.ajax({
        type: 'post',
        url: 'selectsingular.php',
        data: datastring,
        dataType: 'json',
        success: function(rec){
            if (rec.length>=1) {
                for (i=0;i<rec.length;i++) {
                    seq=rec[i].seq;
                    type=rec[i].type;
                    x=rec[i].xpos;
                    y=rec[i].ypos;
                    $('#'+type+seq+' x').val(x);
                    $('#'+type+seq+' y').val(y);
                    drawSingular(seq,x,y,type);
                    if (document.getElementById("myDebug").checked)
                        console.log('seq:'+seq+', type:'+type+', x,y:'+x+', '+y);
                }
            }
            else{
                if (document.getElementById("myDebug").checked)
                    console.log(result[0]);
            }
        }
    });
}
```

Fig. 12. An ajax function in JavaScript on the browser calls a PHP program selectsingular.php on the web server by passing a fingerprint name and display the returned the singular point sequence, type and coordinates at corresponding HTML IDs on the browser.

The original fingerprint JPG files are stored on the middle-end web server. The user selected image will be sent from the web server to the browser along with the fingerprint information retrieved from the database. The front-end JavaScript programs use Ajax POST method to call the PHP programs by passing fingerprint name and singular point IDs, and receive the results to display on the browser. Since the information should be shown on the same web page, the ID of HTML tags are used to control the display. Figure 12 shows part of the source code on how the front-end JavaScript communicate with the back-end PHP program.

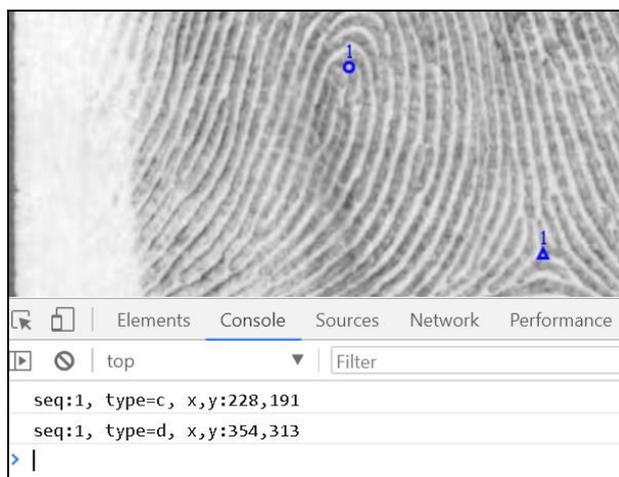


Fig. 13. If the debug mode is checked, the Ajax function will post the message returned from the PHP program on the browser console.

If the program selectsingular.php runs successfully and at least one singular point is found in the database for the given fingerprint name, the Ajax program will display the returned values of the number of singular points, type and x, y coordinates on the corresponding HTML tag IDs. If there is no singular point data returned, the PHP program will return a message which will be displayed in the browser console as shown in Figure 13 if the debug mode is checked. The Chrome browser console can be opened by pressing the “Ctrl”, “Shift” and “J” together.

### C. Back-End Database Server

All the fingerprint and singular points information and relationship are stored in several tables on the database server.

The table *NIST14* has attributes (fname, fdatetime, identity, gender, scan, NIST\_type, loopct, height, width, resolution, bpp, parent, finger, ncore, ndelta, small\_image, path, quality, note, iheader) to keep all information for each fingerprint. The attribute fname is the primary key of this table.

The table *NIST14\_Singular* table has attributes (sid, seq, fname, type, xpos, ypos) to keep the singular point information for each singular point on the fingerprint fname which is a foreign key referencing to the fname column in table *NIST14*. The column sid is the primary key with auto increment. The column seq indicates the sequence of core or delta on that fingerprint.

The table *NIST\_class\_code* has attributes (name, code, note) based on Fig. 1. The column code is the primary key. The code “ul” to represent the name “Ulnar Loop Ridge Counts” with note “01-49”, and “rl” to represent the name “Radial Loop Ridge Counts” with note “51-99”.

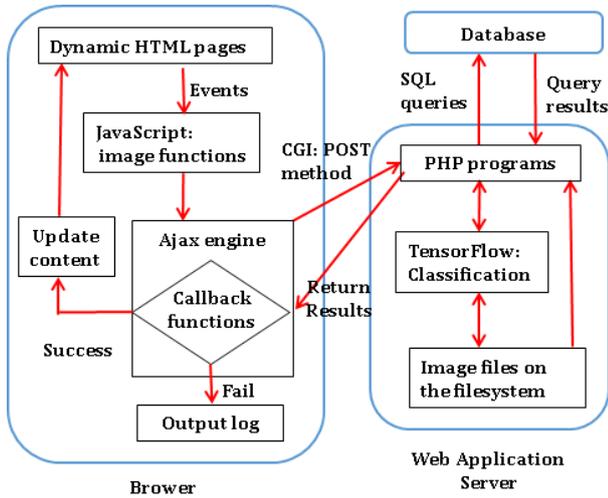


Fig. 14. The function and data flow between the browser and the web applications of the 3-tier architecture shown in Fig. 9. The design will integrate with TensorFlow for further research.

D. TensorFlow Integration

The proposed system provides a platform to integrate with the TensorFlow for the fingerprint image classification and recognition. Fig. 14 shows the control and data flow between the web pages and the application of the proposed architecture. The PHP programs can invoke the external TensorFlow script using the shell\_exec() function and read the results returned from TensorFlow. Ajax has a callback function that allows developers to handle whether the PHP programs run on the server was executed failed or successful. If it failed, the JavaScript program output a message to the console as shown in Fig. 11. If it ran successfully, the results of TensorFlow scripts can be returned in JSON format and the web page content can be updated accordingly.

IV. ANALYTICAL TOOLS

The system also provides some analytical tools to study the fingerprints. Users are able to draw a line and visualize the histogram as shown in Fig. 15. The histogram graph will help users have better understating about the pixel intensity distributions. Users can also select an area to see the pixel values as shown in Fig. 16. Many images in NIST 14 have bad quality so it is very difficult to differentiate the ridge and valley. The imaging related functions provided by the proposed system allow users to see the pixel insensitive values that can help for further research to develop better algorithms for segmentation.

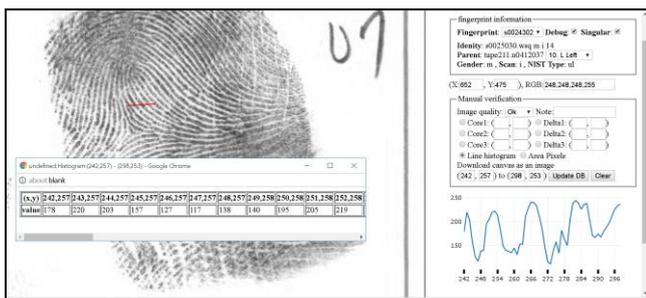


Fig. 15. The histogram line graph is shown at right bottom when user draw a line highlight in red on the image. The pixel intensity values will be also shown in a 1-D table in a separate window.

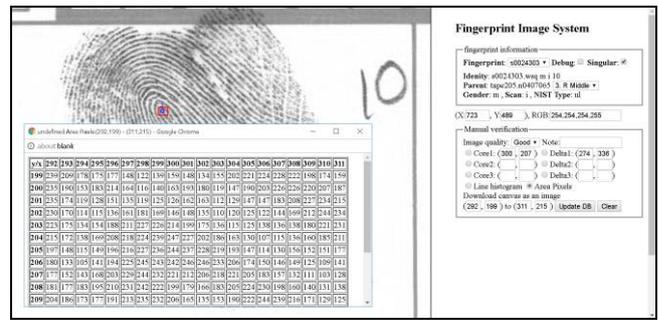


Fig. 16. The pixel intensity values are show in a 2-D table in a separate window when an area highlighted in red is selected on the image.

V. CONCLUSIONS

The framework of the proposed online system has been successfully developed and tested. More than 1500 fingerprints of NIST 14 image sets have been manually reviewed through the system. The singular point types and positions were identified and saved into the database.

The system can significantly help researchers to visualize the fingerprints and understand the different fingerprint image problems might face for further research. Not only many images have bad quality, but also the fingerprint have cut, blank lines, and white spots, etc. In addition, it allows researchers in different institutes to work on the same fingerprint.

The graphical tool and the stored information will help to verify further research on developing better algorithms to automatically detect the singular points, classify the fingerprints and improve the segmentations.

The future works will also include more graphical functions to display the intermediate results of the new algorithms, and compare the results between saved data verified manually and automated methods.

REFERENCES

- [1] C. Y. Huang, L. M. Liu, and D. C. D. Hung, "Fingerprint analysis and singular point detection," *Journal of Pattern Recognition Letters*, vol. 28, no. 15, 2007, pp. 1937-1945.
- [2] An open source machine learning framework for everyone. [Online]. Available: <https://www.tensorflow.org/>
- [3] J. M. Shrein, "Fingerprint classification using convolutional neural networks and ridge orientation images," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, USA, 2017.
- [4] W.-S. Jeon and S.-Y. Rhee, "Fingerprint pattern classification using convolution neural network," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 3, pp. 170-176, 2017.
- [5] A. Perrier, "Tensorflow with Javascript brings deep learning to the browser," *InfoQ Weekly Newsletter*, April 18, 2018.
- [6] J. Toth, L. Kovacs, B. Harangi, C. Kiss, A. Mohacsi, Z. Orosz, and A. Hajdu, "An online benchmark system for image processing algorithms," in *Proc. 5th IEEE International Conference on Cognitive Infocommunications*, Vietri sul Mare, Italy, November 5-7, 2014.
- [7] K. Senthilkumar, N. K. Vivek, and E. Vijayan, "An efficient image processing method based on web services for mobile devices," in *Proc. 14th International Conference on Science, Engineering and Technology (14th ICSET-2017)*, Vol 263, Vellore, Tamil Nadu, India, May 2017.
- [8] S. ReddyBorraa, G. J. Reddyb, and E. S. Reddy, "Classification of fingerprint images with the aid of morphological operation and AGNN classifier," *Applied Computing and Informatics*, 2017.
- [9] C. I. Watson, *NIST Special Database 14 Mated Fingerprint Card Pairs 2*, National Institute of Standards and Technology, Advanced Systems Division, Image Recognition Group, September 15, 1993.
- [10] V. S. Srinivasan and N. N. Murthy, "Detection of singular points in fingerprint images," *Pattern Recognition*, vol. 25, issue 2, February 1992, pp. 139-153.

- [11] E. R. Henry, "Classification and uses of fingerprints," *G. Routledge*, London, 1900.
- [12] Federal Bureau of Investigation, "The science of fingerprints: Classification and uses," *U.S. Government Printing Office*, Washington, D.C., 1984.
- [13] C. Blakemore, J. Redol, and M. Correia, "Fingerprinting for web applications: From devices to related groups," *2016 IEEE Trustcom/BigDataSE/ISPA*, 23-26 Aug. 2016.
- [14] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, "Fingerprinting information in JavaScript implementations," in *Proc. W2SP 2011*, IEEE Computer Society, May 2011.
- [15] E. Okoh *et al.*, "Toward online education for fingerprint recognition: A proof-of-concept web platform," *Information Security Journal a Global Perspective*, June 2017, pp. 186-197.



**Ching-Yu Huang** is an assistant professor of the School of Computer Science at Kean University, Union, New Jersey, USA since September 2014. Dr. Huang received a Ph.D. in Computer & Information Science from New Jersey Institute of Technology, Newark, New Jersey, USA.

Prior to joining Kean University, Dr. Huang had more than 16 years of experience in the industry and academics in software development and R&D in bioinformatics. His research focuses SNP genotype calling and cluster detection; image processing and pattern recognition, especially in microarray and fingerprint; geotagged images and location information reconstruction; database application development; data processing automation; E-learning, educational multimedia, methodology, and online tools for secondary schools and colleges. Dr. Huang has more than 40 publications in journals and conferences and more than 20 presentations in workshops and invited lectures.



**Limin Liu** received his master degree from the Syracuse University in 1994 and PhD in computer and information science from the New Jersey Institute of Technology in 1999. He is currently an associate professor in the Department of Information Management, Shih Hsin University, Taipei, Taiwan. His current research interests include fingerprint technologies, image processing, machine learning, deep learning, and computer assisted language learning.



**Yui-Liang Chen** is a professor in Shih-Hsin University, Taiwan. He was born in Taiwan. He received his B.S. from the National Cheng-Kuang University, Taiwan, in 1985; the MS and the Ph.D. degrees in computer and information science from New Jersey Institute of Technology, Newark, New Jersey, in 1990 and 1995. He has been on the faculty of the Department of Information Management, Shih-Hsin University, Taipei, Taiwan, since 1995. He was the director of computer center in Shih-Hsin University from 1999 to 2008. Later on, he was the chair of the Department of Information Management, Shih-Hsin University from 2011 to 2014. His research interests are in the areas of Enterprise Resource Planning (ERP), Web Accessibility, Web Applications, and Image Processing. He is a member of the Chinese Society of Information Management, a member of the Chinese Enterprise Resource Planning Society, and a member of the Information Service Association of Chinese Colleges. He is leading auditors of both ISO-27001 and BS-10012.