Multimed Tools Appl (2019) 78:3649–3688
https://doi.org/10.1007/s11042-017-5537-5

Impact of digital fingerprint image quality on the fingerprint recognition accuracy

Mohammad A. Alsmirat[1] · **Fatimah Al-Alem**[1] · **Mahmoud Al-Ayyoub**[1] · **Yaser Jararweh**[1] · **Brij Gupta**[2]

Received: 7 August 2017 / Revised: 17 November 2017 / Accepted: 14 December 2017 /
Published online: 20 January 2018

Abstract Despite the large body of work on fingerprint identification systems, most of it focused on using specialized devices. Due to the high price of such devices, some researchers directed their attention to digital cameras as an alternative source for fingerprints images. However, such sources introduce new challenges related to image quality. Specifically, most digital cameras compress captured images before storing them leading to potential losses of information. This study comes to address the need to determine the optimum ratio of the fingerprint image compression to ensure the fingerprint identification system's high accuracy. This study is conducted using a large in-house dataset of raw images. Therefore, all fingerprint information is stored in order to determine the compression ratio accurately. The results proved that the used software functioned perfectly until a compression ratio of (30–40%) of the raw images; any higher ratio would negatively affect the accuracy of the used system.

Keywords Fingerprint recognition · Digital cameras · Raw images · JPEG compression

✉ Mohammad A. Alsmirat
masmirat@just.edu.jo

Mahmoud Al-Ayyoub
maalshbool@just.edu.jo

Yaser Jararweh
yijararweh@just.edu.jo

[1] Computer Science Department, Jordan University of Science and Technology, Irbid, 22110, Jordan

[2] National Institute of Technology Kurukshetra, Kurukshetra, India

1 Introduction

In the current digital age, the ubiquity and the wide reliance on the modern technologies has helped in facilitating counter-theft procedures. Finding such counter-theft solutions is becoming a more stressful issue with the current increase in the crime rates. One technology example is to utilize the electronic fingerprint verification systems that can be used with different automated systems such as ATMs, smart buildings and transportation systems, etc. Because of this, automated fingerprint identification is receiving continuously growing interest from both academic and commercial parties.

The importance of focusing on fingerprints, as a main biometric tool to uniquely identify people, stems from many reasons. The first one is that no two individuals (even identical twins) have the same fingerprints. Another reason is that fingerprints do not change with age. Finally, each finger of a single individual has a different fingerprint. This fact can be utilized to identify not only the person who touched a certain object, but also the finger used by that person.

As shown in Fig. 1, a fingerprint is formed from the following parts.

- Ridge Endings: the terminals of the ridges.
- Bifurcations: the point of dividing ridge into two branches.
- Dots and small ridges.
- Islands: random point with no connectors.
- Pore: a white point surrounded by black point.
- Bridges: small ridges joining two longer adjacent ridges.
- Crossovers: two ridges which cross each other.
- Core: center of the fingerprint.
- Deltas: points, at the lower part of the fingerprint, surrounded by a triangle.

Fingerprints can be classified according to their shapes into three main types: Whorl, Loop and Arch. Each class has its ridge flow and pattern. Figure 2 shows these three types with their existence percentages. Loop fingerprint are the most common with a percent of 60–65% of all fingerprints. Arch fingerprints, which are rare, account for only 5% [58].

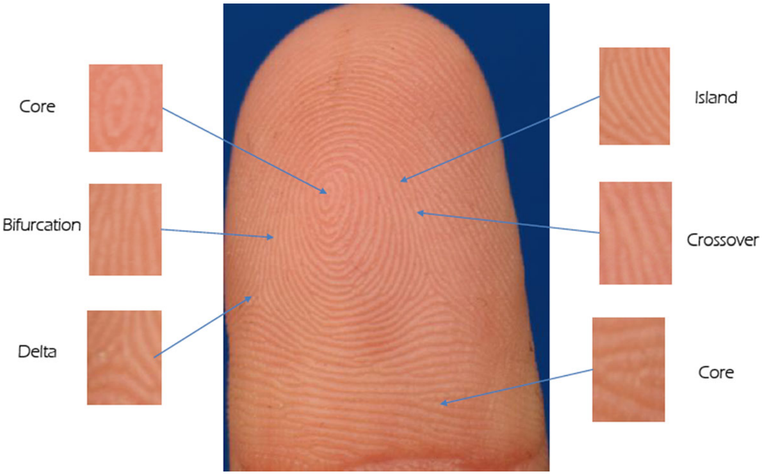
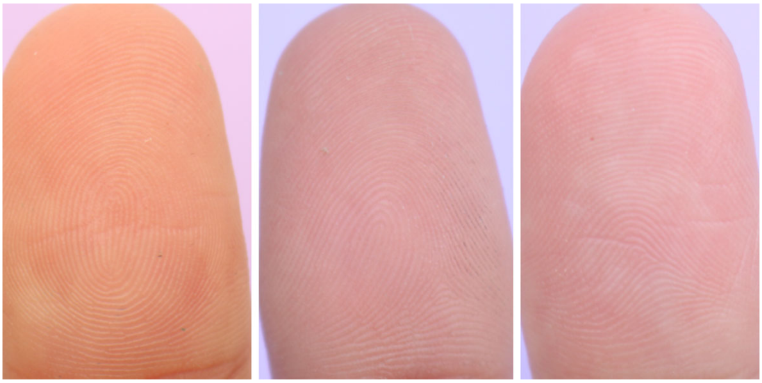


Fig. 1 A Fingerprint clarification



(a) Whorl (30%) (b) Loop (65%) (c) Arch (5%)

Fig. 2 Fingerprint types

A typical Automatic Fingerprint Identification System (AFIS) usually consists of many stages that include: image acquisition, fingerprint segmentation, image preprocessing (enhancement), feature extraction, and matching result [17]. Below, we briefly define each step.

- Image Acquisition: The stage of image acquisition is one of the most important factors that contribute in the success of the recognition system. Here, there are several factors to be taken into account before choosing the device that will be used to capture fingerprints in terms of performance, cost and size of the resulting image. In this study, we focus on the fingerprint capture phase only.
- Segmentation: The basic idea in this step is to separate the image of the finger from the background. It is an initial step of any recognition systems [31, 39, 56, 59].
- Preprocessing: This phase includes some modifications to the image with the aim of enhancement. This would increase the accuracy of the recognition system.
- Feature extraction and Person Verification: At last, similar fingerprints are retrieved. So, only this set of fingerprints (source) are compared with the target fingerprint [7, 18, 44, 57].

Fingerprints are acquired in different ways. For non-automated (or non-online) systems, fingerprints are usually acquired by stamping inked fingertips on a special paper. If these fingerprints need to be stored on a digital device, an image of the paper is taken using a scanner for example. On the other hand, fingerprints are acquired using fingertips scanners for automated systems. Scanners may use different technologies such as optical, capacitive, thermal, and ultrasonic. These scanners usually need the fingertip to be pressed or dragged across some surface. This may reshape or squeeze the resulting fingerprint which may lead to incorrect identification. Also, this makes it impossible to cross use different finger tips scanners in acquiring and verification process as different scanners may results in different reshaping results.

As most modern automated systems are equipped, or can be easily equipped, with high quality digital cameras, a recent direction of research started to study the possibility of acquiring fingerprints using these cameras. According to the studies in [2, 19, 30, 40, 43, 44, 55], this approach is promising and the results are acceptable.

To design different techniques and study the impact of different factors related to automated fingerprint recognition systems, a large fingerprint image set is needed [23]. There are many factors that may affect the accuracy of the recognition process. Most digital cameras perform JPEG compression on the images they capture before storing them. JPEG compression can be applied using different factors, called JPEG quality, which is represented by a number between 1 and 100. The higher the JPEG quality factor the better the resulting image clarity and the larger the image size. If compression is not applied, image sizes become very large and storing these images becomes very expensive. For example, the size of an image captured by 16 mega pixel camera (that represents a pixel with 8 bits) is 16 megabytes. While compressing this image using JPEG compression with quality factor 50 may result in an image of size 200 kilobytes. Another factor that may impact the fingerprint recognition accuracy is the complexity of the background in the image.

To study the impact of any image quality factor on the finger recognition process, a very high quality fingerprint image set is needed. Preferably, the images need to be in what is known as their raw format. If this image set is available, we can easily apply any image distortion factor, with different levels, and calculates the accuracy of recognition algorithm using images distorted with these different levels. This study can be used to find a relationship between distortion levels and recognition accuracy. For example, we can compress the fingerprint images with different JPEG quality factors to study the relationship between the compression level and the recognition accuracy.

In the study [1], a survey was conducted about the available fingerprint image sets and the acquisition devices used to collect them. The conclusion in that study is that the available fingerprint datasets acquired using digital cameras are all taken with built in degradation factors due to specific lighting, different background, low camera quality, image compression, etc. In other words, the available fingerprint databases are far from being suitable for testing the impact of different customizable degradations. In this work, we build a digital camera acquired fingerprint set in perfect conditions that can be customized easily and can be used for testing the impact of different degradations on the accuracy of different fingerprint recognition algorithms. We also experiment with a fingerprint recognition system to study the impact of some factors such as image compression and subject gender.

The rest of this paper is organized as follows. Section 2 discusses the related works on the impact of image compression on various recognition tasks. Section 3 gives an overview of the most common devices used for online fingerprint acquisition. In Section 4, we describe briefly some of the most common fingerprint databases, while a detailed description of our dataset is given in Section 5. The fingerprint recognition system we use is discussed in Section 6 and the experiments and their results are discussed in Section 7. Finally, a conclusion of the paper along with future directions of this work are given in Section 8.

2 Related work

Image quality is very important for the biometric identification [8]. Unfortunately, usually, the better the image quality, the larger the image size which may require more storage space. In this section, we discuss a number of studies concerning the importance of image quality and image compressing operations and their impact on system performance.

Many studies in the literature analyzed digital image compression impact on image quality in general [45, 52]. On the other hand, only few studies analyzed image compression impact on biometric recognition algorithms other than fingerprint recognition [8, 11, 36].

The iris of the eye is one of the biometric features that are commonly used. Iris recognition depends entirely on the quality of the captured iris image. In 2008, Daugman, John, and Cathryn Downing [8] studied the effect of compressing images on the performance of iris recognition systems. They presented an approach that combines region-of-interest isolation with JPEG and JPEG2000 compression at intense levels. Results showed that it is possible to compress iris images to less than 2000 bytes with little impact on the performance identification. Based on previous ratios change in order of bits in the iris code templates are small. According to the study, the impact was between 2% to 3%. In the summary of their study, they concluded that the compression ratio 150 : 1 from original image does not affect the system's performance. Image features and details remain without being affected negatively.

In [38], the authors presented the impact of age to identify people through fingerprints, they divided the images into four different age groups, and they concluded that the details extracted depend on the age group. With each age group, the skin is different in terms of smoothness and dryness. Age group affected the performance of the system significantly. The results showed that the images of people aged 60 years and older achieved better performance. The researchers used an optical sensor to collect their database. Moreover, they used an available commercial assessment tool that counts number of minutiae.

The conclusion in [38] is also confirmed by Modi at el. in [37]. They presented research about measuring the impact of the quality of the fingerprint image on identification of two samples of two different ages: the first sample age is between eighteen and twenty-five and the second sample age is 62 and more. In their study, they used two matching algorithms; Neurotechnologija VeriFinger and bozorth3. Moreover, they used fingerprint Image Software (NFIS) to analyze the quality of the images in there experimental results. False Matches were used for confirming that the old sample fingerprints has the highest number of minutiae compared to the fingerprints of the younger sample.

Other studies were focused on comparing different methods for compression images. In particular studies, they designed a fusion system that combined more than one way to identify individuals, like the combination between the face and the fingerprint that presented in [11, 36]. These studies measured the impact of compression on the images when using many common compression algorithms such as jpeg and jpeg2000. The main difference between their work and ours is that they were interested in comparing between different compression methods only.

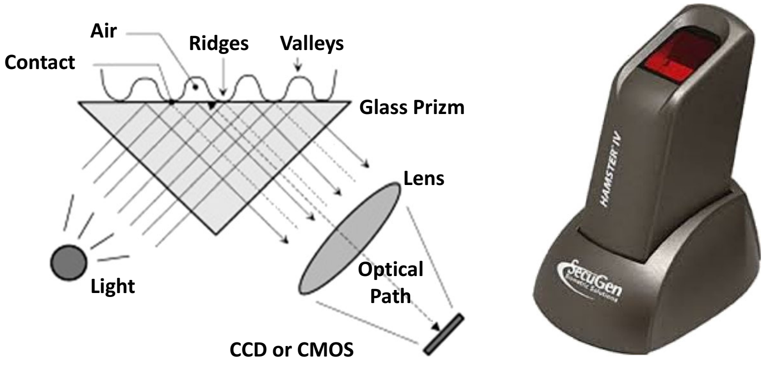
3 Fingerprint acquisition devices

This section gives an overview of the most common devices used for online fingerprint acquisition. Many technologies have been utilized in these devices. Namely, optical, capacitive, thermal, and ultrasonic.

3.1 Optical sensor

Optical fingerprint devices are the oldest and the most commonly used devices to capture fingerprints. Figure 3a[1] clarifies how this sensor works. The fingerprint image is captured by placing a finger on one side of a prism, a light source that emits light vertical on the second prism side, and a camera that is placed in front of the third prism side. The light collected

[1] http://students.iitk.ac.in/projects/roboticsclub_fingerprint



(a) Clarification of how Optical Sesors work

(b) A Comercial optical sesor based Fingerprint Acquisition Device

Fig. 3 Optical technology

by the camera is then converted into a digital signal where bright areas reflect the valleys and dark areas reflect the ridges. This sensor was used to acquire a large number finger print databases, such as FVC2000 (DB1, DB3) [35], FVC2002 (DB1 and DB2) [33], FVC2004 (DB1 and DB2) [34], FVC2006 (DB2) [5]. Figure 3b is an example of a commercial optical sensor. The main advantages of optical sensors are temperature insensitivity, low cost, and high resulting fingerprint resolution. On the other hand, it suffers form many disadvantages. As optical sensor depends on light reflection, as a result, its functionality may suffer from lighting environment. Also, optical sensors must be of a certain large size to be able to generate a high quality image [50]. Moreover, as the finger should be placed in contact of some plate (plastic or glass), the residue of the previous fingerprint on the sensor may impact the quality of the current fingerprint image as they may overlap. Furthermore, this residue can be used to steal the fingerprint.

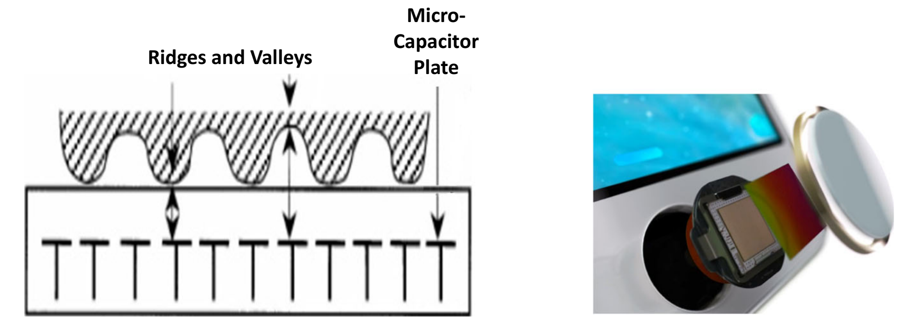
One of the commercial examples that uses this technology is shown in Fig. 3b is the SecuGen Hamster IV.[2] it is one of the common devices in fingerprint systems, which have been certified by the FBI and STQC. Beside SecuGen Hamster IV, other optical devices were used to collect the IIIT-D Multi-sensor Optical and Latent Fingerprint [49].

3.2 Capacitive sensor

This device is based on the principle of measuring the electric capacity of the human skin. Fingerprints are placed on a very thin plate that contains a set of micro-capacitors. Skin of a fingerprint contains ridges and valleys. The capacitive sensor basically measure the capacity of the skin of the finger. The capacity of ridges is different form the capacity of valleys because the capacity depends highly on the distance of the skin from the micro-capacitor plate. A simple prototype of a capacitive sensor is shown in Fig. 4a.[3] This sensor was used to capture a number of fingerprint datasets that includes FVC2000 (DB2) [35] and FVC2002 (DB3) [33].

[2] http://www.secugen.com/products/ph4.htm

[3] http://www.accu-time.com/ats-biometric-devices/fingerprint-scanner-types/



(a) Clarification of how Capacitive Sesors work

(b) The iPhone Capacitive Sensor

Fig. 4 Capacitive technology

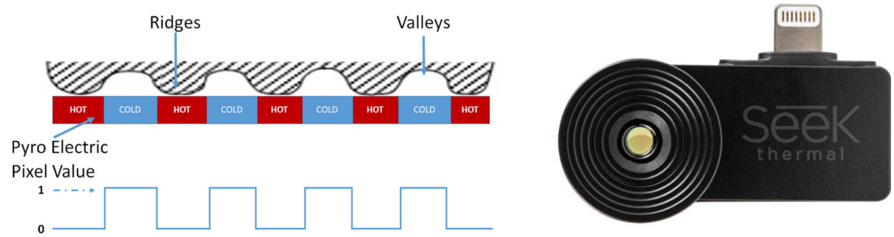
An advantage of capacitive sensors are small in size and therefore they are cheaper to produce. They also consumes low energy [24]. This advantage is particularly useful in the battery powered devices. It also insensitive to environmental effects such as sun light. Many manufacturers, such as Sony, Fujitsu, Hitachi, and Apple, have adopted this technology in their fingerprint recognition systems.

3.3 Synthetic fingerprint generation

This method automatically generates artificial fingerprints based on a model. This model usually is controlled by a set of parameters which can be changed to produce different fingerprints in terms of clearness, area, centered or not, rotation, the difficulty of the recognition of the fingerprint, and many more. This method can be used to build very large fingerprint databases in short time with almost no cost. Also, the generated artificial fingerprints may be tuned to simulate natural fingerprints. This method was used in some fingerprint databases such as FVC2000 (DB4) [35], FVC2002 (DB4) [33], FVC2004 (DB4) [34], FVC2006 (DB4) [5].

3.4 Thermal sensor

This sensor is based on pyro electric material. This material has the ability to change temperature differences to different voltages. By the difference between the temperatures of the



(a) Clarification of how Termal Sensors work Example

(b) A comercial Termal Sensor Example

Fig. 5 Thermal technology

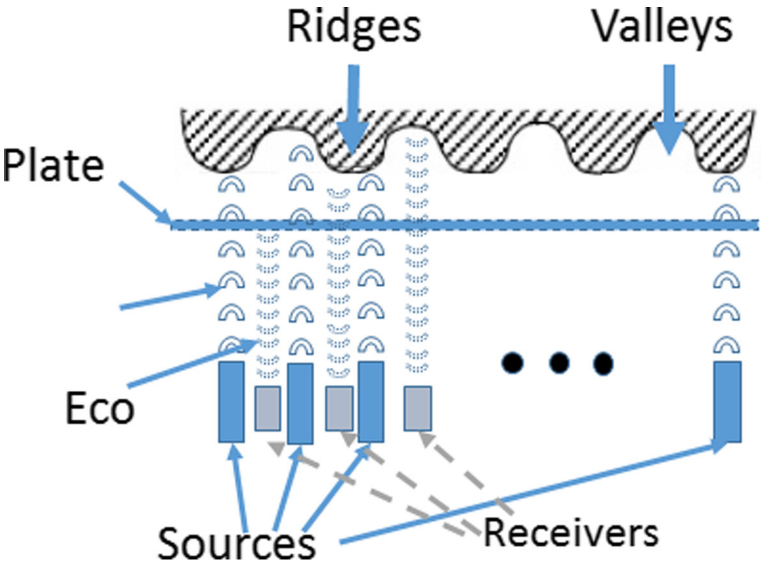


Fig. 6 Clarification of how ultrasonic sensors wok

ridges and the valleys, the voltage generated for each one is different from the other. One of the most common thermal sensors is “Atmel FingerChip” which is used to acquire many well known fingerprint databases. Namely, FVC2004 (DB3) [34] and FVC2006 (DB3) [5].

This technology is not very common commercially. One of the difficulties faced when dealing with this technology is that the thermal image should be captured in a very short time. That is because when a finger is placed over the pyro electric material, the temperature difference between the areas under the ridges and the valleys will be initially detectable. Eventually, after a short period of time, the temperature difference vanishes as the adjacent areas heat each other. As this method depends on body temperature, it is very hard to be deceived (Fig. 5).

3.5 Ultrasound sensor

This device is based on the principle of sending high frequency sound waves, and then capturing the reflection from the finger surface. A clarification of how this technology works is shown in Fig. 6. The advantage of this technology is its ability to pass through dust, dirt, and skin oil that may cover the finger surface. For this reason, fingerprint ultrasound sensing may be used to capture high quality images if if the finger is dirty. On the other hand, this technique has many disadvantage. For example, it is very expensive and may take long time to capture the fingerprint image. Qualcomm[4] are utilizing ultrasonic technology in their new processor (snapdragon 820) security features.

[4] https://www.qualcomm.com/products/snapdragon/security/sense-id

3.6 Digital cameras

Many digital camera types are used in the literature of fingerprint systems. We categorize them into webcams [2, 19, 20, 40, 43], mobile phone cameras [9, 26, 28, 30, 44], and advanced digital cameras [22, 55]. Webcams are digital cameras that are usually low in resolution and very cheap in price. These cameras comes usually built-in personnel computers for video chatting purposes. Because these cameras have low resolution, it is very hard to get the fingerprint details from an image. Therefore, many researches proposed many image pre-processing techniques to overcome this problem.

In the last few years, mobile phones cameras have witnessed a huge improvement which have enabled them to be used to acquire fingerprints. Current resolution of mobile phones have reached 8 mega pixels on average. However, there are a set of criteria that should be considered when taking fingerprint images using these devices, such as lighting, distance between the finger and the camera, the appropriate focus that should be chosen, and the image preprocessing steps.

Advanced digital cameras, also, have been studied and used to acquire fingerprint images. These cameras produces very high resolution images. In these cameras, image acquisition is highly flexible. They can also apply advanced image enhancement and compression algorithms.

4 Fingerprint databases

In this section, we describe briefly some of the most common fingerprint databases:

FVC2000 database is used in the First International Fingerprint Verification Competition. The database consists of four parts; DB1, DB2, DB3 and DB4 [35]. DB1 and DB2 contain images of 110 fingers with eight impressions per finger (880 images in total). The images of up to four fingers per each participant where taken. Fingerprints of the fore and the middle fingers of both hands are taken. Half of the fingerprints are for males. The ages of the individuals in these two databases are 20-30 years old. DB1 was acquired using a low-cost optical sensor (Secure Desktop Scanner) while DB2 was acquired a low-cost capacitive sensor (TouchChip by ST Microelectronics). According to [35], these two databases cannot be used to evaluate the accuracy of the biometric algorithms. That is because no cautions were taken to guarantee the quality of the images and the plates of the sensor were not systematically cleaned after each use In DB3, the fingerprints are acquired from 19 volunteers, of whom 55% are males. The images of six fingers are taken from each participant (thumb, fore, and middle of left and right hands). The ages of the individuals are between 5 and 73. DB3 images were acquired using an optical sensor (DF-90 by Identicator Technology). DB4 was automatically generated using synthetic generation.

FVC2002 database was used in the Second International Fingerprint Verification Competition [33]. It consists of four parts; DB1, DB2, DB3 and DB4. The number of images is the same as in FVC2000. In this database, four different types of technologies were used to acquire the fingerprints in the four different databases (i) an optical sensor “TouchView II” by Identix, (ii) an optical sensor “FX2000” by Biometrics, (iii) a capacitive sensor “100 SC” by Precise Biometrics, and (iv) synthetic fingerprint generation. 90 individuals participated in this database collection. In DB1, DB2, and DB3, images of different 30 participants were collected. Images of 4 fingers (the fore and the middle fingers of both hands) of each participant were collected. No any extra effort were taken to guarantee the quality of the images as the perpos of these images are just to be used in competition. DB4 was synthetically generated.

FVC2004 database was used in the Third International Fingerprint Verification Competition [34]. It consists of four parts DB1, DB2, DB3 and DB4. Each one of first tree parts contains images of different 30 persons. These persons are 24 years old on average. Fingerprints of the fore and the middle fingers of both hands were taken from each person. For each part, 120 fingers were considered with 12 impressions per finger for a total of 1440 images. In this database, four different techniques of of fingerprint acquisition were used: (i) an optical sensor (CrossMatch V300) for DB1, (ii) an optical sensor (URU 4000) by Digital Persona for DB2, (iii) a thermal sweeping sensor (Atmel FingerChip) for DB3, and (iv) synthetic generation (SFinGe v3.0) for DB4.

FVC2006 database was used in the Fourth International Fingerprint Verification Competition [5, 12] consists of four parts: DB1, DB2, DB3 and DB4. Each one of these four parts contains the images of 150 fingers, 12 impression of each finger. The four parts were acquired using Capacitive (Electric field) sensor, Optical Sensor, Thermal sweeping Sensor, and SFinGe v3.0, respectively. This fingerprint data set is a part of a large biometric database collected within the European Project BioSec [4].

CASIA-FingerprintV5 database [6] contains 500 subjects. Each subject contributed 40 fingerprint images of his thumb, index, middle, and ring fingers of both his left and right hands, with 5 images per finger. Different finger images defer in rotation or pressure. The database contains 20,000 fingerprint images in total. All fingerprints are stored as 8 bit gray-level BMP images. The finger print images were captured using the URU4000 optical fingerprint sensor in one session.

IIIT-D multi-sensor optical and latent fingerprint [49] contains 19200 images acquired from 100 subjects using different sensors. Table 1 contains the main details about this database.

IIIT-D Latent database [47] consists of images of all 10 fingers for 15 subjects. Latent fingerprints were taken using Canon EOS camera with resolution 4752x3168. Fingerprints were taken with different effects (dry, wet and moisture skin) as well as different backgrounds (tile and ceramic plate) with a total number of 1046 images. Those images are freely available.

Low-cost Equipment database [40] consists of 400 fingerprints of 6 participants captured using two webcams: Microsoft Lifecam VX3000 and Philips SPC630NC.

Smartphone Camera database [44] consists of 100 different fingerprints taken from 25 participants. Fingerprints of the index and the middle fingers of both left and right hands were taken. Fingerprints were captured using iPhone 4, Samsung Galaxy s1, and Nokia N8 mobile phones.

Table 1 IIIT-D multi-sensor optical and latent fingerprint

Subset	No. of Images	Capture Protocol	Comment
DB1	4000	100 users x 10 fingers x 2 sessions x 2 instances	Aquired using Lumidigm Venus IP65 Shell
DB2	4000	100 users x 10 fingers x 2 sessions x 2 instances	Aquired using Secugen Hamster-IV
DB3	1200	100 users x 3 slap prints x 2 sessions x 2 instances	Aquired using CrossMatch L-Scan Patrol
DB3_A	4000	100 users x 10 fingers x 2 sessions x 2 instances	Cropped images from DB3 using the NFSEG tool [41]
DB4	4400	100 users x 2 hands x 2 sessions x 11 instances	Cropped Latent fingerprint form DB5
DB5	1600	100 users x 2 hands x 2 sessions x 4 instances	Users left their fingerprints (multifingerprints) on a ceramic tile and then these latent fingerprints were captured using a USB programmable UEye camera and the setup in [48] after a black powder dusting process [29].

Camera-based fingerprint database [28] consists of fingerprints of 13 different subjects. Three samples were taken for each finger of the participating subjects. The total is 390 fingerprint images. The images were captured using Nokia 6700 classic and HTC magic mobile phones.

5 Description of our database

As shown in Section 4, none of the available fingerprint databases are suitable for our study goal which is to study the impact of digital image compression on fingerprint recognition. For this reason, we built our own high quality digital fingerprints database. To capture high quality fingerprint images, we used a digital single-lens reflex (DSLR) to capture the images and we made sure that the camera stores the images as raw images to avoid any compression. We use a Canon EOS 750D (also known as Canon EOS Rebel T6i) camera, which is fast and easy to use. Canon t6i provides high-resolution options and auto focus. In addition, it is equipped with Wi-Fi technology, which enables the users to easily access the captured images.

Our database, the Raw Fingerprint Image Database (RFID), contains (6,000) fingerprint images of 500 high school students and teachers, 90% of them aged between 13 and 18. Half of the volunteers were males, and the other half were females.

We have designed a high table to help the volunteer and the photographer fix their hands on and to install on different wooden backgrounds. The volunteers are asked to fix their hands on this high table to guarantee the quality and the clearance of fingerprints in the image. Twelve images were taken for each volunteer. Eight images were taken for the thumb and one image was taken for each of the index, middle, ring, and pinkie fingers of the volunteers' right hand. The images of the index, middle, ring, and pinkie fingers are captured in front of a wooden white background plate. For the thumb, we took eight pictures, six of them with a white background and two of them with colored backgrounds. The two colored backgrounds were blue and violet. Table 2 summarizes the fingerprint capturing environment parameters and Fig. 7 clarifies the environment of our database collection.

Table 2 Environment of the used camera in database collection

Parameter	Value
Camera type and model	Canon EOS Rebel T6i (EOS 750D)
Camera Sensor size	22.3mm x 14.9mm
Camera Viewfinder	Optical /LCD
Image file size	27+- 3 MB
Image format	CR2 (Canon Raw Version 2)
Flash	Off
Background	Use colored wood panels, one is white and two nonwhite panels
Distance between the finger and camera	6 – 8 cm
Light conditions	Uniform outdoor environment.
Image resolution	24.20 megapixels

The images are stored in CR2 format. We use a file naming convention that encodes the number and gender of the volunteer and the name of the finger. Figure 8 shows file naming convention in details. For example, the file (P016_MTh8.CR2) represents the image of the thumb with colored background for the 16th male volunteer. Figure 9 below shows some examples of the annotated dataset.

5.1 Difficulties faced while collecting data

The database collection process lasted for three months (from December 2015 until March 2016). During this period, many challenges were faced. Most notably are the following:

- The process of capturing clear images of fingerprints required many trials because of involuntary movement of the finger.

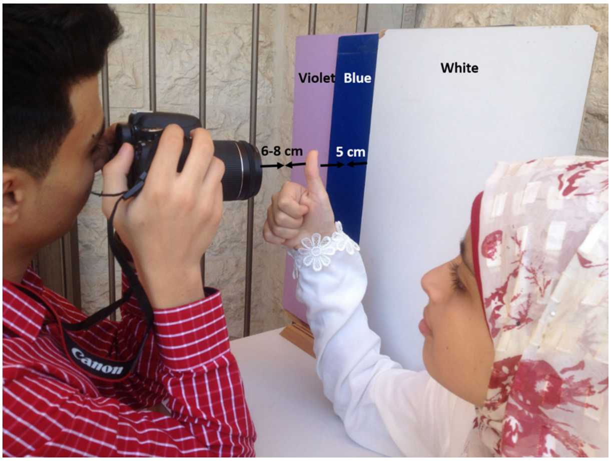


Fig. 7 Database collection environment

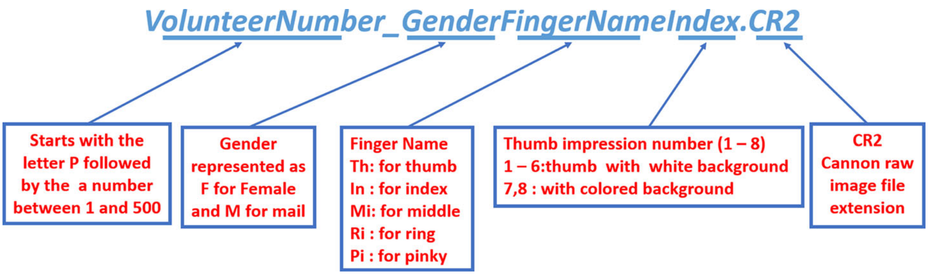
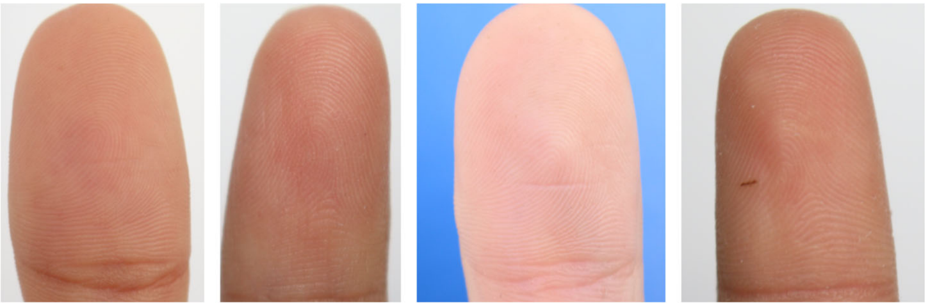

VolunteerNumber_GenderFingerNameIndex.CR2
Starts with the letter P followed by the a number between 1 and 500
Gender represented as F for Female and M for mail
Finger Name
Th: for thumb
In : for index
Mi: for middle
Ri : for ring
Pi : for pinky
Thumb impression number (1 – 8)
1 – 6:thumb with white background
7,8 : with colored background
CR2
Cannon raw image file extension

Fig. 8 File naming convention

- Dealing with uncompressed images requires large storage capacity, where one image size is greater than 27 MB.
- Adjusting the lighting and distance requires considerable effort and some experience.
- There are people who have soft fingerprints, which makes it difficult for us to capture a detailed image of their fingerprints. This forced us to re-capture some images several times.
- Because most of volunteers are students, their fingers are sometimes dirty and have ink stains. This resulted in poor fingerprint image quality. We used wet wipes to clean their fingers before image acquisition.
- The collection of the data took place in a county in the Middle East and North Africa (MENA) region. Unfortunately, this region has witnessed several notorious police states, which created fear in minds of the public about security-related issues. Collecting fingerprints is something that would always raise serious suspicions. We had to explain thorough our project and its scholarly value. We also presented the subjects with firm assurances that there is no hidden agenda or malicious intent behind this process. Some were convinced with our arguments while others simply refused to participate.



(a) P178_FTh1.CR2: A Female Thumb with white background

(b) P101_FRi.CR2: A Female Ring Finger with white background

(c) P171_FTh8.CR2: A Female Thumb with colored background

(d) P101_FIn.CR2: A Female Index fingr with white background

Fig. 9 Examples of the annotated dataset

6 Our fingerprint recognition system

To study the impact of image compression on fingerprint recognition, we must have a fingerprint recognition system in hand to perform some experiments. In this section, we describe the fingerprint recognition system that we put together for this purpose.

A system that identifies people through fingerprint, must perform several successive steps that are mainly (1) converting the image jpeg, (2) convert the image to grayscale, (3) segment the fingerprint from the image, (4) converted the finger image into binary image, (5) extract features from the finger image, and (6) compare features of the image with other images in the system to find a match.

Any automated identification system is designed in two main phases, training and testing. In the training phase is about processing a database of fingerprint images and represent them with some useful features. This phase passes through several steps which are (1) database collection, (2) image preprocessing, (3) feature extraction, and (3) storage of these features The second phase is called testing, where a query fingerprint image (the testing image) is put through the same preprocessing and feature extraction steps of the training phase. Then the query image is compared with all the database images (training images). If a query image matches one of the database images with a match that is above some threshold, the result query image is considered a match. Otherwise, the system will reject the test image. This step is called identification/recognition.

Fingerprint matching in our framework is not carried out using direct image-to-image comparison since this will need long time and large memory size [25]. Furthermore, dirt or distortion in images might lead to a mismatch between two images of the same fingerprint. Hence, this method is not practical; instead, the matching stage in most fingerprint systems is based on certain marks and features in the fingerprint [46].

If there are two fingerprints with the same terminations and bifurcations, so that, they form the same shape and the same dimensions, then, there is a high possibility that they belong to the same person. Recognition systems do not need the entire minutia recorded for both persons, but a certain number of detail is enough to be compared. This number of details varies depending on the fingerprint identification system and the quality of the original image. In the following sections, we illustrate in details our methodology starting from converting the captured raw files to JPEG images with different QFs and ending with determining the best QF that produces the highest accuracy. Figure 10 shows the framework stages, which will be explained in details next in this section.

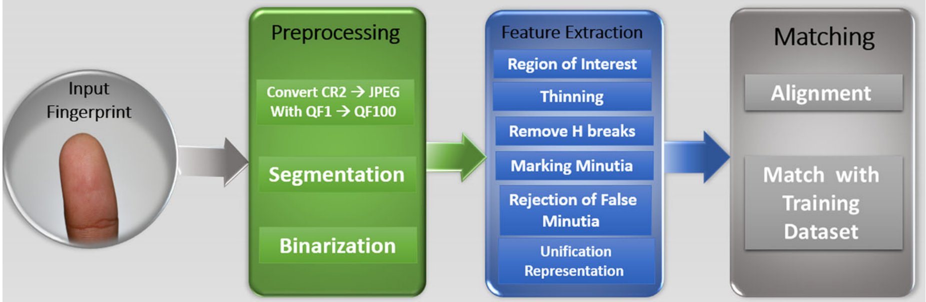


Fig. 10 Our framework phases

6.1 Building the database

For an individual to be recognized by an automated fingerprint recognition system, he should be first registered to the systems. Figure 11 illustrates the registration of an individual in the proposed system. The following subsections details the registration process.

6.1.1 Image conversion and compression

At the registration stage, the fingerprint images are captured by a Canon camera and saved without any compression as raw images. Particularly, in a the used Canon camera version, raw images are saved with CR2 extension. CR2 is also called digital negative. Specialist photographers prefer to use this type of file, because it retains many details of the image. However, it is not widely used because the image size is very large (in our database, a single image reaches up to 30 MB in size). This makes it difficult for such images to be stored and uploaded to the Internet. The next step is to convert the CR2 images to JPEG using different QF ranging from 1 to 100. JPEG stands for Joint Photographic Experts Group, which is usually used as a name of the a standard for still image compression algorithm. It compresses the image to ensure the preservation of quality and reduce the storage size of the image. It is widely used in many imaging-related applications, where smaller sizes help in storing, transferring and uploading the images on the Internet [21].

To conduct this conversion, we used a special library for this purpose called FreeImage. FreeImage is an open source graphics library, written in C language and used in image processing; it is compatible with Linux, OSX and Windows. It is widely used due to its speed and simplicity. Floris van den Berg created it and it has been under development for more than 9 years. It is worth mentioning that this library is free and not included within the automatically installed libraries in different programming languages or instructional machines.

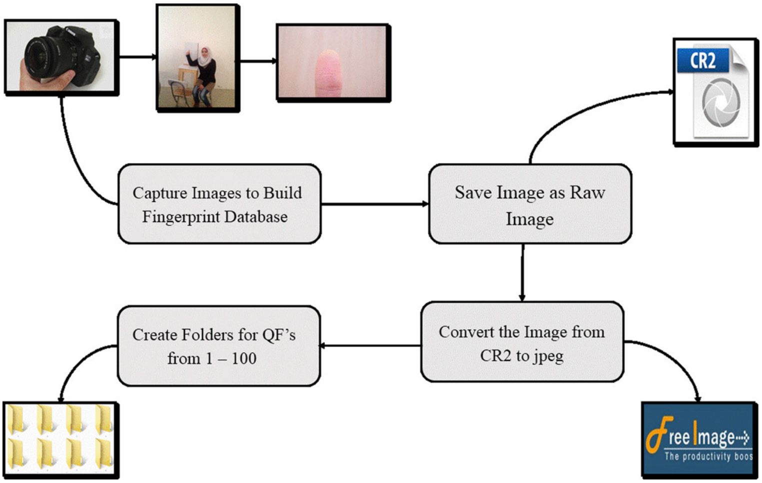

CR2
Capture Images to Build Fingerprint Database
Save Image as Raw Image
Create Folders for QF's from 1 – 100
Convert the Image from CR2 to jpeg
Free Image
The productivity boos

Fig. 11 Illustration of the templates registration

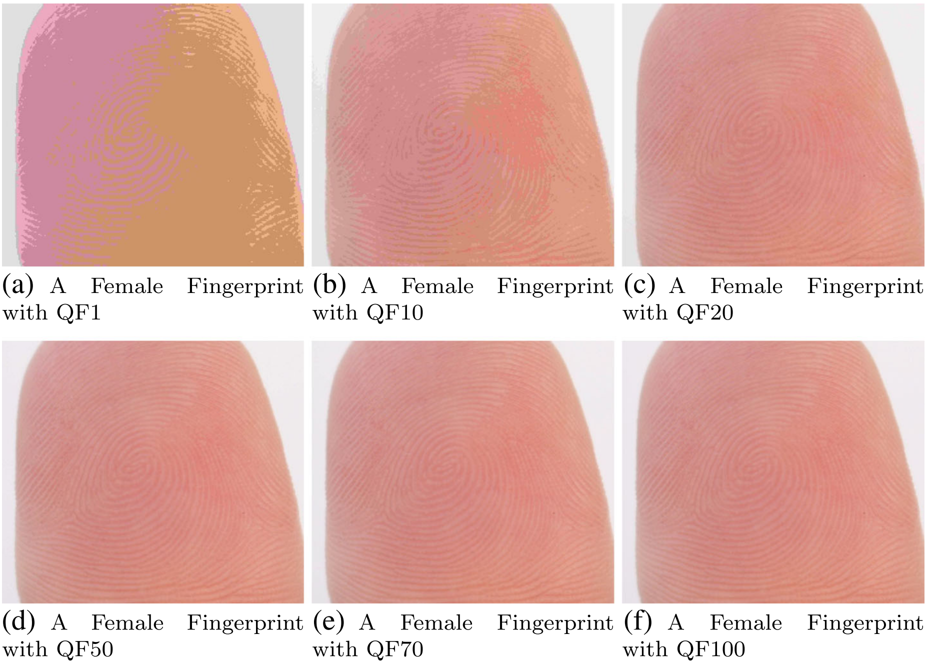

(a) A Female Fingerprint with QF1
(b) A Female Fingerprint with QF10
(c) A Female Fingerprint with QF20
(d) A Female Fingerprint with QF50
(e) A Female Fingerprint with QF70
(f) A Female Fingerprint with QF100

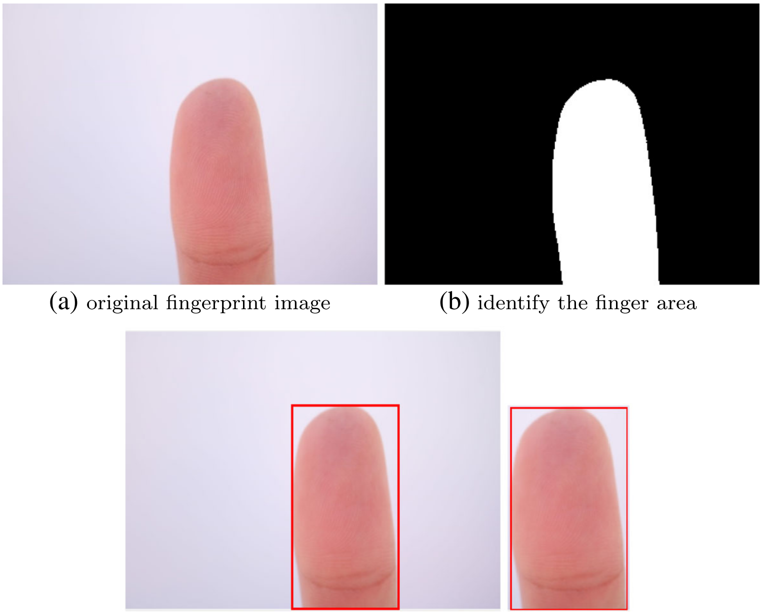
Fig. 12 Fingerprint images after conversion in different qualities

Therefore, it needs to be added to our source code, to make sure that it is compatible with our system.[5] Examples of converted images in different QFs are shown in Fig. 12.

6.1.2 Segmentation

After database collection, each image must be segmented into two regions: foreground (finger region) and background. This step is called segmentation. There are several methods for segmentation. One method separates the object from the background using the color information, was first proposed by Hassanat et al. in 2015 [15]. This method is simple, efficient and deals with digital color images and thus we rely on it for our segmentation phase. The method utilizes a novel fusion of color space models based on neural networks to separate the foreground from the background. In the training step, users click ten times on the foreground (finger region), and another ten clicks on the background. The system stores the pixels that are clicked on and other eight neighboring pixels. After that, it saves these pixels to be the input layer of the neural network. Consequently, the neural network learns how to identify the ROI from the background. This way, we identify the finger area, and then store the new image, which contains only the finger area. Figure 13 show a sample output of this stage.

[5] http://api.256file.com/freeimage.dll/en-download-77473.html



(a) original fingerprint image

(b) identify the finger area

(c) segmented image

(d) new image after segmentation

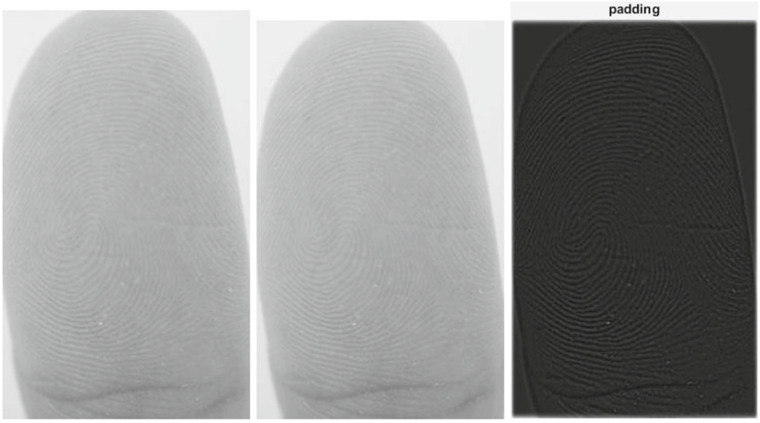
Fig. 13 Image segmentation result

6.1.3 Binarization

After taking the segmented images, we must recognize the minutia (ridges and valleys), by converting the image into two colors, white and black. This step is called binarization. One of the important methods for binarization of colored images is Khalil's method [26, 28]. This algorithm performs image enhancement as follows. First, it converts the colored image to grayscale image to reduce the problems of different illuminations in the image. This problem results in non-uniform components in the image. After converting the image to grayscale as shown in Fig. 14a, a simple normalized image is required as shown in Fig. 14b. It utilized the algorithm proposed by Hiew et al. [16]. The normalization is a pixel-wise procedure. According to Khalil's method, the normalization is used to reduce the illumination problems through calculating the variance and mean as shown in (1). It is worth noting that the normalization step does not affect the structure of the minutiae whether it is ridge or valley.

$$G(x, y) = \frac{f(x, y) - m_f(x, y)}{\sigma_f(x, y)} \tag{1}$$

where, $f(x, y)$ expresses the grayscale fingerprint image, $m_f(x, y)$ is the mean of f and $\sigma_f(x, y)$ expresses the standard deviation of the segmented image. The spatial smoothing is used for evaluating m_f and σ_f. After that, two main steps are performed. The first is Discrete Fourier transform (DFT)-based fingerprint ridge analysis, where the ridge analysis method is utilized to detect the ridge area. The ridge information is extracted by performing



(a) image converted into grayscale format (b) normalized image (c) padded image

Fig. 14 Binarization stages

a high-pass filter based on the DFT on input images. The second step is the Short Time Fourier Transform (STFT) method, which is used for image enhancement. The first step consists of four sub-steps as follows. At first, a Gaussian high-pass filter $G(u, v)$ is applied as a scalar product of the matrices on the input normalized images. It was created through a padding size (P x Q) as shown in Fig. 14c, and filter width of Fourier transform according to the following equations:

$$D_0 = 0.05 \times P \tag{2}$$

$$D(u, v) = \sqrt{(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2)} \tag{3}$$

$$G(u, v) = 1 - e^{\frac{-D^2(u,v)}{2D_0^2}} \tag{4}$$

where, P and Q: padding size block. D_0: is the cut-off frequency, $D(u, v)$ is the distance between coordinates (u, v) and the center of frequency at $(0, 0)$.

In these equations, the size of input image is used to determine the padded size as $2 \times M \times N$, where M and N is the size of input image. The appropriate width of the filter generated during the experimental results in [10] which equals 5% of the padding size, P. The Fourier domain is used to transform the input image $F(X, Y)$ into the frequency domain (DFT domain) (see (6)). Results from the previous steps are reversed into the spatial domain using the matrix $H(u, v)$ (see (7)).

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})} \tag{5}$$

$$H(u, v) = F(u, v)G(u, v) \tag{6}$$

$$h'(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} H(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \tag{7}$$

where M and N is the original image dimension.

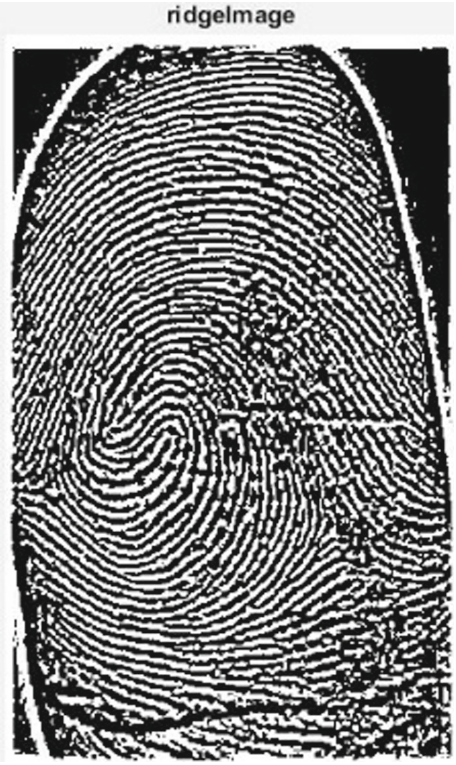


Fig. 15 Ridge image resulted from Algorithm 1

In the fourth and final phase of this step, the image is cropped to dispose of the previous padding. Then a simple threshold is used to extract the ridge information as defined in (8) and (9). Algorithm 1 illustrates the entire algorithm.

$$I_{ridge} = \{I_{ridge}(i, j) \in h' | i = 1, 2, ..., I_{h'}, j = 1, 2, ..., I_W\} \tag{8}$$

$$I_{ridge} = \begin{cases} 1, & I_{ridge}(i, j) > 0.005 \times \max(I_{ridge}) \\ 0, & otherwize \end{cases} \tag{9}$$

Algorithm 1 Overview of ridge analysis using DFT [Khalil]

```
Input: Digital image containing human fingerprint (from segmentation step)
Output: Binary image
Step 1: Read the segmented image file from segmentation step.
Step 2: Normalize the image. (Equation2)
Step 3: Calculate a padding size (P x Q).
Step 4: Generate filter width for the Fourier transform. (Equation 5)
Step 5: Perform a high-pass filter based on DFT. (Equation 6)
Step 6: Transform the input image F(x, y) into the wavelet domain by Fourier domain.
    (Equation 5)
Step 7: Invert back into the spatial domain using the matrix H (u, v). (Equation 7)
Step 8: Crop image to dispose the previous padding. (Equation 8)
Step 9: Use a simple threshold to extract the ridge information. (Equation 9)
```

According to Khalil's method, the first step is to identify the ridge area (as shown in Fig. 15, whereas the second step is to enhance the image obtained from the ridge analysis step. The step of enhancing the image is required to obtain highly accurate results. For this reason, the Short Time Fourier Transform (STFT) analysis is used to improve the image. At first, the image that contains information of the ridge is divided into overlapping windows. The purpose of this overlap between windows is to preserve the coherence between the lines of the ridges and to get rid of the cluster impact that naturally happens because of applying the window-by-window method. However, there is a procedure preceding the application of the STFT analysis, which is the Fourier spectrum of the block $F(r, \theta)$, where θ is the orientation. is a random value. r is ridge frequency, r is presumed to consist of random values. Then, the (ROI) is identified by calculating two functions: the marginal density function $p(\theta)$, $p(r)$ and the probability density function $p(r, \theta)$.

$$P(r,\theta) = \frac{\mid F(r,\theta) \mid^2}{\int_r \int_\theta \mid F(r,\theta) \mid^2} \tag{10}$$

$$P(r) = \int_\theta P(r,\theta) d\theta \tag{11}$$

$$P(\theta) = \int_\theta P(r,\theta) dr \tag{12}$$

After applying the STFT analysis in small windows one by one, the results from this procedure are a Ridge Orientation Image, Energy Image (EI), and Ridge Frequency Image (RFI). The next step is smoothing the orientation image to get the coherence image based on the vector averaging as in (10) where θ is the orientation. Moreover, r denotes ridge frequency. Coherence image is responsible for the elimination of false spots, which appear because of the ridge flow ends at the small window boundary. As shown in the following equations.:

$$E(\theta) = 0.5tan^{-1}\left(\frac{\int_\theta P(r,\theta)d\theta}{\int_r P(r,\theta)dr}\right) \tag{13}$$

$$E(r) = \int \int_r P(r).rdr \tag{14}$$

To get the region mask, the common thresholding Otsu's method is applied.

$$E(x, y) = \log\{\int_r \int_\theta \mid F(r,\theta) \mid^2\} \tag{15}$$

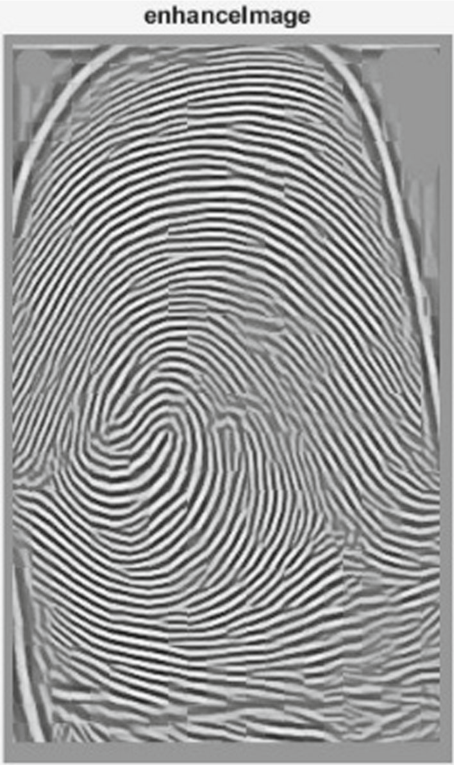
The ridge orientation, the coherence and the frequency images are used to generate two filters; angular filter and radial filter, see (16) and (17). A filter window of 16 × 16 is used to smooth the image. Eventually, the image that the filters have been applied on, will be rebuilt with a new enhanced image as shown in Fig. 16.

$$H(r) = \sqrt{\frac{(rrBW)^{2n}}{(rrBW)^{2n} + (r^2 + r^2bw)^{2n}}} \tag{16}$$

$$H(\Phi) = \begin{cases} \frac{\cos^2 \pi \mid \Phi - \Phi c \mid}{2\Phi BW}, & \mid \Phi - \Phi c \mid \leq \Phi BW \\ 0, & otherwize \end{cases} \tag{17}$$

where, rBW is the radial bandwidth, ΦBW is the angular bandwidth, and ΦC is the mean orientation. Algorithm 2 summarizes the steps of converting ridge image to enhanced image:

Fig. 16 Enhanced Image Resulted From Algorithm 2



Algorithm 2 Converting Ridge Image to Enhanced Image

```
Input: Ridge Analysis image
Output: Mask Region
Read a ridge analysis image
Split an image into overlapping blocks each one B(X, Y).
Use B=B-average (B) to eliminate the DC component of B.
Perform multiply by spectral block w
Apply the FFT of the block F=FFT(B)
Apply root filtering on F.
Apply STFT analysis.
Smooth the orientation image O(X, Y).
Generate region mask R(X, Y).
The output is the enhanced as enhancement image.
```

6.1.4 Fingerprint feature extraction technique

After converting the image into a binary image, the next step is window direction estimation, followed by implementing some morphological operations. The previous two steps are to extract the (ROI) [27]. For estimating the window direction, the image is divided into blocks, where the size of a block is WxW. According to [3, 27], the best values of W is 16. To carry out these steps, in each window in an image, the gradient values are calculated on the

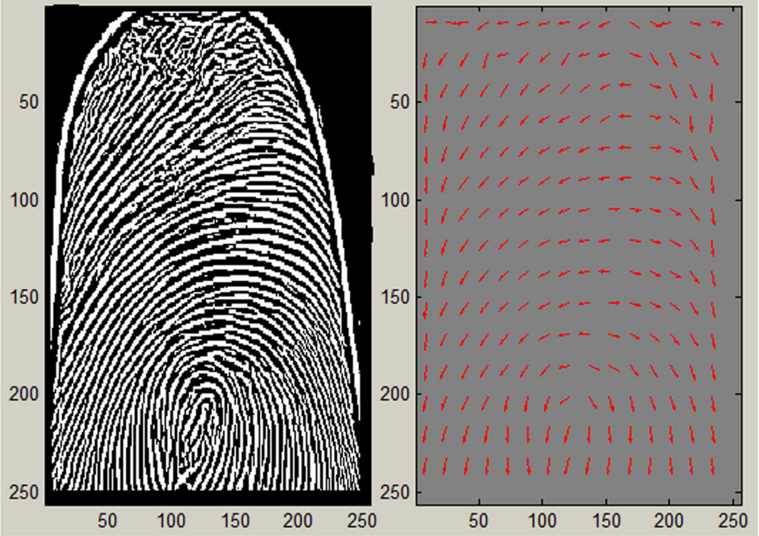


Fig. 17 A binerized fingerprint (left) along with its direction map (right)

X-axis and the Y-axis by using two Sobel filters for all pixels. After that, the value of the least square approximation is calculated based on the following equation:

$$tan2\beta = \frac{2\sum\sum(g_x \times g_y)}{\sum\sum(g_x^2 - g_y^2)} \tag{18}$$

where, g_x and g_y are the gradients taken along x-axis and y-axis, respectively.

The orientation of ridges is estimated according to the equation above. The equation is easy to interpret where the gradient values g_x and g_y represent the cosine and sine values, respectively. So, the tangent value of the window direction is estimated using the following formula:

$$tan2\theta = \frac{2sin\theta cos\theta}{cos^2\theta - sin^2\theta} \tag{19}$$

After estimating each window, any window that does not carry any information around the edges and grooves is rejected based on the following formula:

$$E = \frac{2\sum\sum(g_x \times g_y) + \sum\sum(g_x^2 - g_y^2)}{W \times W \times \sum\sum(g_x^2 - g_y^2)} \tag{20}$$

The output of these steps is called the direction map that is shown in Fig. 17.

The next step is to perform some morphological operations. These operations are of the most common operations in image processing. They are mainly two operators; Open and Close. Both of these operators depend on the sequence of two operations. The first is called erosion which is an operation used to shrink or thin objects in binary images. The second is called dilation, which is an operation used to grow or thicken objects in binary images. The opening operation uses erosion then dilation, but the closing operation uses dilation then erosion. The fingerprint area is obtained in addition to its boundaries, which are obtained by the subtraction of the two previous operators: open and close. Figure 18 shows the output of this step.

The next step is **thinning**. This step is used to remove small parts of the pixels. These redundant pixels of ridges are eliminated by dividing the image into windows with $M \times M$ matrix, where M is dimension of matrix $M = 3$ [13]. This operation plays an important role to get the best accuracy [14]. In each window, the sliding neighborhood algorithm used to calculate the centroid. The centroid pixel location is changed for each window. After finding the centroid, the summation is performed for each pixel of the thinned matrix to obtain a new matrix consisting of minutiae points. Then subtracts 1 from the computed sum, which gives the value of the pixel. After this step, delete the redundant and singular pixels that may arise after the previous step. These conditions can be summarized as follows:

1. Divide the image into 3 × 3 blocks.
2. Thin each line of ridge to its center pixel.
3. Remove redundant and singular pixels.

Many algorithms can be used for this task. According to the study presented recently in 2015 by Hannah et al. [13], Zhang-Suen's Algorithm and Hall's algorithm are two of the most common techniques. According to [13], the former algorithm is more accurate and is more efficient than the latter. This is why we use Zhang-Suen's algorithm in this work. Algorithm 3 presents the basic steps of the thinning algorithm. Any pixel that lies on the edge of the foreground regions is converted to its neighboring background. Also, any pixel that has more than one foreground neighbor is removed along the ridge line. This procedure repeated until we reach convergence.

Algorithm 3 Thinning Algorithm

```
Input: Length (L) and width (W) of an image.
Output: Image after thinning.
L = length of image
W = width of image
for X =1 : L do
    for Y =1 : W do
        Pixels p (X, Y)
        if ($2 \leq B(p) \leq 6$) AND (A(p) == 1) then
            Perform one of the following condition
            a) (x - 1) × (x , y + 1) × (x + 1, y) = 0 in odd iterations
            b) (x-1, y) × (x, y+1) × (x , y - 1) = 0 in even iterations
            Perform one of the following
            a) (x , y + 1) × (x + 1, y )× (x , y - 1) = 0 in odd iterations
            b) (x - 1, y ) × (x + 1, y ) × (x , y - 1) = 0 in even iterations
        else
            Delete p(X , Y )
        end if
    end for
end for
```

The next step is **removing the H breaks and spikes**. This step is considered as another morphological operation to eliminate some isolated points and spikes, which are usually called the H breaks, where any single-pointed ridge or single-pointed break in a ridge is considered as noise and should be removed. This step can be coded in MATLAB using the following commands;

bwmorph (binaryImage, ‘hbreak’, k)
bwmorph(binaryImage, ‘clean’, k)
bwmorph(binaryImage, ‘spur’, k)
Where bwmorph applies a specific morphological operation to the binary image, binaryImage is the image name, ‘clean’ is th option to remove isolated pixels (individual 1’s that are surrounded by 0’s), such as the center pixel in this pattern, ‘spur’ to Removes spur pixels. For example:
0 0 0 0
0 0 1 0
0 1 0 0
1 1 0 0
becomes
0 0 0 0
0 0 0 0
0 1 0 0
1 1 0 0
‘hbreak’ is used to unconnected H-connected pixels. For example:
1 1 1
0 1 0
1 1 1
becomes
1 1 1
0 0 0
1 1 1

The next step is **marking minutiae points**. The output image from the previous step is used to determine the minutiae. Simply a thinned image is divided into windows of size 3 × 3. If the value of the central pixel equals to 1 and surrounded by at least three neighbors with value 1, then it is considered as bifurcation. If the value of the central pixel is equal to 1 and surrounded by only one neighbor with value 1, it is considered as termination. This technique is denoted by Crossing Number (CN), and it is applied commonly to extract the minutiae [53, 54].

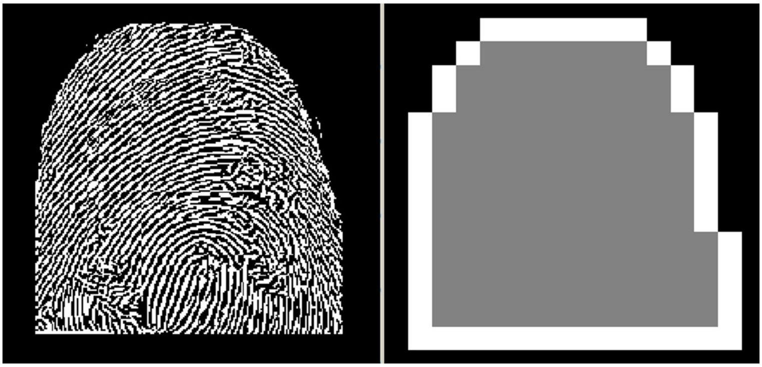
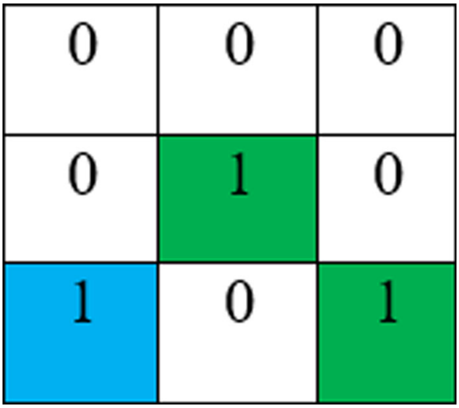


Fig. 18 Region of Interest and its boundaries



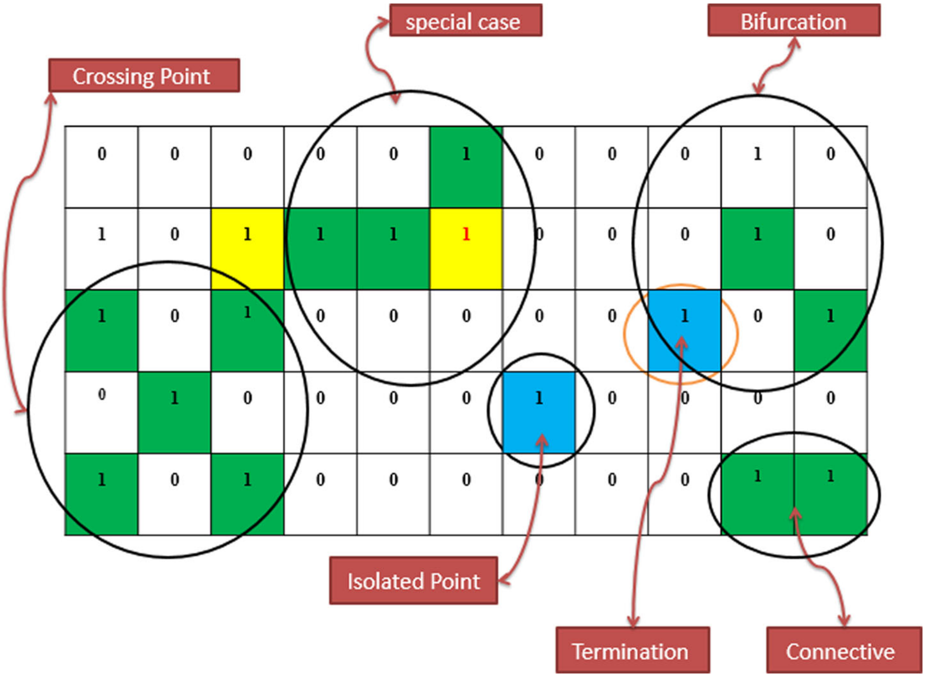
0	0	0
0	1	0
1	0	1

Fig. 19 A block of an image (3 × 3 window)

The CN is calculated according to following equation:

$$CN(Pxl) = \frac{1}{2\sum_{i=1}^{8} \mid P_i - P_{i+1} \mid} \tag{21}$$

Where, Pxl is the pixel value (zero or one).
If number of 1s = 0 then, isolate the point,
If number of 1s = 1, then it is a termination point,
If number of 1s = 2, then it is a connective point,
If number of 1s = 3 then it is a bifurcation,



0	0	0	0	0	1	0	0	0	1	0
1	0	1	1	1	1	0	0	0	1	0
1	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	1	1

Fig. 20 Different pixel types

If number of 1s = 4 then it is a crossing point.
See Fig. 20.

For example, Fig. 19 defines a block from an image. According to the previous equation, $CN = 1/2 \times (1 + 1 + 1) = 1.5$. Then, this block is defined as bifurcation.

Figure 20 illustrates a special case that a genuine branch is triple counted. From Fig. 20, the ridgelines area is represented by ones and the background area is represented by zeros. We color pixels according to their types where green indicates bifurcation and blue indicates termination. The yellow pixels represent a special case called triple counting and is considered as a bifurcation.

A post step is performed to reduce the number of false minutiae. False minutiae may be generated in the image due to factors such as noisy images, and image artifacts produced by the thinning process. Hence, after the minutiae is extracted, it is necessary to perform a post step in order to validate the minutiae.

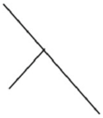
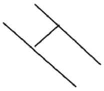
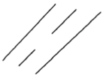
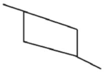
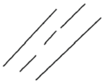
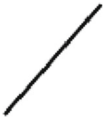
The impact of false minutiae is significant on the accuracy of the system. In our work, we focus on eliminating any false minutia by calculating the average distance between two neighbor ridges.

Table 3 shows the various types of false minutiae.

There are three important rules to remove false minutia:

1. If the value of the distance between bifurcation and termination is less than D and the two minutia are on the same ridge as shown in Minutiae 1 in Table 3, then reject minutia.
2. If the value of distance between two bifurcations is less than D and the two minutia are in the same ridge as shown in Minutiae 2 and Minutiae 3 of Table 3, then reject minutia.
3. If two terminations are equal to distance D and their directions are coincident with a small angle variation, and no other termination is located between the two terminations as shown in Minutiae 4, Minutiae 5 and Minutiae 6 of Table 3, then reject minutia.

Table 3 Various types of false minutiae

Number	Figure	Reason to be Refused	Number	Figure	Reason to be Refused
1		A spike piercing into a valley	2		A spike falsely connecting two ridges
3		Two near endings	4		Two near bifurcations located in the same ridge
5		Similar to number 3 with extra ridge	6		One short ridge

It is worth mentioning that the removal of the false minutia has important positive implications on the accuracy of the matching. Moreover, the reduction of the number of minutiae reduces the time complexity.

The next step is the **unification of minutiae representations**. Because of the diversity of the fingerprint image capturing conditions, the fingerprints should be captured in a standard way to be able to identify the minutiae features. These standards are related to x-coordinate, y-coordinate, the orientation of the associated ridges. Regarding bifurcation, the calculation of the orientation needs special treatment, where the bifurcation point contains three ridges and their own x-y coordinates. To resolve this issue, we use the method of Shobhraj et al. [51]. According to Shobhraj method, the orientation between two terminations is implemented using the following procedure:

1. Track a ridge segment starting from the point that termination and length is D.
2. Sum up all the x-coordinates (s_x) of points on this ridge segment.
3. Get s_x then divide the above summation with D and determine s_y using the same technique [42].
$s_x = \frac{\sum x-coordinatesofpointsintheridgesegment}{D}$
$s_y = \frac{\sum y-coordinatesofpointsintheridgesegment}{D}$
4. Finally, determine the direction using Equation 22:

$$Tan^{-}1\left(\frac{sy-ty}{sx-tx}\right) \tag{22}$$

6.2 Fingerprint identification

After th registration, the fingerprint features are stored in the form of a template, with (.dat) extension, in the database. After that, the system should be able to identify the registered fingerprint. This stage is called fingerprint identification/recognition. The identification process depends on the minutiae features matching. After extracting and saving the minutia from the query fingerprint and stored as vector, this vector is matched against the stored minutiae victores in the database. The similarities between the minutiae are calculated using the minutiae match algorithm. Based on this, it is determined whether two images are for the same person or not. In this work, we use the alignment-based algorithm proposed in [32]. This algorithm goes through two stages: the alignment stage, where the two minutiae are aligned, and the match stage, where the similarity score of the two minutiae is computed.

The first stage is the **alignment stage**. When two fingerprint images are compared, the selection of the minutia of each image is selected, and the similarity of the two ridges related to two reference points of the minutiae is calculated. The algorithm computes all similarity scores and chooses the highest similarity for the match.

Let us assume that ridge is related to each minutia in the form of a series of x-coordinates $(X1, X2...XN)$ of points on the ridge. For further illustration, we show an example. Consider that I1 and I2 are the two minutiae obtained by the following (23) and (24):

$$I_1 = m_1, m_2, ..., m_M \tag{23}$$

$$I_2 = m'_1, m'_2, ..., m'_M \tag{24}$$

where $m_i = (x_i, y_i, \theta_i)$. A sample point is taken along the ridge (L) where L defines the average inter-ridge length. In order to obtain the similarity score between the minutiaes, we use the following equation.

$$S = \frac{\sum_{i=1}^{m} xiXi}{\sqrt{\sum_{I=\theta}^{m} xi^2 Xi^2}} \tag{25}$$

where $(xi \; xn)$, $(Xi \; XN)$ are the x coordinates for each minutia and m is minimum of the n and N value. If the similarity score is greater than the threshold value, then execute next step, else, compare the next new two ridges.

The second stage is **Match Stage**. If we assume that almost all minutiae are found in the same position and direction, then the minutiae can be compared to obtain similarity scores. However, the minutiae are exposed to some problems, where it is impossible to have the same minutiae at the exact position and direction. This problem is solved by surrounding each minutia pattern with a bounding box. Thus, the comparison will no longer be between the exact minutiae location and direction, but rather between the location and orientation within the surrounding boxes.

7 Experiments and results

7.1 Evaluation methodology

We have selected a random sample of 100 volunteers (50 males and 50 females; aged between 15 and 18 years) and captured their fingerprint images. After that, we converted each raw image to a compressed image using JPEG compression with different QFs ranging from 1 to 100. This resulted in 10,000 images (100 images per volunteer).

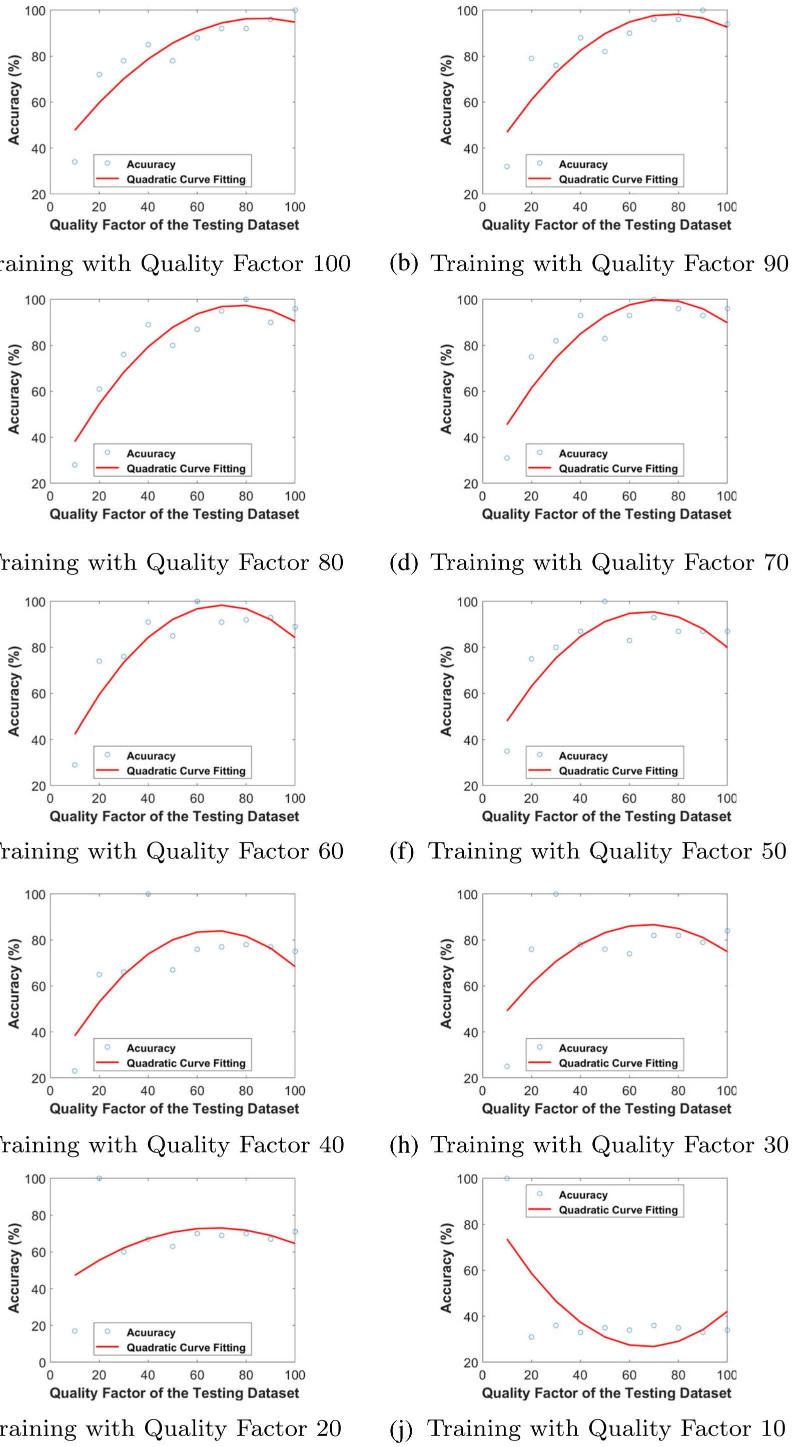
The first experiment was conducted with the hundred random volunteers on QFs 10, 20, 30, ..., 100. The main objective of this experiment is to determine the best tradeoff between compression and fingerprint recognition accuracy.

While capturing images of the sample, the disparity in images clarity, regarding to volunteer's gender sparked my attention. As details of the male fingerprint was clearer than females, this observation led me to conduct my experiments on separate male and female samples, and record the results. This may help to make a decision in this regard.

One criterion related to the reliability of the matching process is related to the difference between the highest similarity value between the query image and the database images and the second highest similarity value. So, we study the results of this criterion. If the differences are little, then the results are not reliable.

Another criterion to judge the effectiveness of the used system is how it handles unknown fingerprints. A good system should produce very low similarity scores in such cases. So say that a recognition system is authorized, and can be used effectively, you must identify the fingerprint even changed his image. System users need match differently every day with original images stored within the system. To check the system, we take a sample of 50 images on behalf of session 1 and 50 other images of the same people but in a different session. After that, we measure the system's accuracy using images from different sessions.

Our experiments involve processing the obtained images in different ways, each with its own software requirement. For example, we compress the images using different QFs. We use C# code for this task. We also segment the compressed images to get rid of background. We use MATLAB code to conduct this task. A binarzation step is applied using MATLAB



(a) Training with Quality Factor 100 (b) Training with Quality Factor 90

(c) Training with Quality Factor 80 (d) Training with Quality Factor 70

(e) Training with Quality Factor 60 (f) Training with Quality Factor 50

(g) Training with Quality Factor 40 (h) Training with Quality Factor 30

(i) Training with Quality Factor 20 (j) Training with Quality Factor 10

Fig. 21 Fingerprint recognition accuracy when using training sets of different quality factors

code. The matching algorithm, which includes feature extraction as well as similarity score computation, is also conducted with MATLAB.

7.2 Impact of using different training and testing QFs

In the first set of experiments, we aim at measuring the impact of using different QFs for the training as well as the testing sets. For the training dataset, we randomly choose 100 volunteers and take their fingerprint images with QFs ranging from QF10 to QF100.

For the results shown in Fig. 21, the experiments were conducted using 100 images in different QFs ranging from 10 to 100. Each figure represents the results of the experiments on a different quality factor specified in the training dataset. In these figures, we note that the accuracy of the fingerprint recognition system increased proportionally with the quality factor. In Fig. 21j, the experiment was conducted regarding to QF100 in training dataset. Here we note that the framework achieved satisfying accuracy for testing QFs of 30 and above. For training QFs of 50 and above, the achieved accuracy between 80% and 100% for testing QFs of 40 and above. However, when as we decrease the training QFs from 30 to 20 to 10, the accuracy range gradually decreases from 70–80% to 60–70% to 30–40%. I think that these ranges are not generally acceptable for fingerprint recognition systems.

7.3 Using low testing QFs

The previous set of experiments revealed that while moderate and high testing QFs resulted in high recognition accuracy, low testing QFs (QF30 and below) presented more interesting trends. So, it is worth it to study such QFs in more detail especially since it is reasonable to assume that training fingerprint images might be captured and stored with high QF whereas

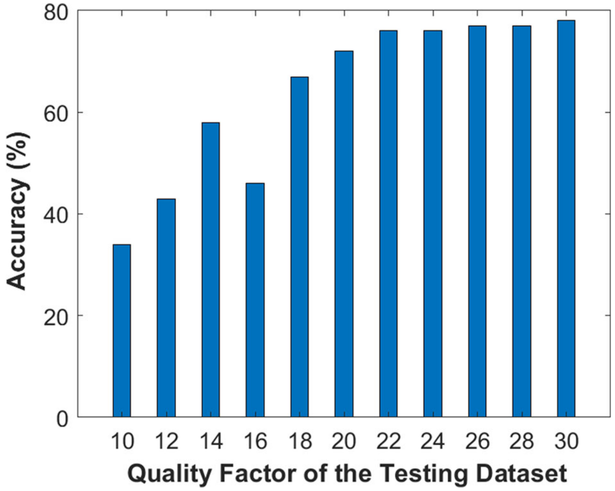


Fig. 22 Fingerprint recognition accuracy for low testing QFs

testing (query) fingerprint images might be captured and stored with low QF. In this experiment set, the training dataset contained fingerprint images of 100 volunteers with QF100. As for the testing images, they have QFs ranging from 10 to 30.

Figure 22 shows the results of this experiment. The Figure shows that the accuracy increases as the testing QF goes from 10 to 22. However, for testing QFs between 22 and 30, the accuracy saturates on about 75%. This accuracy is acceptable for certain applications such as when using fingerprint recognition to record the attendance of students in classes or employees in an organization.

7.4 Considering gender

During the sample collection phase, it has come to our attention that there are differences between male and female fingerprints in terms of clarity. This motivated us to further explore this issue in a separate set of experiments in order to determine whether fingerprints of different genders are affected differently by the different QFs. To the best of our knowledge, this issue has not been addressed before.

For this experiment, we randomly choose a set of 100 volunteers (spread equally among the two genders). Figure 23 shows the results of the experiments. The figure shows that there is no difference in the system accuracy on testing QFs 60 and above. However, if we look at the results for testing QFs below that, we find that the accuracy for male fingerprints is significantly higher than that for female fingerprints. Most interestingly are the results for testing QF 10 where the accuracy for male fingerprints reached 50% while the accuracy for female fingerprints is only 10%. One justification for such disparity is the typical lifestyle of members of each gender in the region from which the volunteers hail. Specifically, we are talking about how women do the housework and are continuously exposed to erosion factors that affects the fingerprints negatively. This is a result of several causes including: washing pots, big chemical detergent exposure leading to the smoothness of their fingerprint. Unlike males who are characterized by performing tough jobs making their fingerprints clearer.

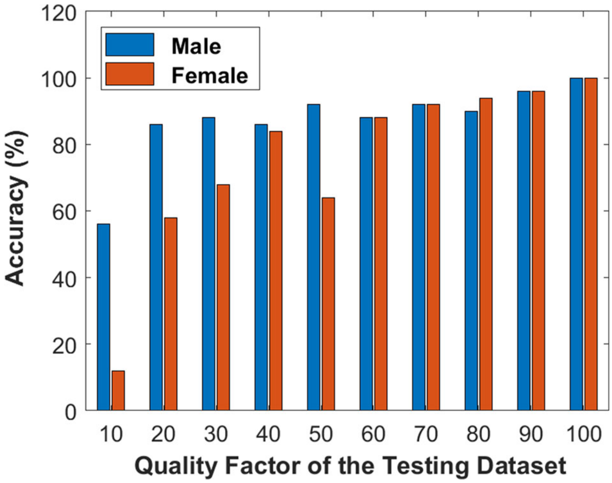


Fig. 23 Impact of Gender on QF

7.5 Difference between highest and second highest similarity scores

As mentioned earlier, the fingerprint recognition system works in a "nearest neighbor" manner in the sense that it calculates the similarity between the test image and all training images and then it chooses the training image that is closest to the test image. It would be interesting for such systems to study how reliable is the decision they make by considering the how close is the system to choose the second nearest neighbor. Specifically, consider the training image with the highest similarity (call it i1) and the training image with the second highest similarity (call it i2). If the similarity score of i1 is very close to that of i2, then the system cannot be considered reliable.

Figure 24 shows the results of this experiment. The figure shows that with higher QF the differences in similarity score between image with maximum similarity and image with second maximum similarity becomes higher, which means that the recognition system is more reliable. The figure also shows that the differences for male fingerprints are higher than those for female fingerprints, which supports the observation made in the previous experiment about the system higher effectiveness in matching male fingerprints compared with female fingerprints.

7.6 Using unseen testing images

In all of the previous experiments, the testing images exist in the training dataset, but with different QFs. However, in realistic fingerprint recognition scenarios, the query (testing) image might be for someone whose fingerprint images are in the training database, but the testing image itself does not appear in the training database. For this purpose, we consider a training dataset of 50 volunteers where each image is stored in three different QFs: QF30, QF50 and QF70. As for the testing images, they are for the same volunteers, but they are not the same as the training images (the training and testing images were each taken in a separate session). Also, for the testing images, we consider more QFs: QF10, QF20, QF100.

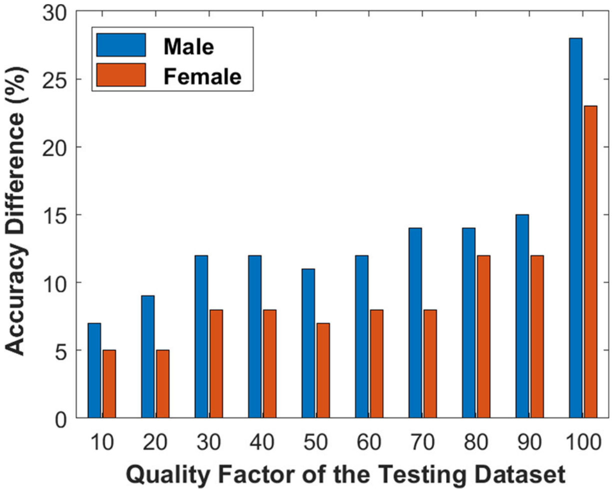


Fig. 24 Differences in similarity score between image with maximum similarity and image with second maximum similarity

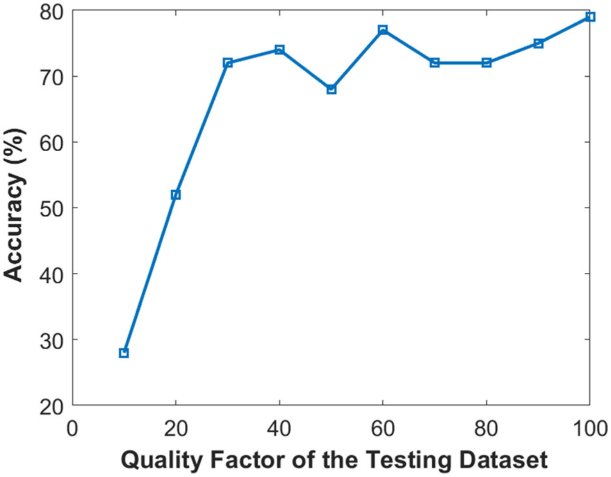


Fig. 25 Accuracy between two different sessions on QF30

Figures 25, 26 and 27 show the results of this experiment. The figures show the same trend in the sense that the accuracy is low for QF10 and, as the testing QF is increased, the accuracy exhibits sharp increase followed by a quick saturation after QF30. This conforms with the previous experiments.

7.7 Rejecting unauthorized people

In all of the previous experiments, the query (testing) image is for someone whose fingerprint images are already in the training database. So, we are actually testing the system's ability to correctly identify people already in the database. This means that system's output is a true positive or a false negative (known as Type II error). To use more technical terms,

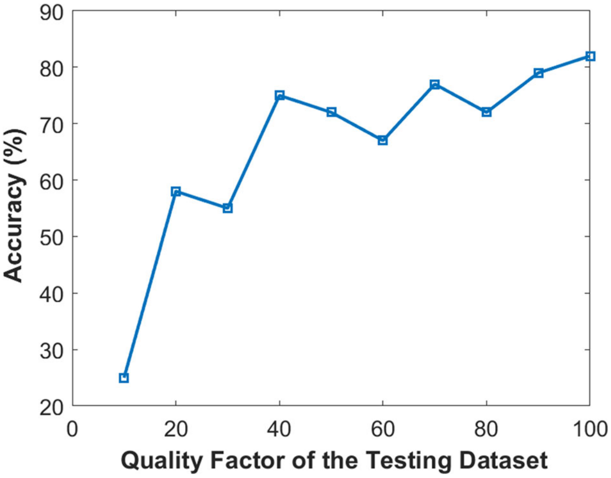


Fig. 26 Accuracy between two different sessions on QF50

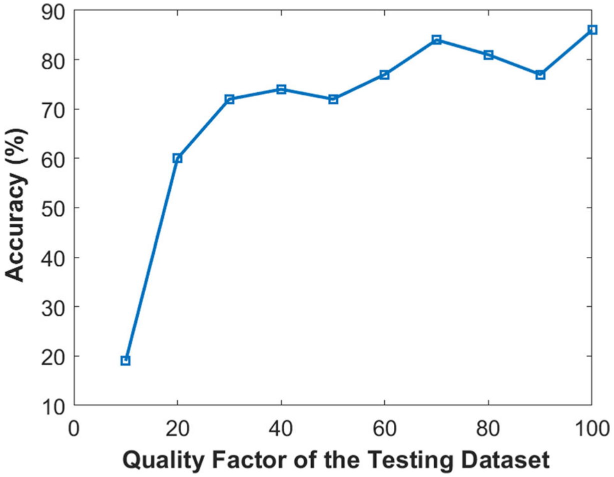


Fig. 27 Accuracy between two different sessions on QF70

the accuracy that we have been measuring so far is known as the sensitivity or recall of the system, which measures the percentage of true positives to the summation of true positives and false negatives. However, such errors are not the only errors faced in realistic scenarios. In a false positive (known as Type I error), the system mistakenly identifies a person as being in the database where in fact he/she is not. To account for such errors, we use a measure called specificity, which computes the percentage of true negatives to the summation of false positives and true negatives. Now, to use this measure, it does not make sense to keep using the “nearest neighbor” approach discussed in the previous section. Instead, we must introduce a minimum threshold on the similarity such that if the highest similarity score is below this threshold, then the testing image must be rejected (i.e., not matched with any training image).

For this experiment, we consider a training dataset of 50 volunteers where each volunteer has two images taken on two different sessions and each image is stored in three different QFs: QF30, QF50 and QF70. As for the testing images, they are for the six volunteers different in the session from the volunteers of the training dataset, but from same quality factor. Table 4 shows the similarity score for each of the six testing images. The results show that low QFs gives higher similarity scores, however, the similarity scores are still very low

Table 4 Similarities for 6-peoples out of the systems with 3 different QFs:

Quality Factor	P1	P2	P3	P4	P5	P6	Average
QF30	50%	47%	48%	49%	49%	49%	48.7%
QF50	46%	40%	43%	46%	43%	44%	43.9%
QF70	41%	41%	42%	41%	43%	41%	41.5%

to be on any concern. In fact, the table shows that using a low similarity threshold of 60%, which is considered to be very low in the literature, can guarantee that our system would have a specificity of 100%.

8 Conclusions and future work

In this paper, we have surveyed the literature for the most common fingerprint acquisition devices and the different available fingerprint image sets. We have noticed that the latest trend in fingerprint acquisition is to use different types of digital cameras and specially the use of mobile phone cameras. We have also noticed that the available fingerprint datasets acquired using digital cameras are all taken with build in degradations due to specific lighting, different background, low camera quality, image compression, and many more. In this paper we built our own high quality fingerprint image set that can be used to study the impact of different factors on the accuracy of a fingerprint recognition accuracy. Based on the results of our experiments, we concluded that the area of image acquisition must be given more attention than what it has now. We must consider the image quality with no less importance than other main stages of biometrics framework such as feature extraction and matching techniques. Through our pilot study, we focus on choosing the best QF of image that provides no loss of information that may be important in determining the identity of the individual. Moreover, there is no need to keep the large size of the image when it is enough to use a certain percentage of the size of this image to identify the individual. Our experiments prove that using (20%–30%) of the original image size is good to extract enough features with no loss of important information. In this work, we built a full fingerprint recognition system. In the future, we can study other applications such as the effect of image compression on identification through iris recognition, hand geometry, etc. Another future direction could answer the question: Can we compress children (under 10 years) fingerprint with the fingerprint information preserved? This is important because I notice during dataset collection that fingerprints of children are not sufficiently clear.

Acknowledgments This work is supported by the Jordan University of Science and Technology Deanship of Research project number 20150348.

References

1. Al-alem F, Alsmirat MA, Al-Ayyoub M (2016) On the road to the internet of biometric things: a survey of fingerprint acquisition technologies and fingerprint databases. In: 13th ACS/IEEE international conference on computer systems and applications (AICCSA 2016). IEEE
2. Behera B, Lalwani A, Awate A (2014) Using webcam to enhance fingerprint recognition. In: Articulated motion and deformable objects, pp 51–60. Springer
3. Bhargava N, Bhargava DR, Mathuria M, Dixit P (2013) Fingerprint minutiae matching using region of interest. International Journal of Computer Trends and Technology 4(4):515–518
4. Biometric Recognition Group - ATVS. http://atvs.ii.uam.es/. Accessed: 2016-08-16
5. Cappelli R, Ferrara M, Franco A, Maltoni D (2007) Fingerprint verification competition 2006. Biometric Technology Today 15(7):7–9

6. CASIA-FingerprintV5. http://biometrics.idealtest.org/ (2010). Accessed: 2016-08-16
7. Chang X, Ma Z, Lin M, Yang Y, Hauptmann AG (2017) Feature interaction augmented sparse learning for fast kinect motion detection. IEEE Trans Image Process 26(8):3911–3920. https://doi.org/10.1109/TIP.2017.2708506
8. Daugman J, Downing C (2008) Effect of severe image compression on iris recognition performance. IEEE Transactions on Information Forensics and Security 3(1):52–61
9. Derawi MO, Gafurov D, Larsen R, Busch C, Bours P (2010) Fusion of gait and fingerprint for user authentication on mobile devices. In: The 2nd international workshop on security and communication networks (IWSCN), pp 1–6. IEEE
10. Drake D (2008) Fingerprint abstraction layer for linux
11. Funk W, Arnold M, Busch C, Munde A (2005) Evaluation of image compression algorithms for fingerprint and face recognition systems. In: Proceedings from the 6th annual IEEE SMC information assurance workshop, pp 72–78. IEEE
12. FVC2006: the Fourth International Fingerprint Verification Competition. http://bias.csr.unibo.it/fvc2006/ (2006). Accessed: 2016-08-13
13. Hannah JG, Gladis D (2015) Feature extraction with thinning algorithms for precise cretoscopy. Int. J. Comput. Appl. 8(29):1–7
14. Hannah GJ, GD (2014) Dactyloscopy and comparison of algorithms for efficacious minutiae extraction. In: International conference on advance research in engineering and technology, pp 52–57
15. Hassanat AB, Alkasassbeh M, Al-awadi M, Alhasanat EA (2015) Colour-based lips segmentation method using artificial neural networks. In: 2015 6th international conference on information and communication systems (ICICS), pp 188–193. https://doi.org/10.1109/IACS.2015.7103225
16. Hiew BY, Teoh ABJ, Yin OS (2010) A secure digital camera based fingerprint verification system. J Vis Commun Image Represent 21(3):219–231. https://doi.org/10.1016/j.jvcir.2009.12.003. http://www.sciencedirect.com/science/article/pii/S1047320309001576
17. Hu C, Yin J, Zhu E, Chen H, Li Y (2010) A composite fingerprint segmentation based on log-gabor filter and orientation reliability. In: 17Th IEEE international conference on image processing, pp 3097–3100. IEEE
18. Irtaza A, Jaffar MA (2015) Categorical image retrieval through genetically optimized support vector machines (gosvm) and hybrid texture features. SIViP 9(7):1503–1519. https://doi.org/10.1007/s11760-013-0601-8
19. Islam MR, Sayeed MS, Samraj A et al (2008) Fingerprint authentication system using a low-priced webcam. In: The international conference on data management (ICDM 2008), IMT Ghaziabad, India, pp 689–697
20. Islam MR, Sayeed MS, Samraj A et al (2010) Technology review: image enhancement, feature extraction and template protection of a fingerprint authentication system. J Appl Sci (Faisalabad) 10(14):1397–1404
21. Ives RW, Broussard RP, Kennell LR, Soldan DL (2008) Effects of image compression on iris recognition system performance. Journal of Electronic Imaging 17(1):011,015–011,015–8. https://doi.org/10.1117/1.2891313
22. Jain AK, Arora SS, Best-Rowden L, Cao K, Sudhish PS, Bhatnagar A (2015) Biometrics for child vaccination and welfare: Persistence of fingerprint recognition for infants and toddlers. arXiv:1504.04651
23. Johnson P, Hua F, Schuckers S (2013) Texture modeling for synthetic fingerprint generation. In: the IEEE conference on computer vision and pattern recognition workshops, pp 154–159. https://doi.org/10.1109/CVPRW.2013.30
24. Jung SM (2013) Design of low power anf high speed cmos fingerprint sensor. International Journal of Bio-Science and Bio-Technology 5(2)
25. K PV, Pradsad G, Chandrasekhar B (2013) Image compression effects on face recognition for images with reduction in size International Journal of Computer Applications 61(22)
26. Khalil MS (2015) Reference point detection for camera-based fingerprint image based on wavelet transformation. Biomedical engineering online 14(1):40
27. Kumar A, Jilani TA (2015) A simple and efficient roadmap to process fingerprint images in frequency domain. Int J Comput Appl 112(4):19–25
28. Kurniawan F, Khalil MS, Khan MK (2013) Core-point detection on camera-based fingerprint image. In: International symposium on biometrics and security technologies (ISBAST), pp 241–246. IEEE
29. Lee HC, Ramotowski R, Gaensslen RE (2001) Advances in fingerprint technology, 2nd edn. CRC press, Boca Raton

30. Li G, Yang B, Busch C (2013) Lightweight quality metrics for smartphone camera based fingerprint samples. In: 9th international conference on intelligent information hiding and multimedia signal processing, pp 342–345. IEEE
31. Liu E, Zhao H, Guo F, Liang J, Tian J (2011) Fingerprint segmentation based on an adaboost classifier. Frontiers of Computer Science in China 5(2):148–157
32. Ma L, Tan T, Wang Y, Zhang D (2004) Efficient iris recognition by characterizing key local variations. IEEE Trans Image Process 13(6):739–750
33. Maio D, Maltoni D, Cappelli R, Wayman J, Jain AK (2002) Fvc2002: Second fingerprint verification competition. In: Proceedings of 16th international conference on pattern recognition (ICPR2002), Quebec City, pp 811–814
34. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2004) FVC2004: 3rd fingerprint verification competition, pp 1–7. Springer, Berlin. https://doi.org/10.1007/978-3-540-25948-0_1
35. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain PK. FVC2000: Fingerprint Verification Competition. Tech. rep. (2000). [Online: http://bias.csr.unibo.it/fvc2000/default.asp, accessed 13-August-2016]
36. Mascher-Kampfer A, Stögner H, Uhl A (2007) Comparison of compression algorithms' impact on fingerprint and face recognition accuracy. In: Visual communications and image processing, pp 650,810–1
37. Modi SK, Elliott SJ (2006) Impact of image quality on performance: comparison of young and elderly fingerprints. In: Sirlantzis K (ed) Proceedings of the 6th international conference on recent advances in soft computing (RASC 2006), pp 449–45
38. Modi SK, Elliott SJ, Whetsone J, Kim H (2007) Impact of age groups on fingerprint recognition performance. In: 2007 IEEE Workshop on Automatic Identification Advanced Technologies, pp 19–23. https://doi.org/10.1109/AUTOID.2007.380586
39. Mohammedsayeemuddin S, Pithadia PV, Vandra D (2014) A simple and novel fingerprint image segmentation algorithm. In: International conference on issues and challenges in intelligent computing techniques (ICICT), pp 756–759. IEEE
40. Mueller R, Sanchez-Reillo R (2009) An approach to biometric identity management using low cost equipment. In: 5th international conference on intelligent information hiding and multimedia signal processing, pp 1096–1100. IEEE
41. NIST Biometric Image Software. http://www.nist.gov/itl/iad/ig/nbis.cfm (2015). Accessed: 2016-08-16
42. Patel V, Thacker K, Shah APV (2014) An approach for fingerprint recognition based on minutia points. International Journal of Advance Engineering and Research Development 1(4):1–9
43. Piuri V, Scotti F (2008) Fingerprint biometrics via low-cost sensors and webcams. In: 2nd IEEE international conference on biometrics: theory, applications and systems, pp 1–6. IEEE
44. Raghavendra R, Busch C, Yang B (2013) Scaling-robust fingerprint verification with smartphone camera in real-life scenarios. In: IEEE 6th international conference on biometrics: theory, applications and systems (BTAS), pp 1–8. IEEE
45. Saad MA, Pinson MH, Nicholas DG, Van Kets N, Van Wallendael G, Da Silva R, Jaladi RV, Corriveau PJ (2015) Impact of camera pixel count and monitor resolution perceptual image quality. In: Colour and visual computing symposium (CVCS), 2015, pp 1–6. IEEE
46. Sahu D, Shrivas R (2013) Fingerprint reorganization using minutiae based matching for identification and verification. International Journal of Science and Research
47. Sankaran A, Dhamecha TI, Vatsa M, Singh R (2011) On matching latent to latent fingerprints. In: 2011 international joint conference on biometrics (IJCB), pp 1–6, https://doi.org/10.1109/IJCB.2011.6117525
48. Sankaran A, Vatsa M, Singh R (2012) Hierarchical fusion for matching simultaneous latent fingerprint. In: IEEE 5th international conference on biometrics: theory, applications and systems (BTAS), pp 377–382. https://doi.org/10.1109/BTAS.2012.6374604
49. Sankaran A, Vatsa M, Singh R (2015) Multisensor optical and latent fingerprint database. IEEE Access 3:653–665. https://doi.org/10.1109/ACCESS.2015.2428631
50. Setlak D (1999) Electric field fingerprint sensor apparatus and related methods. https://www.google.com/patents/US5963679. US Patent 5,963,679
51. Shobhraj NR, Kidwai MA (2014) Fingerprint recognition system. International Journal of Innovative Science, Engineering and Technology 1(3):2348–7968
52. Silvestre-Blanes J (2015) Scalability in industrial image processing applications. In: Telecommunications forum telfor (TELFor), 2015 23rd, pp 744–747. IEEE
53. Stoney DA (1988) Distribution of epidermal ridge minutiae. Am J Phys Anthropol 77(3):367–376. https://doi.org/10.1002/ajpa.1330770309

54. Thai R (2003) Fingerprint image enhancement and minutiae extraction. Ph.D. thesis, Computer Science and software engineering University of western Australia
55. Teoh AB, Ngo DC (2006) Preprocessing of fingerprint images captured with a digital camera. In: The 9th international conference on control, automation, robotics and vision (ICARCV), pp 1–6. IEEE
56. Tong XF, Li PF (2011) Fingerprint image segmentation based on fingerprint ridge intensity. In: International conference on machine learning and cybernetics (ICMLC), vol 4, pp 1780–1784. IEEE
57. Uysal M, Gorgunoglu S (2014) Ridge pattern representation for fingerprint indexing. Elektronika ir Elektrotechnika 20(7):65–68
58. Webb L, Mathekga M (2014) Towards a complete rule-based classification approach for flat fingerprints. In: 2nd international symposium on computing and networking, pp 549–555. IEEE
59. Wu J, Bisio I, Gniady C, Hossain E, Valla M, Li H (2014) Context-aware networking and communications: Part 1 [guest editorial]. IEEE Commun Mag 52(6):14–15. https://doi.org/10.1109/MCOM.2014.6829939

Mohammad A. Alsmirat received his PhD Degree in computer engineering from Wayne State University in 2013, his Master Degree in Computer Science from New York Institute of Technology in 2003, and his Bachelor Degree in Computer Science from Jordan University of Science and Technology in 2002. He was a full-time lecturer in the Department of Computer Science at Jordan University of Science and Technology from October, 2003 until December, 2006. He then joined the Department of Electrical and Computer Engineering (ECE) at Wayne State University as a Graduate Research Assistant in January, 2007 to pursue his PhD degree in Computer Engineering. He served as a Graduate Teaching Assistant in the ECE Department between August, 2007 and December, 2012. After that, Dr. Alsmirat worked as a Research and Development Engineer for General Motors for 9 months until he moved to Jordan to join the department of computer science in Jordan University of Science and Technology as an assistant professor and he is presently working there.

Fatimah Al-Alem is a Research Associate at Computer Science Department at Jordan University of Science and Technology. Her M.Sc. degree has been received from at Jordan University of Science and Technology in Computer Science. Her main research interests is image processing and machine learning.

Mahmoud Al-Ayyoub Received his Ph.D. in computer science from Stony Brook University in 2010. He is currently an associate professor of computer science at Jordan University of Science and Technology (JUST). His research interests include cloud computing, high performance computing, machine learning and AI. He is the co-director of the High Performance and Cloud Computing research lab at JUST.

Yaser Jararweh received his Ph.D. in Computer Engineering from University of Arizona in 2010. He is currently an associate professor of Computer Science at Jordan University of Science and Technology, Jordan. He has co-authored about seventy technical papers in established journals and conferences in fields related to cloud computing, HPC, SDN and Big Data. He was one of the TPC Co-Chair, IEEE Globecom 2013 International Workshop on Cloud Computing Systems, and Networks, and Applications (CCSNA). He is a steering committee member for CCSNA 2014 and CCSNA 2015 with ICC. He is the General Co-Chair in IEEE International Workshop on Software Defined Systems SDS-2014 and SDS 2015. He is also chairing many IEEE events such as ICICS, SNAMS, BDSN, IoTSMS and many others. Dr. Jararweh served as a guest editor for many special issues in different established journals. Also, he is the steering committee chair of the IBM Cloud Academy Conference. He is associate editor in the Cluster Computing Journal (Springer), Information Processing & Management (Elsevier) and others.

Brij Gupta received PhD degree from Indian Institute of Technology Roorkee, India in the area of Information and Cyber Security. In 2009, he was selected for Canadian Commonwealth Scholarship and awarded by Government of Canada Award ($10,000). He spent more than six months in University of Saskatchewan (UofS), Canada to complete a portion of his research work. Dr. Gupta has excellent academic record throughout his carrier, was among the college toppers, during Bachelor?s degree and awarded merit scholarship for his excellent performance. In addition, he was also awarded Fellowship from Ministry of Human Resource Development (MHRD), Government of India to carry his Doctoral research work. He has published more than 70 research papers (including 01 book and 08 chapters) in International Journals and Conferences of high repute including IEEE, Elsevier, ACM, Springer, Wiley Inderscience, etc. He has visited several countries, i.e. Canada, Japan, Malaysia, Hong-Kong, etc to present his research work. His biography was selected and publishes in the 30th Edition of Marquis Who's Who in the World, 2012. He is also working principal investigator of various R&D projects. He is also serving as reviewer for Journals of IEEE, Springer, Wiley, Taylor & Francis, etc. Currently he is guiding 08 students for their Master's and Doctoral research work in the area of Information and Cyber Security. He also served as Organizing Chair of Special Session on Recent Advancements in Cyber Security (SS-CBS) in IEEE Global Conference on Consumer Electronics (GCCE), Japan in 2014 and 2015. Earlier he served as co-convener of National Conference on Emerging Trends in Engineering, Science Technology & Management (ETESTM-12), India, April, 2012. In addition, Dr Gupta received Best Poster presentation award and People choice award for Poster presentation in CSPC-2014, Aug., 2014, Malaysia. He served as Jury in All IEEE-R10 Young Engineers' Humanitarian Challenge (AIYEHUM-2014), 2014. He has also served as founder and organizing chair of International Workshop on Future Information Security, Privacy and Forensics for Complex Systems (FISP-2015) in conjunction with ACM International Conference on Computing Frontiers (CF-2015), Ischia, Italy in May 2015. He is also serving as guest editor of various Journals.