

Research Article

Research and Development of Palmprint Authentication System Based on Android Smartphones

Xinman Zhang ¹, Kunlei Jing ¹ and Guokun Song ²

¹School of Automation Science and Engineering, Faculty of Electronic and Information Engineering, MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

²Sichuan Gas Turbine Research Institute of AVIC, No. 6 Xinjun Road, Xindu District, Chengdu, Sichuan, China

Correspondence should be addressed to Xinman Zhang; zhangxinman@mail.xjtu.edu.cn

Received 27 June 2020; Revised 8 August 2020; Accepted 23 October 2020; Published 16 November 2020

Academic Editor: Peter Brida

Copyright © 2020 Xinman Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The security problems of online transactions by smartphones reveal extreme demand for reliable identity authentication systems. With a lower risk of forgery, richer texture, and more comfortable acquisition mode, compared with face, fingerprint, and iris, palmprint is rarely adopted for identity authentication. In this paper, we develop an effective and full-function palmprint authentication system regarding the application on an Android smartphone, which bridges the algorithmic study and application of palmprint authentication. In more detail, an overall system framework is designed with complete functions, including palmprint acquisition, key points location, ROI segmentation, feature extraction, and feature coding. Basically, we develop a palmprint authentication system having user-friendly interfaces and good compatibility with the Android smartphone. Particularly, on the one hand, to guarantee the effectiveness and efficiency of the system, we exploit the practical Log-Gabor filter for feature extraction and discuss the impact of filtering direction, downsampling ratio, and discriminative feature coding to propose an improved algorithm. On the other hand, after exploring the hardware components of the smartphone and the technical development of the Android system, we provide an open technology to extend the biometric methods to real-world applications. On the public PolyU databases, simulation results suggest that the improved algorithm outperforms the original one with a promising accuracy of 100% and a good speed of 0.041 seconds. In real-world authentication, the developed system achieves an accuracy of 98.40% and a speed of 0.051 seconds. All the results verify the accuracy and timeliness of the developed system.

1. Introduction

Mobile Internet technologies have brought humans a new lifestyle. The emergence of smartphones has better accelerated people's daily life with popular social networks such as online transactions [1]. However, frequent online transactions by smartphones often bring security problems. Specifically, to facilitate the payment, the "mobile wallet" should be linked with a cash card. Then, phenomena like Internet fraud and network "phishing" begin to spread extensively, which leads consumers to be still distrustful of the reliability of mobile payment. Moreover, when people trade online, they have to set different complex passwords on various occasions, which may be forgettable. These facts indicate that mobile payment still demands for reliable mechanisms extremely [2]. Over the

last years, fingerprint authentication has drawn significant attention. However, fingerprint can be easily damaged due to physical work and is not informative enough for identity authentication. For iris authentication, users have to suffer from an uncomfortable acquisition mode.

Recently, many researchers focus on face biometrics. Nevertheless, face information is always exposed to the environment and can be easily stolen for identity forgery. As for hand geometry biometric, its physical uniqueness needs further discussion [1]. All in all, with the advantages of abundant textures, comfortable acquisition mode, and lower risk of forgery, palmprint is more acceptable to users in some situations, compared with the other biometrics [1].

The earliest palmprint authentication system was developed on an embedded device by Zhang and Shu team in

1999. They acquired palmprint by scanning the printed palmprint templates [3]. The offline operating mode renders the system not only time-consuming but also unacceptable to users. To make improvement, in 2003, they utilized CCD to acquire images and set pegs to restrain the palm position. However, the improved equipment is immovable for its large size and still unacceptable for its contacting acquisition mode [4]. In 2010, Zhang et al.'s team further developed a new palmprint device with a smaller size compared with the previous ones, but this device remains to suffer from the contact-based acquisition mode [5]. In 2015, Aykut and Ekinci developed a contactless palmprint authentication system that obtained good usage experience whereas it is inconvenient for mobile applications. In summary, the existing palmprint authentication systems are still not mature enough for general application and deserve in-depth research. Table 1 presents a total comparison among the abovementioned systems.

From Table 1, the contact-based palmprint authentication systems are unacceptable for the uncomfortable acquisition mode, while the contactless systems are not robust to the real disturbance [7]. Besides, the widely adopted feature extractor, Log-Gabor filter, in these systems has a high computation cost, which can greatly reduce the timeliness of the system [8]. Furthermore, the adopted Quadrant coding is unsuitable to make full play of the discrimination of the extracted features [9]. In current years, the Android operation system-based smartphones have occupied 76% of the smartphone market in the world. So, it is meaningful to further study an efficient contactless palmprint authentication system with the application on the Android smartphone.

According to the aforementioned analysis, we design an overall system framework with complete functions for palmprint authentication. Specifically, a vital acquisition scheme is established to ensure the image quality in the contactless acquisition process. Some improvements are introduced to the Log-Gabor filter and the original coding method to reduce the computational complexity and enhance the discriminability of the coding results. On the basis, a system with user-friendly interfaces and good compatibility is developed for the application on the Android smartphone.

Most systems only focus on the palmprint authentication method. Actually, as the primary work, it is crucial to establish an acquisition scheme. Thus, the image quality and accurate ROI segmentation can be ensured for more reliable identity authentication. Besides, the authentication algorithm is the core issue of our system. Considering the practicality and effectiveness, the Log-Gabor filter is the most suitable one for our system. However, to guarantee the timeliness of our system, we should consider reducing the computational complexity of the Log-Gabor filter. Moreover, a proper coding method should be exploited to enhance the discriminability of the extracted features, which helps the system distinguish one class from the others more easily. Finally, all the procedures should be programmed into modules and linked to the well-designed functional interfaces to

facilitate the operation. Depending on the above motivations, the overall system framework proposed in this paper is displayed in Figure 1. Table 2 lists all the involved evaluation parameters and their definitions in advance. The main contributions of our work are summarized as follows:

- (i) A palmprint acquisition scheme is proposed against the potential disturbances such as the variation of palm posture and illumination. Then, we present a complete set of procedures for palmprint ROI extraction, such as key points location, palmprint angle correction, and the final ROI segmentation.
- (ii) To guarantee the efficiency of the system and obtain more discriminative feature coding results, we discuss the impact of filtering direction and downsampling to the performance of Log-Gabor filter and improve the original coding scheme [10, 11].
- (iii) A palmprint authentication framework with complete functions is designed. We explore and show the details of developing the corresponding system with user-friendly interfaces and good compatibility with Android smartphones.
- (iv) Our work provides an open technology to extend the palmprint authentication algorithms to real-world applications, which bridges the algorithmic studies and applications of biometrics.

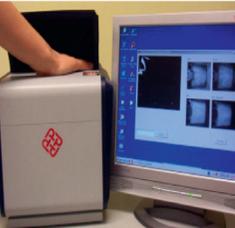
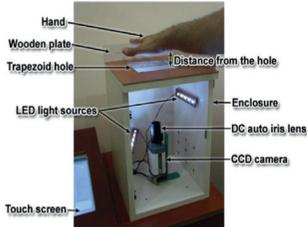
The rest of the paper is organized as follows: Section 2 defines an acquisition scheme and provides a method for palmprint ROI extraction. Section 3 describes the improvements for palmprint authentication. Section 4 introduces the hardware components of the smartphone and the software realization of the palmprint authentication system. Section 5 performs experimental verifications. Section 6 concludes this paper.

2. Palmprint Acquisition and ROI Extraction

2.1. Acquisition Scheme. In our system, ROI segmentation is finished according to three key points among fingers. However, palm gesture and the position of the palm can affect the key point locating [12, 13]. Taking some cases into consideration. The palm deformation often produces visual textures. The large gaps among fingers can lead to imprecise locating results. A long distance between palm and camera can lead the acquired image to have too much background. All of these potential disturbances will bring difficulties for key points locating. So, it is necessary to define an acquisition scheme.

Considering the aforementioned potential disturbances, the acquisition scheme in this paper is determined as follows: five fingers extended and the other four fingers close to each other except for the thumb. A red line and a center point are marked on the interface to conduct the position of the finger root and palm center.

TABLE 1: Comprehensive comparisons among some typical palmprint authentication systems.

Team/company	Products	Acquisition approach	Performance online/offline	Advantages	Disadvantages
HKPU, in 1999 [3]		Contact-based	Offline	(i) Robust to disturbance	(i) Poor usage experience (ii) Poor portability (iii) Poor timeliness
HKPU, in 2003 [4]		Contact-based	Online	(i) Robust to disturbance (ii) Good timeliness	(i) Poor usage experience (ii) Poor portability
HKPU, in 2010 [5]		Contact-based	Online	(i) Robust to disturbance (ii) Good timeliness	(i) Poor usage experience (ii) Poor portability
Karadeniz Technical University, in 2015 [6]		Contactless	Online	(i) Good usage experience (ii) Good timeliness	(i) Not robust to disturbance (ii) Poor portability

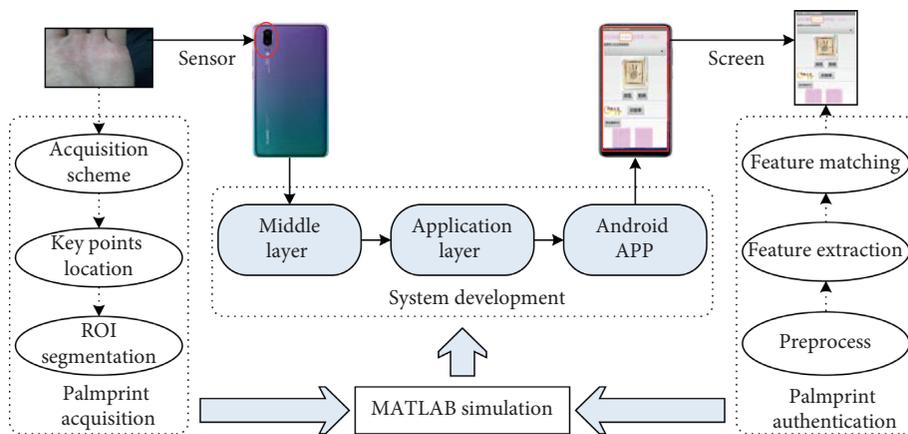


FIGURE 1: The designed framework of the palmprint authentication system.

TABLE 2: Parameter definitions.

Parameter	Definition
FP (false positive)	The number of imposter acceptances
TN (true negative)	The number of imposter rejections
FN (false negative)	The number of legitimate rejections
TP (true positive)	The number of legitimate acceptances
AA (authentication accuracy)	$AA = (TP + TN)/(TP + FP + TN + FN)$
FAR (false acceptance rate)	$FAR = FP/(TN + FP)$
FRR (false rejection rate)	$FRR = FN/(TP + FN)$
EER (equal error rate)	Equates to the point at which the FAR and FRR cross
D_1	Interval length of the intraclass hamming distance distribution
D_2	Interval length of the interclass hamming distance distribution
Dis	The distance from the upper bound of D_1 to the lower bound of D_2
L_1	Dis/D_1
L_2	Dis/D_2

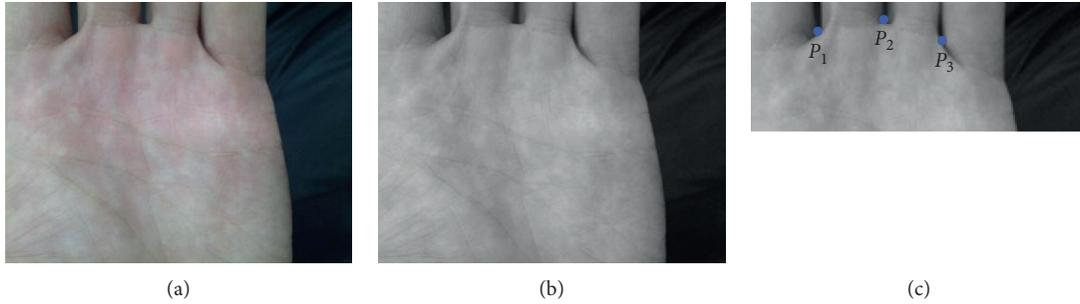


FIGURE 2: Palmprint images. (a) Original palmprint image. (b) Gray palmprint image. (c) Reserved subimage.

2.2. Key Point Location

2.2.1. Image Graying. The original palmprint image (540×600) shown in Figure 2(a) is converted into a gray image shown in Figure 2(b). To cut down the searching range, the upper portion (270×600) of the gray image is reserved, shown in Figure 2(c).

In the reserved subimage, the gaps between fingers root are deemed as the key points and denoted as P_1 , P_2 , and P_3 , respectively, as shown in Figure 2(c).

2.2.2. Columns of Key Points. The average gray value versus the column of the reserved subimage can be calculated by

$$\begin{cases} F_y(i, j) = \frac{(F(i, j+1) - F(i, j-1))}{2}, \\ F_x(i, j) = \frac{(F(i+1, j) - F(i-1, j))}{2}, \end{cases} \quad (1)$$

where F_y and F_x denote the gradient versus column and row, respectively. (i, j) denotes the pixel coordinate. The three local minima are shown in Figure 3, which correspond to the column indexes of the three key points.

2.2.3. Rows of Key Points. The three subimages (100×20) containing three key points are, respectively, cropped from Figure 4(a) shown in Figure 4(b). For each subimage, the row index of the key point can also be obtained according to formula (1). The average gradient value versus the rows of the first subimage is shown in Figure 5. The X -coordinate of the maximal value corresponds to the row index of the key point.

2.3. ROI Segmentation. The acquired images often contain redundant information such as fingers and background. Palmprint ROI refers to the palm center that contains three flexor lines [14, 15]. Here, we segment ROI according to the above located key points.

Since the size of a smartphone screen is generally different from the size of a palmprint image and there exists rotation between them, it is necessary to correct the scale and angle of the palmprint image before ROI segmentation.

2.3.1. Scale Correction. The key point lying in the original image with a size of $w \times h$ is denoted as $P_i(x_i, y_i)$ ($i = 1, 2, 3$). The corresponding position in the smartphone interface with a size of $w' \times h'$ is denoted as $P'_i(x'_i, y'_i)$ ($i = 1, 2, 3$). Then, the transformation can be defined as

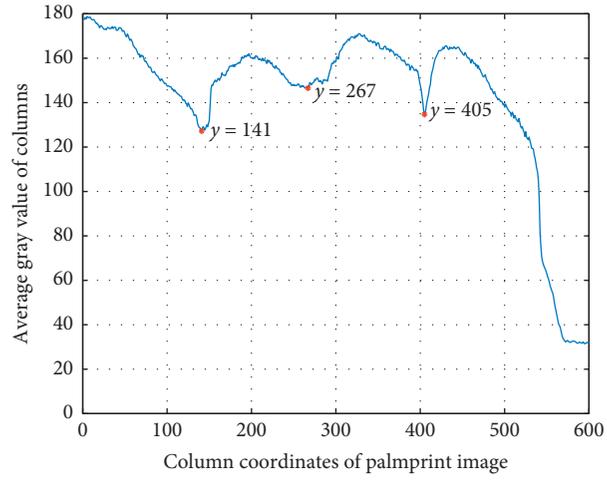


FIGURE 3: Average gray value versus the columns.

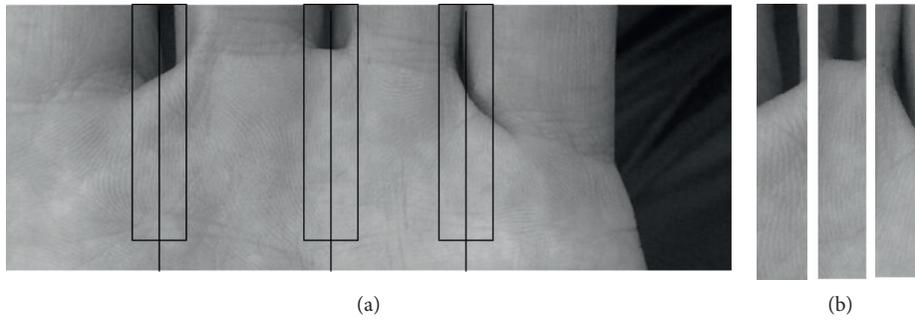


FIGURE 4: Subimages containing the three key points. (a) Marked subimages. (b) Cropped subimages.

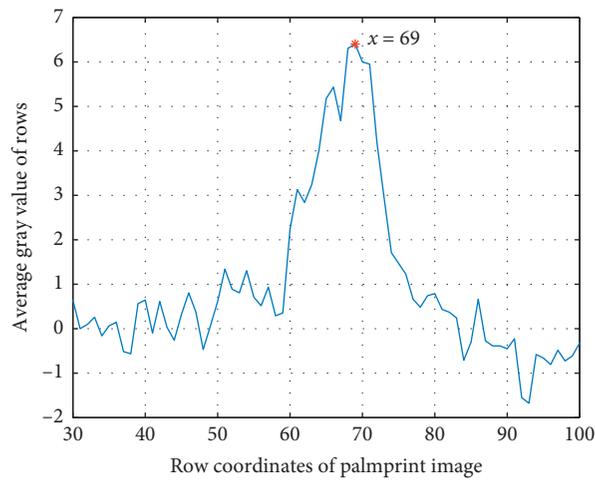


FIGURE 5: Average gray value versus the rows.

$$\begin{cases} x'_i = x_i \times \left(\frac{w'}{w}\right), \\ y'_i = y_i \times \left(\frac{h'}{h}\right). \end{cases} \quad (2)$$

2.3.2. Rotation Correction. The purpose of the rotation is to place P_1 and P_3 on a horizontal line to facilitate ROI segmentation. The calculation of the rotation angle is shown as follows:

$$a = \arctan\left(\frac{(y'_3 - y'_1)}{(x'_3 - x'_1)}\right), \quad (3)$$

where (x'_1, y'_1) and (x'_3, y'_3) are the coordinates of P'_1 and P'_3 , respectively. The corresponding coordinates after rotation are denoted as $P''_i(x''_i, y''_i)$ ($i = 1, 2, 3$). We define the rotation transformation as follows:

$$\begin{cases} x''_i = \left[\left(x'_i - \frac{w'}{2} \right) \times \cos\left(\frac{a}{180} \times \pi\right) + \left(y'_i - \frac{h'}{2} \right) \times \sin\left(\frac{a}{180} \times \pi\right) + \frac{w'}{2} \right] \\ y''_i = \left[-1 \times \left(x'_i - \frac{w'}{2} \right) \times \sin\left(\frac{a}{180} \times \pi\right) + \left(y'_i - \frac{h'}{2} \right) \times \cos\left(\frac{a}{180} \times \pi\right) + \frac{h'}{2} \right], \end{cases} \quad (4)$$

where $\lceil \cdot \rceil$ reserves the integer value, while the line $P''_1P''_3$ is set as the x -axis, and its vertical line is set as the y -axis. Then, the coordinate system is established in the palmprint image. The length of $P''_1P''_3$ is denoted as d . Moving it down alongside the y -axis by $d/3$, a square is built up with the side length of d , as shown in Figure 6(a). To obtain a standard ROI, we scale the segmented square to a size of 128×128 , shown in Figure 6(b).

3. Palmprint Authentication

In experiments, we find that image preprocessing has a little effect on ROI segmentation results. To shorten the running time of the system, we perform image preprocessing after ROI segmentation.

3.1. Image Preprocessing

3.1.1. Gray Enhancement. The pixel value of the original ROI image is denoted as $F_{old}(i, j)$, and the pixel value after the

preprocessing is denoted as $F_{new}(i, j)$. The gray enhancement is defined as [16]

$$F_{new}(i, j) = \begin{cases} low_out, & F_{old}(i, j) < low_in, \\ high_out, & F_{old}(i, j) > low_high, \\ CLog(1 + F_{old}(i, j)), & \text{else.} \end{cases} \quad (5)$$

where $low_in, high_in, low_out$, and $high_out$ are four given constants, according to which the original pixel value between $[low_in, high_in]$ is then transformed between $[low_out, high_out]$.

3.1.2. Gray Normalization. To balance the uneven illumination, we normalize the gray image by the following formulas [17]:

$$F_{new}(i, j) = \begin{cases} M_o + \sqrt{\frac{VAR_o(F_{old}(i, j) - M(F))^2}{VAR(F)}}, & F_{old}(i, j) > M(F), \\ M_o - \sqrt{\frac{VAR_o(F_{old}(i, j) - M(F))^2}{VAR(F)}}, & \text{else,} \end{cases} \quad (6)$$

where $M(F)$ and $VAR(F)$ signify the mean function and the variance function, respectively. The preprocessed ROI images are shown in Figure 7(c), from which we observe that the three flexor lines and the surrounding details are more obvious and abundant.

3.2. Discussion on Log-Gabor Filter. The three principal lines in palmprint are not sufficient to signify the uniqueness of individuals. So, the feeble wrinkles around them should be amplified to provide more valuable information. What is more, direction information in palmprint should be

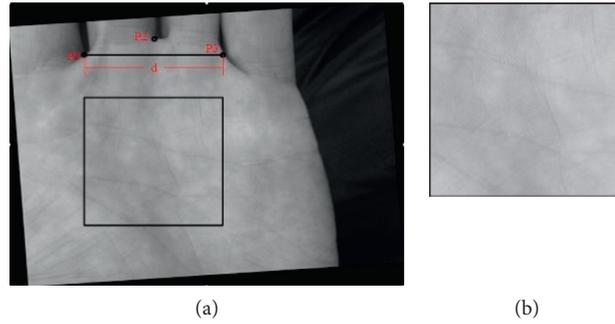


FIGURE 6: Palmprint ROI segmentation. (a) Marked ROI. (b) Segmented ROI.

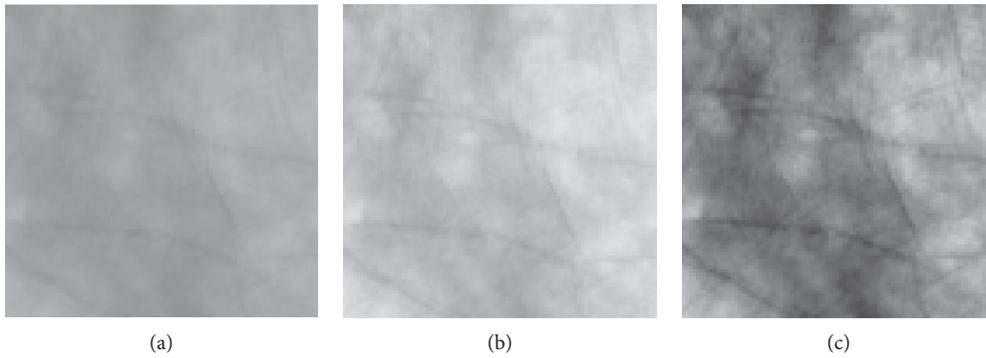


FIGURE 7: ROI image preprocessing. (a) Segmented ROI image. (b) Gray enhancement. (c) Gray normalization.

characterized. The Log-Gabor filter is capable of supporting us to extract such a kind of features. By setting various factors and filtering directions, we can obtain more representative abundant features through the Log-Gabor filter [18, 19]. Feature coding is the other important process, in which a more discriminative feature coding method helps to further improve the performance of the system.

Note feature extraction corresponding multiple factors and directions can result in a large computation. To guarantee the timeliness and the effectiveness of our system, we will take the discussion on the original Log-Gabor filter and feature coding method (we call the original algorithm in the later content).

3.2.1. Log-Gabor Filter. The transfer function of the Log-Gabor filter in the linear frequency domain is defined as [20]

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right), \quad (7)$$

where f_0 and σ denote the central frequency domain and the bandwidth, respectively. To maintain the shape of the filter, the ratio σ/f_0 is usually fixed [21]. The output of the Log-Gabor filter is related to the filtering direction. Here, we set $\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ]$ to discuss the influence of filtering direction to feature extraction results. The scale wavelength is set as 3, and the ratio is fixed by $\sigma/f_0 = 0.6$. The filtered results are shown in Figure 8. When $\theta = 0^\circ$, textures in all directions are

well characterized except for the horizontal textures. When $\theta = 45^\circ$, textures in direction 135° is mostly represented. When $\theta = 90^\circ$, the horizontal textures are the clearest. When $\theta = 135^\circ$, textures in direction 45° are clearly represented. Thus, we conclude that the extracted textures are more representative when the filtered direction is perpendicular to texture.

3.2.2. Feature Extraction and Coding. Since the distribution of palmprint textures is complex, we set $\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ]$ to cover the most directions. In addition, five scale factors, say $\sigma = [3, 4.5, 6.75, 45.5]$, are chosen in each direction to extract more abundant information, with $\sigma/f_0 = 0.6$. Then, a feature matrix with the dimension of $4 \times 5 \times 128 \times 128 = 327680$ is obtained by the original algorithm, which leads to a computation disaster [7]. To this end, we bring some improvements to the Log-Gabor filter.

In detail, the ROI image and extracted features will be downsampled to get rid of the redundant information while reducing the computational time. To obtain distinguishable coding results for the final identity matching, we adopt the Gray code to replace Quadrant code to enhance the discriminability of features. The specifics are shown as follows:

Step 1: ROI image (128×128) is downsampled to the size of 64×64 . The pixel value before and after filtering is denoted as $i(x, y)$ and $I(x, y)$, respectively. $G_m(x, y) (m = 1, 2, 3, 4)$ denotes the Log-Gabor filter

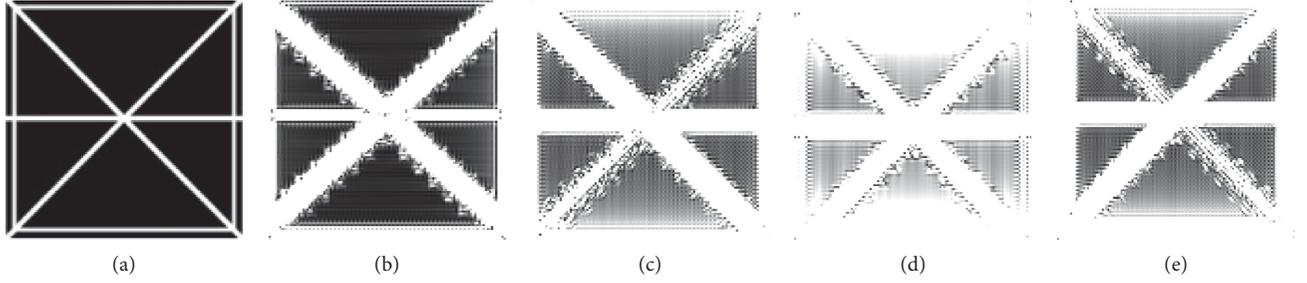


FIGURE 8: Filtering results in different directions: (a) Test image. (b) Direction 0° . (c) Direction 45° . (d) Direction 90° . (e) Direction 135° .

with different filtering directions. The filtering process (feature extraction) is formulated as follows:

$$I_m(x, y) = i(x, y) \times G_m(x, y). \quad (8)$$

Step 2: the features are downsampled with a ratio of 1 : 2 to obtain a feature matrix with a dimension of

$4 \times 32 \times 32$. Let $I_m(x, y)$ and $I'_m(x, y)$ denote the pixel value before and after the feature downsampling. The feature downsampling can be performed by the following formula:

$$I'_m(x, y) = I_m(x-1, y-1) + x[I_m(x+1, y-1) - I_m(x-1, y+1)] + y[I_m(x-1, y+1) - I_m(x-1, y-1)] + xy[I_m(x+1, y+1) + I_m(x-1, y-1)] - xy[I_m(x+1, y-1) + I_m(x-1, y+1)]. \quad (9)$$

Step 3: according to the real part and imaginary part of the features, Quadrants coding achieves two kinds of coding results, while Gray coding achieves four kinds of coding results. Hence, Gray coding is conducive to enhance the discriminability of features. Denoting the coding result as $Code I'_m = [Code I'_m R, Code I'_m I]$, then Gray coding can be defined as follows:

$$Code I'_m R = \begin{cases} 1, & \text{Re}[I'_m(x, y)] \geq 0, \\ 0, & \text{Re}[I'_m(x, y)] < 0, \end{cases} \quad (10)$$

$$Code I'_m I = \begin{cases} 1, & \text{Im}[I'_m(x, y)] \geq 0, \\ 0, & \text{Im}[I'_m(x, y)] < 0. \end{cases}$$

All the above steps are repeated regarding five scales to obtain a matrix with a dimension of $4 \times 5 \times 32 \times 32 \times 2 = 40960$. The flow chart of the improved algorithm is shown in Figure 9.

3.2.3. Feature Matching. Hamming distance is exploited for feature matching to judge whom the acquired palmprint image belongs to. By the bitwise XOR operation on two feature vectors, the rate of the number “1” is defined as Hamming distance. If the samples come from the same user, the Hamming distance can be small enough. Otherwise, the Hamming distance can be large. The calculation of Hamming distance is defined as

$$HD = \frac{\sum_{i=1}^N P(i) \otimes Q(i)}{N}, \quad (11)$$

where $P(i)$ and $Q(i)$ denote the i th elements in two vectors and N denotes the vector dimension. Theoretically, when P and Q are exactly the same, $HD = 0$. Otherwise, $HD = 1$. There exist no palmprint images that are totally the same or different in practice, so generally $HD \in (0, 1)$. The final authentication is achieved by comparing the Hamming distance with the predetermined matching threshold.

4. System Development

4.1. Structure of the Hardware and Software

4.1.1. Hardware. In recent years, Android smartphones have occupied a large market share in the world. Among the most mainstream models in 2018, HUAWEI P20 is the most popular one that carries a host processor HiSilicon Kirin 970, 6 GB RAM, 128 GB ROM, and a 6.1-inch touchscreen with a resolution ratio of 2240×1080 . Its operating system is Android 8.1. Among these, the camera and GPU are the key components. Four mainstream brands released in the same period with HUAWEI P20 are listed in Table 3 [22].

For the CMOS camera, the larger the camera numbers and the pixel size, the more information can be captured. The smaller the aperture size, the more light could be passed to enhance the image. What is more, G72MP12 is more powerful than Adreno540 and 418 for image processing. After the synthetic comparison, HUAWEI P20 is chosen as the platform of our system. The main hardware structure of HUAWEI P20 is shown in Figure 10.

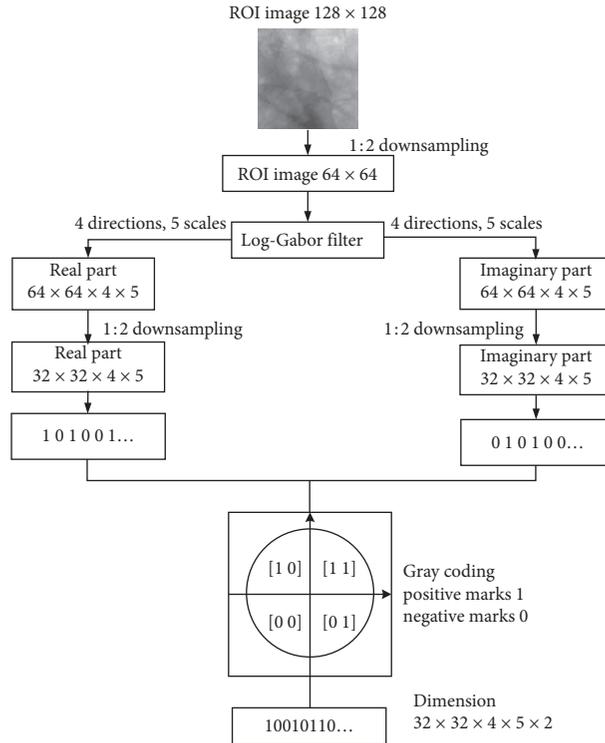


FIGURE 9: The flow diagram of the improved algorithm.

TABLE 3: Camera and GPU comparison of four mainstream Android smartphone brands.

Manufactures	Numbers	Camera			GPU
		Megapixel	Pixel size (μm)	Aperture size	
MOTOROLA X	1	21	1.1	F2.0	Adreno418
SAMSUNG S8+	1	12	1.4	F1.7	Adreno540
HTC U11+	1	12	1.4	F1.7	Adreno540
HUAWEI P20	2	20 + 8	1.55	F1.8	Mali-G72MP12

4.1.2. *Software.* According to the designed framework in Figure 1, the system development can be finished through three steps: middle layer development, application layer development, and Android APK generation [23]. The entire development structure is displayed in Figure 11. The details are described as follows:

- (i) Underlying layer development: in the QT environment, we first develop executable files to call the hardware of the smartphone to finish each procedure in the designed framework. Procedures, like palmprint acquisition, ROI segmentation, feature extraction, feature coding, and feature matching, are all separately programmed into executable functional files by C++ language.
- (ii) Application layer development: using Android SDK, we develop interfaces to call the executable functional files. Through user-friendly interfaces, the user can operate the system in real time. The development in this stage is finished in Eclipse Java EE environment.

- (iii) Android APK generation: we use NDK to compile the functional files developed in the underlying layer. The compiled files are packaged with the JNI standard to facilitate the callback from the application layer. Then, dynamic link libraries are generated and packaged into an APK to install on the smartphone.

4.2. *System Function Design.* Activity is the functional unit in the application layer. Since an activity occupies a certain part of memory, to save the resource, we integrate the interrelated functions into one activity. It is also convenient for us to debug the code and maintain the software. Our APK is composed of four activities: main activity (main interface), database activity (database interface), authentication activity (authentication interface), and acquisition activity (acquisition interface).

4.2.1. *Main Activity.* The main activity provides entrances for the other activities. Additionally, the initialization and

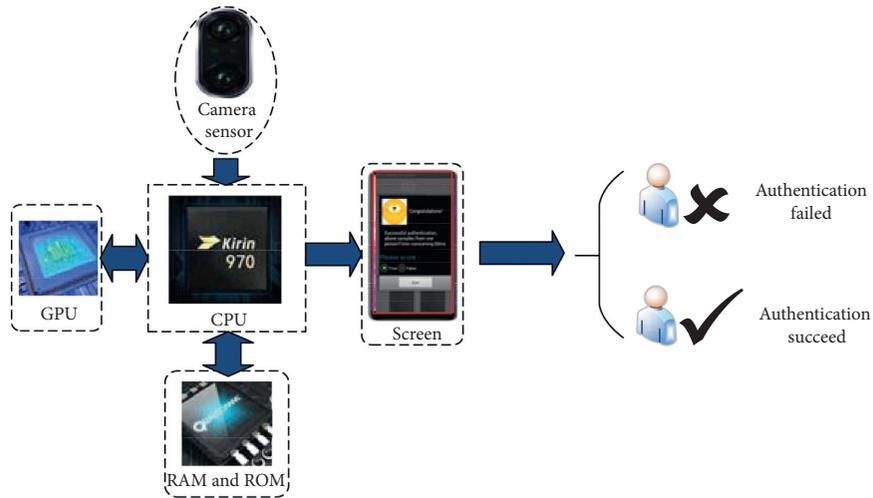


FIGURE 10: The hardware structure of HUAWEI P20.

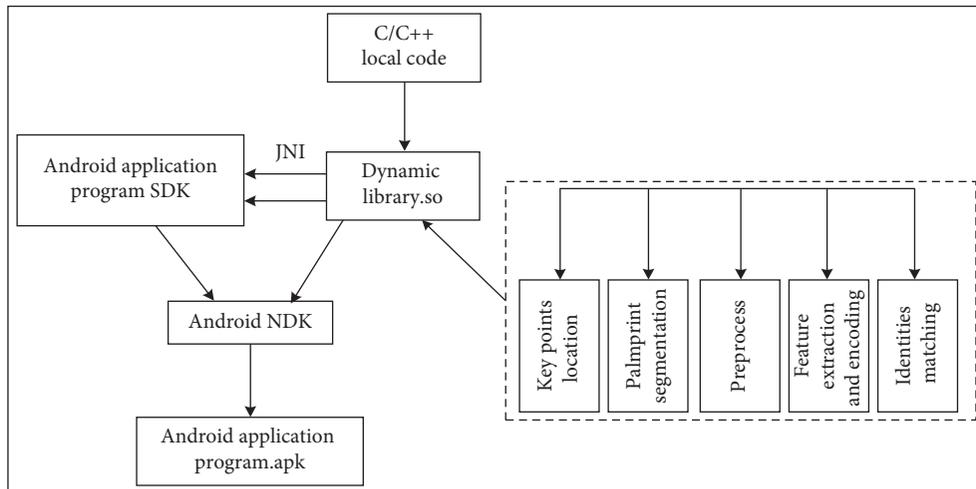


FIGURE 11: The entire structure of software development.

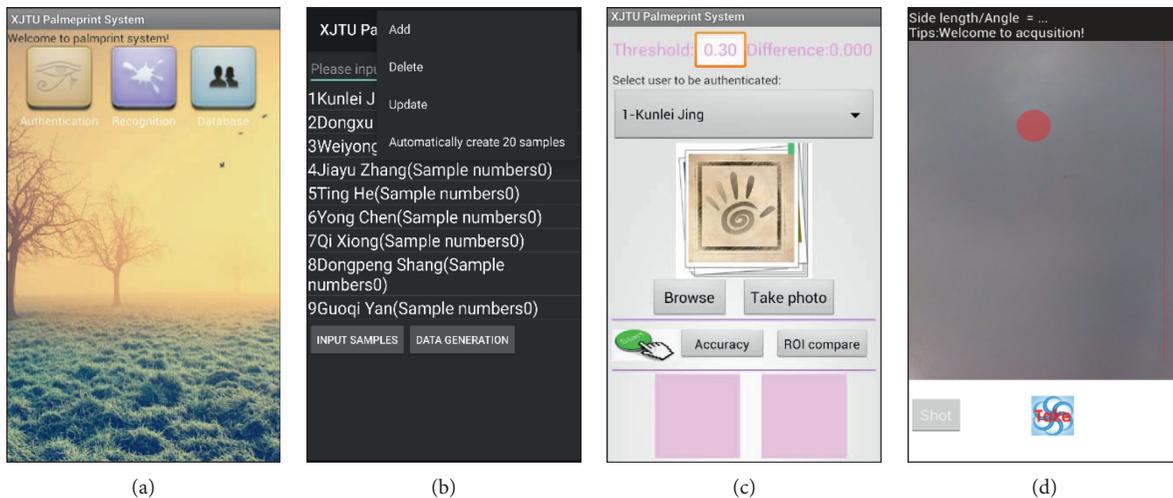


FIGURE 12: The developed four interfaces: (a) Main interface. (b) Database interface. (c) Authentication interface. (d) Acquisition interface.

storage of the database are finished in this activity. The developed main interface is shown in Figure 12(a).

4.2.2. Database Activity. Users finish identity registration in this activity. They can append their identity information by clicking “Add” button and input their palmprint images for the later authentication (5 samples are necessary) by clicking “INPUT SAMPLES” button. Features extraction is finished by clicking “DATA GENERATION” button. Meanwhile, we design “Delete” and “Update” buttons to facilitate database management. The database interface is shown in Figure 12(b).

4.2.3. Authentication Activity. Identity authentication is achieved in this activity. After choosing the identity to be confirmed in the identity list, users can click the “Browse” button or “Take photo” button to select the test sample. When clicking the “Start” button, the system will display authentication result, elapsed time, and encourage users to evaluate the authentication result. Users can examine the correct authentication number and the accuracy by clicking the “Accuracy” button. Also, they can click the “ROI Compare” button to check the difference between two segmented ROI images. The authentication interface is shown in Figure 12(c).

4.2.4. Acquisition Activity. When users enter the real-time acquisition, the rear camera is driven to work. Two modes, “long press” and “click,” are designed for the “Take” button. When it is long pressed (about 2 seconds), the located key points and ROI are displayed on the screen. Tips for distance and angle adjustment are given on the top to guide the acquisition. Users can click the “Take” button (about 0.5 seconds) to finish the final acquisition. The developed acquisition interface is shown in Figure 12(d), where the red line and point are set to conduct the palm position.

5. Results and Discussion

5.1. Verification of the Improved Algorithm

5.1.1. Statistical Analysis. The blue spectrum samples in the PolyU multispectral palmprint database containing 6000 samples from 500 volunteers are chosen to verify the improved algorithm [18]. 12 samples of each volunteer are divided into two equal parts. The first part is regarded as the training samples, while the second part is used for the test.

Each volunteer is tested 6 times, and the total intraclass test times reach 3000. For interclass verification, test samples of each volunteer are, respectively, utilized to match each training sample of a selected volunteer. The minimum Hamming distance is deemed as the final matching result. If it is less than the threshold, they are regarded as the same class. Otherwise, they are regarded as different ones [24].

In our simulation, the intraclass Hamming distances distribute in interval [0.1317, 0.2265], while the interclass

Hamming distances distribute in interval [0.3118, 0.5271]. The distribution density curves are shown in Figure 13.

Based on the above discussion, *AA*, *FRR*, and *FAR* are brought to evaluate the improved algorithm. We set the threshold $th \in [0.10, 0.55]$, and the step length is 0.01. The *FAR* and *FRR* threshold curves are drawn in Figure 14. It can be seen that the *FAR* and *FRR* curves have no intersection, which means *EER* is 0. Note Figure 14 is more intuitive for us to find the optimum threshold interval, compared with the *ROC* curve [20].

A comprehensive comparison between the improved algorithm and the original one is listed in Table 4. The evaluation items include the intraclass and interclass Hamming distance, indicator *L*, feature dimension, and the authentication time. For Hamming distance and indicator *L*, compared with the original algorithm, the improved algorithm presents more obvious separation between the intraclass and interclass Hamming distance distributions, which greatly reduce the *ERR*. In addition, we find that the feature dimension is reduced by 6.25%. This guarantees the efficiency of our system.

5.1.2. Comparison with the Other Biometrics. Apart from palmprint, there are some other biometrics that can be used for identity authentication. This subsection carries out a comparison among the proposed system and some other biometrics-based methods. We particularly consider the new emerging biometrics, *EEG* [25] and *ECG* [26–28]. The comparison results are shown in Table 5.

The collected data in Table 5 indicates that the methods based on the new biometrics all achieved promising accuracy. Recent years have witnessed the great application value of the *EEG* and *ECG* biometrics based on the brain wave and heartbeat pattern [25, 26]. Meanwhile, the good performance presented in [25–28] also manifests the great potential of the deep neural networks to learn valuable pattern information in the complicated biometric signals. All of these facts motivate us to further study identity authentication using the new biometrics by deep neural networks.

5.1.3. Complexity Analysis. As is introduced in Subsections 3.2.2 and 3.2.3, given an ROI image $y \in \mathbb{R}^{N \times M}$, the entire authentication procedure includes three stages, that is, feature extraction, feature coding, and feature matching [29, 30]. The first stage involves image downsampling and Log-Gabor filtering. The computational complexity of image downsampling is $N \times M$. After that, the image size becomes $(N/2) \times (M/2)$. According to formula (8), the filtering process should be repeated along 4 filtering directions, respectively, with 5 different scale factors [31, 32]. Hence, its computational complexity is $4 \times 5 \times (N/2) \times (M/2)$. So, the operation complexity of the feature extraction stage is $N \times M + 5 \times N \times M$. The feature coding stage involves feature downsampling and coding. Formula (9) indicates that the computational complexity of feature downsampling is $(N/2) \times (M/2)$. Then, the feature dimension becomes $(N/4) \times (M/4)$. From formula (10), we know the coding process

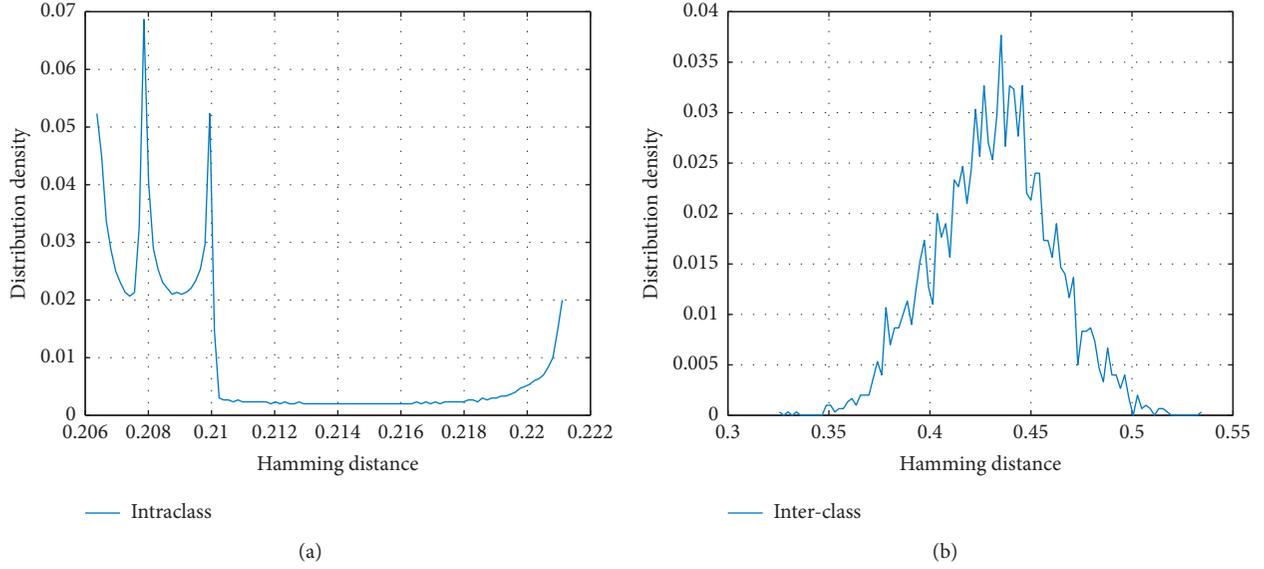


FIGURE 13: Distribution density curves of the Hamming distance. (a) Intra-class. (b) Inter-class.

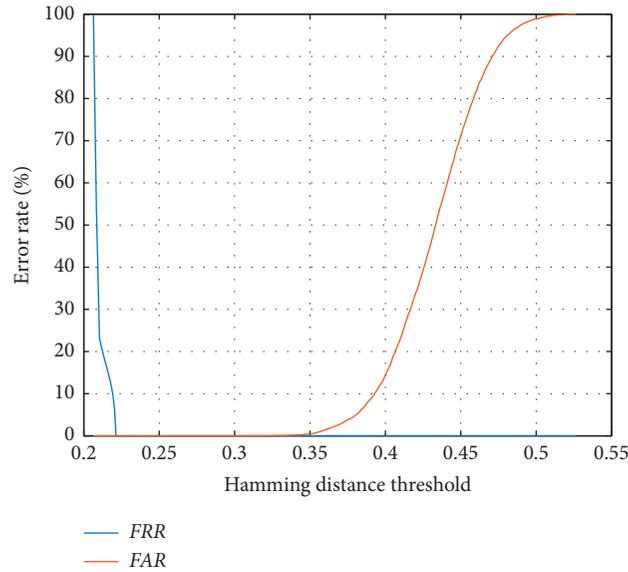


FIGURE 14: Threshold curve of FAR and FRR.

TABLE 4: A comprehensive comparison of the algorithms.

Algorithm	Intra-class	Inter-class	Indicator L	Dimension	Authentication time (s)
Original	0.1441–0.1501	0.1721–0.2681	3.6667, 0.2292	327680	0.2607
Improved	0.2054–0.2214	0.3254–0.5344	6.5000, 0.4976	40960	0.0278

has a complexity of $2 \times (N/4) \times (M/4)$. Thus, the feature coding stage has an operation complexity of $N \times M/4 + N \times M/8$. Finally, formula (11) indicates the operation complexity of the feature matching stage is $2 \times (N/4) \times (M/4)$. In summary, the total operation complexity of the entire authentication procedure is $(N \times M + 5 \times N \times M) + (N \times M/4 + N \times M/8) + N \times M/8 = 13NM/2$.

5.2. Verification of the Developed System

5.2.1. Test of Palmprint Acquisition. To demonstrate the effectiveness of the defined acquisition scheme, we acquire the palmprint samples during a random period in one day and randomly rotate the palm within a finite angle range [33]. The palmprint acquisition interface is shown in Figure 15.

TABLE 5: Comparison with some other biometrics.

Literature	Biometrics	Running time (s)	AA (%)	FAR (%)	FRR (%)
[25]	EEG	—	100.00	—	—
[26]	ECG	0.082	99.27	0.81	0.54
[28]	ECG	—	99.2	—	0.60
[27]	ECG + fingerprint	0.126	99.74	0.30	0.20
This paper	Palmprint	0.051	100	0.00	0.00

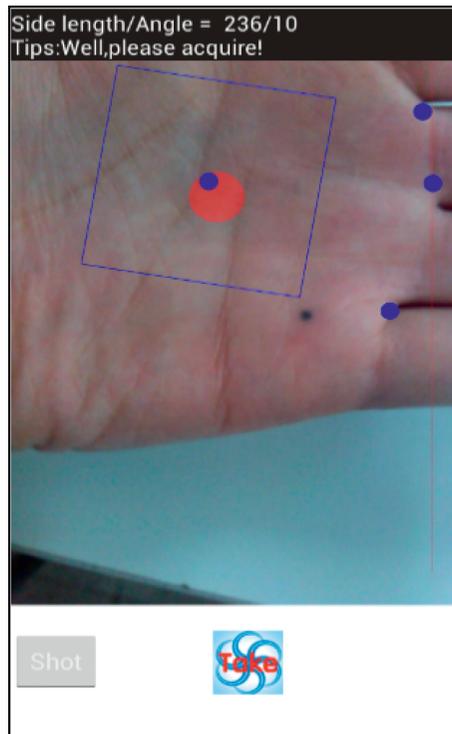


FIGURE 15: Palmprint acquisition.

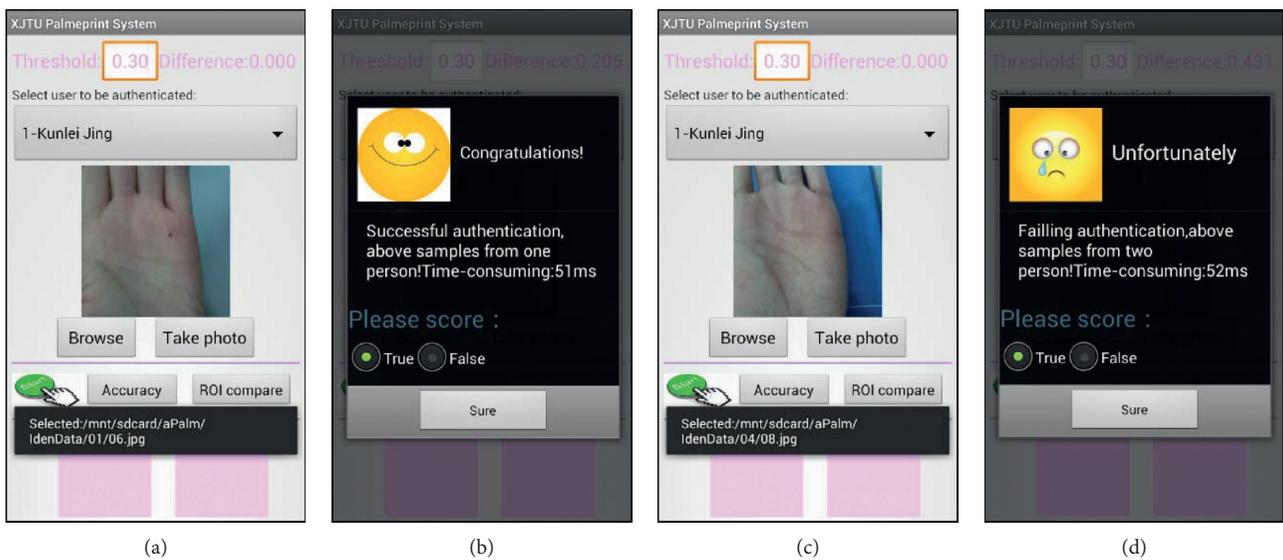


FIGURE 16: Palmprint authentication interfaces 1. (a) The acquired sample. (b) Intraclass authentication. (c) Sample selected in database. (d) Interclasses authentication.

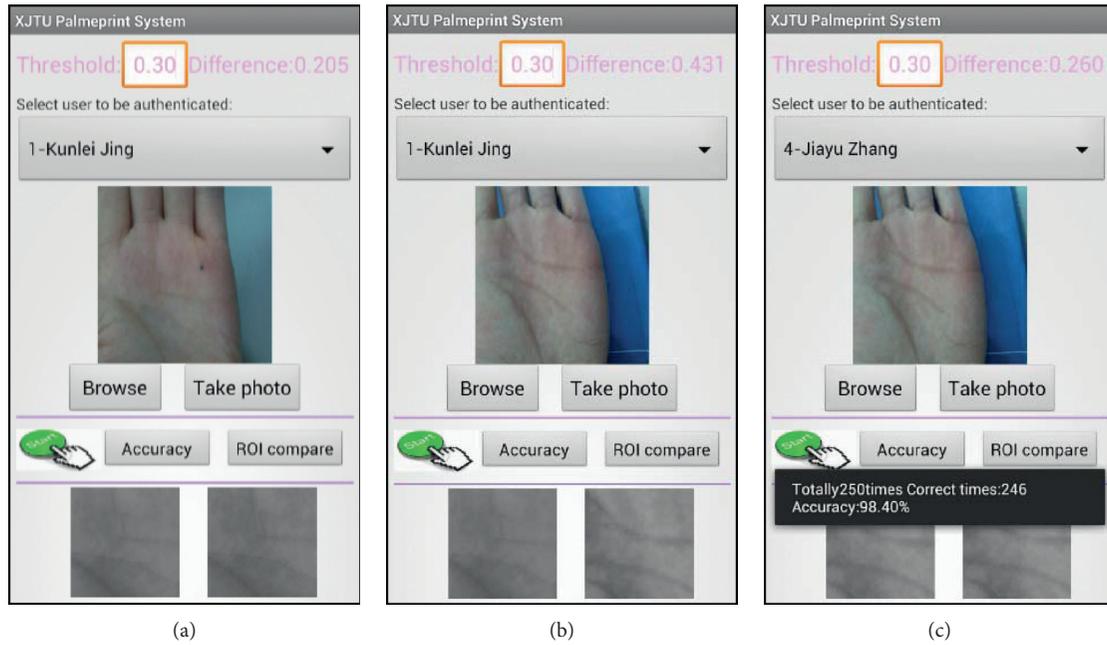


FIGURE 17: Palmprint authentication interfaces 2. (a) Intra-class ROI comparison. (b) Inter-class ROI comparison. (c) Authentication accuracy.

TABLE 6: Comprehensive comparison on different platforms.

Environment	Samples	AA (%)	FAR (%)	FRR (%)	EER (%)	Authentication time (s)
MATLAB	6000	100	0	0	0	0.028
Smartphone	500	98.40	1.60	0	—	0.051

Under the constraint of the acquisition scheme, most potential disturbances are suppressed. Meanwhile, with the conduction of the set marks, palm appears on the screen with a proper scale. After long pressing the “Take” button (about 2 seconds), key points locating and ROI segmentation are all finished. The side length and rotation angle of ROI are calculated and displayed on the top to suggest users to make the final acquisition.

5.2.2. Test of Palmprint Authentication. We recruited 500 samples acquired from 50 volunteers in the real world to verify the effectiveness of the developed system. At the beginning, each volunteer was required to register identity with 5 training samples. Then, they are required to implement 5 times intra-class real-time authentication and 5 times inter-class real-time authentication, respectively. Since the palmprint database is established in real application [34], the variations of palm posture, background, and illumination commonly exist in the samples. Here, we set the threshold as 0.30. Specifically, after selecting the identity to be confirmed from the identity list, users can click the “Take photo” button to finish real-time acquisition for the test or click the “Browse” button to choose a previously acquired sample for the test. The system will display the storage path of the

selected sample. The intra-class and inter-class authentication operations are shown in Figures 16(a)–16(d), respectively.

Users can click the “Start” button to trigger authentication. Then, the system will display authentication result and running time and encourage the user to make an evaluation. Additionally, we can check ROI of the test sample and training sample by clicking the “ROI compare” button. We can also view the total authentication times, correct authentication number, and authentication accuracy, by click the “Accuracy” button, as shown in Figure 17(c).

After a 250-time test, we find that the authentication accuracy of the developed system is close to that in MATLAB simulation. Due to the unavoidable disturbances in the acquired sample, the authentication accuracy in real-time acquisition is reduced to 98.40%. Note the authentication in MATLAB simulation takes about 0.0278 seconds for average, while it is 0.051 seconds on the smartphone. All the test results on MATLAB and smartphone are listed in Table 6.

Meanwhile, we find that if the acquisition process follows the defined constraint scheme, we can obtain more promising accuracy. Otherwise, the accuracy will be impacted. This indicates the importance of the established acquisition scheme. We also find that when the light intensity changes within a limited degree, the authentication accuracy can be

nearly not reduced, which demonstrates that our system shows robustness to some extent.

6. Conclusion

In recent years, there emerges an extreme demand for reliable identity authentication system for secure mobile transactions. This paper presents an effective and efficient palmprint authentication framework towards the application on the Android smartphone. It contains all the required functions during palmprint authentication, including palmprint acquisition, ROI segmentation, feature extraction, and the final identity authentication. More importantly, our method provides an open technology to extend the palmprint authentication algorithms to real-world applications by exploring the development of a user-friendly palmprint authentication system based on the Android smartphone, which well bridges the algorithmic study and the real-world application of palmprint authentication.

This paper presents our exploratory work of propelling the application of palmprint authentication on the mobile smart terminals. Considering the timeliness of the system, we choose the practical and effective Log-Gabor filter for feature extraction. However, when the palmprint is subjected to corruption or occlusion, the Log-Gabor filter is unable to extract robust features anymore. In the next stage, we will manage to enhance the robustness of our system concerning more complicated application scenarios. Furthermore, we will also consider achieving identity authentication using the new emerging EEG and ECG biometrics with the powerful deep neural networks.

Data Availability

The adopted public PolyU multispectral palmprint database was downloaded from <http://www.comp.polyu.edu.hk/~biometrics/MultispectralPalmprint/MSP.html>.

Disclosure

This paper is an extended version of an earlier conference paper: Research of Palmprint Authentication Arithmetic Based on Smartphone, in proceedings of the 2018 VII International Conference on Network, Communication and Computing, Taipei City, Taiwan, Dec. 14–16, 2018

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation under Grant 61673316 and in part by the project commissioned by the Sichuan Gas Turbine Research Institute of AVIC.

References

- [1] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, "Feature extraction methods for palmprint recognition: a survey and evaluation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 346–363, 2019.
- [2] S. Zhao and B. Zhang, "Learning salient and discriminative descriptor for palmprint feature extraction and identification," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [3] D. Zhang and W. Shu, "Two novel characteristics in palmprint verification: datum point invariance and line feature matching," *Pattern Recognition*, vol. 32, no. 4, pp. 691–702, 1999.
- [4] W. K. Kong, D. Zhang, and W. Li, "Palmprint feature extraction using 2-D Gabor filters," *Pattern Recognition*, vol. 36, no. 10, pp. 2339–2347, 2003.
- [5] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An online system of multispectral palmprint verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 480–490, 2010.
- [6] M. Aykut and M. Ekinici, "Developing a contactless palmprint authentication system by introducing a novel ROI extraction method," *Image and Vision Computing*, vol. 40, pp. 65–74, 2015.
- [7] X. Pan and Q. Ran, "Palmprint recognition using Gabor-based local invariant features," *Neurocomputing*, vol. 72, no. 7–9, pp. 2040–2045, 2009.
- [8] S. Fisher, F. Sroubek, L. Perrinet, R. Redondo, and G. Cristobal, "Self-invertible 2D log-gabor wavelets," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 231–246, 2007.
- [9] P. Patil, S. Kolhe, and R. Patil, "The comparison of Iris recognition using principal component analysis, Log gabor and gabor wavelets," *International Journal of Computer Applications*, vol. 43, no. 1, pp. 29–33, 2012.
- [10] J. Tao, W. Jiang, Z. Gao, S. Chen, and C. Wang, "Palmprint recognition based on 2DPCA," in *Proceedings of the 9th Pacific Rim International Workshop on Multi-Agents*, pp. 455–462, Wuhan, China, August 2006.
- [11] M. Miura, S. Sakai, S. Aoyama, J. Ishii, K. Ito, and T. Aoki, "High-accuracy image matching using phase-only correlation and its application," in *Proceedings of the 51th Annual Conference Of Society Instrument And Control Engineers Of Japan*, pp. 307–312, Akita, Japan, August 2012.
- [12] M. Franzgrote, C. Borg, B. Ries et al., "Palmprint verification on mobile phones using accelerated competitive code," in *Proceedings of the 2011 International Conference On Hand-Based Biometrics*, pp. 124–129, Hong Kong, China, November 2011.
- [13] Y. Lee, N. Kim, H. Lim, H. Jo, and H. Lee, "Online banking authentication system using mobile-OTP with QR-code," in *Proceedings Of the 5th International Conference On Computer Sciences And Convergence Information Technology*, pp. 617–716, Seoul, South Korea, December 2010.
- [14] D.-J. Kim, K.-W. Chung, and K.-S. Hong, "Person authentication using face, teeth and voice modalities for mobile device security," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2678–2685, 2010.
- [15] M. Choras and R. Kozik, "Contactless palmprint and knuckle biometrics for mobile devices," *Pattern Analysis and Applications*, vol. 15, pp. 73–85, 2012.
- [16] Y. Hao, Z. Sun, T. Tan, and C. Ren, "Multispectral palm image fusion for accurate contact-free palmprint recognition," in

- Proceedings Of the 15th IEEE International Conference On Image Processing*, pp. 281–284, San Diego, CA, USA, October 2008.
- [17] C. Li, F. Liu, and Y. Zhang, “A principal palm-line extraction method for palmprint images based on diversity and contrast,” in *Proceedings Of the 3rd IEEE International Conference On Image And Signal Processing*, pp. 1772–1777, Yantai, China, October 2010.
- [18] F. Yue, W. Zuo, D. Zhang, and B. Li, “Fast palmprint identification with multiple templates per subject,” *Pattern Recognition Letters*, vol. 32, no. 8, pp. 1108–1118, 2011.
- [19] X. Zhang, K. Jing, G. Yan, X. Xu, and W. Gong, “Research of palmprint authentication arithmetic based on smartphone,” in *Proceedings of the International Conference On Network Communication And Computing.*, pp. 242–246, Taipei, Taiwan, December 2018.
- [20] W. Wang, J. Li, F. Huang, and H. Feng, “Design and implementation of Log-Gabor filter in fingerprint image enhancement,” *Pattern Recognition Letters*, vol. 29, no. 3, pp. 301–308, 2008.
- [21] X. Gao, F. Sattar, and R. Venkateswarlu, “Multiscale corner detection of gray level images based on log-gabor wavelet transform,” *Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 868–875, 2007.
- [22] <https://www.phonearena.com/phones/>, 2018.
- [23] S. K. Kwok, J. S. L. Ting, A. H. C. Tsang, W. B. Lee, and B. C. F. Cheung, “Design and development of a mobile EPC-RFID-based self-validation system (MESS) for product authentication,” *Computers in Industry*, vol. 61, no. 7, pp. 624–635, 2010.
- [24] Z. Guo, W. Zuo, L. Zhang, and D. Zhang, “A unified distance measurement for orientation coding in palmprint verification,” *Neurocomputing*, vol. 73, no. 944–950, pp. 2359–2362, 2010.
- [25] M. Miao, W. Hu, H. Yin, and K. Zhang, “Spatial-frequency feature learning and classification of motor imagery EEG based on deep convolution neural network,” *Computational and Mathematical Methods in Medicine*, vol. 2020, Article ID 1981728, 13 pages, 2020.
- [26] M. Hammad, P. Plawiak, K. Wang, and U. R. Acharya, “ResNet-Attention model for human authentication using ECG signals,” *Expert Systems*, no. e12547, 2020.
- [27] M. Hammad, Y. Liu, and K. Wang, “Multimodal biometric authentication systems using convolution neural network based on different level fusion of ECG and fingerprint,” *IEEE Access*, vol. 7, pp. 26527–26542, 2019.
- [28] M. Hammad, S. Zhang, and K. Wang, “A novel two-dimensional ECG feature extraction and classification algorithm based on convolution neural network for human authentication,” *Future Generation Computer Systems*, vol. 101, pp. 180–196, 2019.
- [29] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [30] G. Michael, T. Connie, and A. Teoh, “Touch-less palm print biometrics: novel design and implementation,” *Image and Vision Computing*, vol. 26, no. 12, pp. 1551–1560, 2008.
- [31] J. Chen, Y.-S. Moon, M.-F. Wong, and G. Su, “Palmprint authentication using a symbolic representation of images,” *Image and Vision Computing*, vol. 28, no. 3, pp. 343–351, 2010.
- [32] A. Meraoumia, S. Chitroub, and A. Bouridane, “Palmprint and Finger-Knuckle-Print for efficient person recognition based on Log-Gabor filter response,” *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 1, pp. 17–27, 2011.
- [33] W. Li, D. Zhang, and Z. Xu, “Palmprint identification by Fourier transform,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 4, pp. 417–432, 2002.
- [34] A. Kong, D. Zhang, and M. Kamel, “Palmprint identification using feature-level fusion,” *Pattern Recognition*, vol. 39, no. 3, pp. 478–487, 2006.