A NOVEL LOW-COST IRIS RECOGNITION SYSTEM

[1]Hoang Hai Bui, [2]Jr-Jen Huang

Abstract--*Among biometric authentication methods, iris recognition has always received special attention from researchers over the world besides face and fingerprint recognition. However, most of the available works still lack practicality due to various reasons, mainly caused by the complexity of the methods themselves, the overall cost, andnot considering large scale deployment. In this research, we aim to build an iris recognition system consisting of multiple devices working remotely with reasonable prices and ready-to-use. Raspberry Pi 3 Model B+ is utilized as the core of hardware components. Following that idea, we designed an optimal image processing algorithm in OpenCV/C++ for the platform, and Python is chosen for application development in this project. Communication in the system's network is supported by lightweight messaging protocol MQTT. Experiments show satisfying results in both terms of accuracy and execution time, which stimulates us to keep improving the performance of the prototype in the future.*

Keywords-- *Biometric identification, Iris recognition, Low-cost, Remote system, Automatically capturing, Raspberry Pi, OpenCV, Daugman algorithm, MQTT.*

[1]*Dept. of Electronic Engineering, Ming Chi University of Technology, New Taipei City, Taiwan, hoanghai1246.hust@gmail.com*
[2]*Dept. of Electronic Engineering, Ming Chi University of Technology, New Taipei City, Taiwan, jrjen@mail.mcut.edu.tw*

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

I. INTRODUCTION

Biometric authentication is recognizing people's identity by studying extracted distinctive features from the human's body like face, fingerprint, iris (Physiological biometrics) or their actions like the signature, gesture (Behavioral biometrics)[1]. Among these features, the iris is considered as the most reliable organ to distinguish different people thanks to unique and stable characteristics of iris texture through lifetime [2], which has been proven in practice by the National Physics Laboratory, UK [3]. In the 1990s, Daugman developed an effective algorithm to detect, process, and estimate the similarity degree between two iris images [4]. Later his work was introduced widely to the public in 2004 [2] and it has inspired a lot of researchers to follow similar mechanisms. These methods and many prior iris recognition works are often referred to as traditional iris recognition.

The year 2012 marked an important event in the computer vision field when a team won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contest in that year by adopting the convolutional neural network (CNN). Since then, Artificial Intelligence(AI)hasdrastically become the dominant trend in image processing related researches, including iris recognition [5][6].Nonetheless, the ultimate weakness of neural networks is computational complexity.Another problem is the lack of iris data for network training [7]. Furthermore, most of up to date studies about iris recognition, either by traditional methods or AI approaches, often emphasize theoretical performance more than practical effectiveness due to complexity. The expensive and tiresome developing process is the main reason making the cost of existing iris recognition devices on the market still remain in pricey rance and is less affordable than other solutions like fingerprint and face recognition as reported in work [8].

Hence, we initialized this project to develop a new low-cost iris recognition system and tackle some problems which haven't been solved by similar prior works. Raspberry Pi 3 Model B+ with several required qualities is chosen for this task. The single-board computer has an adequate processing unit with frequency up to 1.4Ghz, 1GB RAM, can be connected with various types of peripherals including Pi Camera – a compatible type of camera for the platform [9]. On top of that, it is designed for Internet of Things (IoT) applications [10] with self integrated Ethernet port and Wifi module. Overall, this systemconsists of multiple iris recognition devices working remotely. All units are controlled by a central computer through a Local Area Network (LAN), which allows large scale deployment like at a national border or an airport [11].Each device is constructed by a Raspberry Pi 3B+, a Pi Camera for sample capturing, and a couple of necessary components with a total cost of only 90$. With this limited hardware resource, our machine still able to accomplish more than 90% recognition accuracy and around1 second of execution speed.

The remainder of this paper is organized as follows. Related works are reviewed in Section 2. Section 3 focuses on our method to develop the system. Experimental results and discussions are provided with details in Section 4. Finally, Section 5 gives remarkable conclusions from our work.

II. RELATED WORKS

Despite there have beennumerous researches related to iris recognition with ANN, none of them was implemented on a single-board platform like the Raspberry Pi. With the maximum processing speed of only around 1GHz, these boards seem to be incapable of doing terrific classifying tasksusing neural networks in real-time. Monteiro *et al.* reported that their five layers CNN deployed on Raspberry Pi 3B consumes up to 8 seconds

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

to process a single frame for a simple structural health monitoring, with the training stageusing 144 samples taking 90 minutes [12].

On the other hand, Raspberry Pi is quite preferred to cut down on the budget for an iris recognition devicein a combination with the classical approach. Cruz *et al*. [8], Lavric*et al.* [13], and Kunik*et al.*[14] all followed the same key ideato deal with the price issue. However, the main limitation of these works is the authors only consider an authentication system with solely a device. That being said, in a large place with multiple entrances, each recognition device at each entrance works as an independent system. Registering or removing users would require to be done directly at every location, which makes the task seems to be impractical. On the other hand, while work [13] not describe in detail their capturing method, the solutions proposed by studies [8] and [14] to acquire high-quality iris samples also cause disadvantages to their machines. Lavric*et al.* demand accessing subjects to stick their eye to a large pipeline for agood iris image. In order to put the user's head at a fixed distance and angle to the camera for decent image resolution, Kunik*et al.* utilize the motorized table with the forehead and chin rest frame. These apparatuses increase the size of the system significantly and would be inconvenient to implement in practice. Thus, we aim for better solutions, providing compact size, comfortable interaction, as well as handy user management with the proposed authentication system.

III. PROPOSED METHOD

1. Recognition Device's Hardware and Sample Acquisition

Our recognition device is built from Raspberry Pi 3B+ as the main component for developing all of the software programs and installing other hardware elements. The first one is the Pi Camera, which takes responsibility for iris sample acquisition. Human iris shows rich patterns under near-infrared (NIR) light [2], therefore the camera module is assisted with ten 940nm light-emitting diodes (LED) installed around to capture iris images under NIR wavelengths. 25mm CCTV lens is adopted in order to take user's samples from a distance of around 30cm to achieve good resolution iris images. For the camera type option, the 5MP PI IR CUT module from the Pi Camera family is selected, which supports the non-IR filter mode. In front of the camera, we put an additional infrared-passband filter to limit the amount of visible light reflection in captured images. In our experiment, the standard image dimension is 640x480 pixels.

Following the idea of a mechanism capturing the user's iris sample automatically in mind, we install near the camera an HC-SR04, which is a low-cost ultrasonic sensor for distance measuring. When the gap measured is smaller than 40cm, the camera records an image each time the ultrasonic sensor returns a distance value.The program then searches for circles inthe image, which is the shape of the pupil, using Canny Edge Detection and Hough Circles algorithms [14]. If the measured distance lies between a pre-defined range and exactly one circle presenting the shape of the iris is detected in the image, the corresponding image will be passed to the subsequent processing stages. The defined distance range guarantees a good image resolution. For system state indication, we use 3 5mm color LEDs. *Fig.1* presents a finished prototype of the device for iris recognition.

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

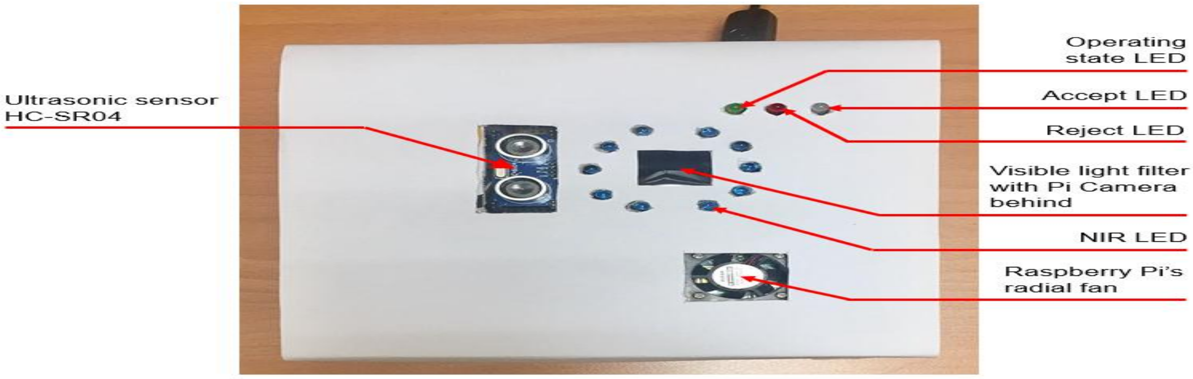

Ultrasonic sensor HC-SR04
Operating state LED
Accept LED
Reject LED
Visible light filter with Pi Camera behind
NIR LED
Raspberry Pi's radial fan

Figure 1: The completed iris recognition device

2. Iris Recognition Process

The sample image acquired from the previous stage is then processed using the OpenCV library and C/C++ language for fast image processing ability[14]. Ouroverall algorithm based on Daugman's classical iris recognition approach includes 6 stages:Iris pre-processing, iris segmentation, iris normalization, feature extraction, iris masks generating, and template matching. Each step is implemented with a suitable method considering theaverage computational capability of the Pi.

2.1.Pre-Processing

Before dealing with the iris area, the sample image needs some quick modifications to improve the performances of the following steps. The Pi Camera can only capture color images, therefore we first convert it to grayscale format to reduce processing complexity. The following step is noise filtration. External noises caused be dust or stain can be effectively removed by applying a Gaussian filter [15].

2.2.Iris Segmentation

In this stage, the iris area is segmented from pupil and sclera by detecting the iris boundary and the pupil boundary. Since iris usually be partly occluded by upper and lower eyelids, they also need to be detected and excluded them from iris codes comparison.

We initialize the segmentation process by detecting the center and radius of the pupil boundary firstconsidering its simpleness in appearance compared to iris and eyelid. After thresholding, the grayscale image into a binary image with an appropriate threshold, the OpenCV functionFindContours[16]is then applied and easily detects the contour of a block of black pixels which is the pupil. From the detected contour, the program calculates the radius and the center's coordinates for a circle which well represents the contour. Techniques Opening [17] and Closing [18] are additionally applied after thresholding the image to improve the result.

Regarding iris outer boundary detection, using the Integro-Differential Operator presented inDaugman's work[2] would return a very precise outcome, but this algorithm iteratively searches for a circle's parameters pixel by pixel through the whole image and consumed a lot of time. Inspired by the idea proposed by Li *et al.* in work [19], we create a new approach from the original Integro-Differential Operator to significantly decrease the execution time. We utilize the Hough Circles algorithm to first detect a small number of circles that have the potential to be the boundary we need to find. After that, the average pixel intensity derivative is calculated on

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

each circle. As the sclera is much brighter than the iris, the circle resulting in the highest average intensity difference is recognized as the iris outer boundary.

From the parameters of the pupil's boundary, an ROI can be created to find the parts of the eyelids which impede the iris. For simplicity, this time Hough transform for lines [20] is implemented on that ROI. The algorithm detects lines that possibly run along the eyelids, then the regions formed by these lines and the ROI are masked with black pixels.Aniris segmentation result exampleis demonstrated in *Fig.2*.

(a) (b)

Figure 2: Iris segmentation's result: (a) iris inner and outer boundaries, (b) eyelid mask

2.3.Iris Normalization

For the ease of post-processing as well as creating a template for comparison, the segmented iris region formed between two circles in the Cartesian coordinates system (x, y) will be transformed into a rectangle shape in Polar coordinates system (r,θ). The method is based on the rubber sheet model proposed by Daugman[2] and is illustrated in *Fig.3*.

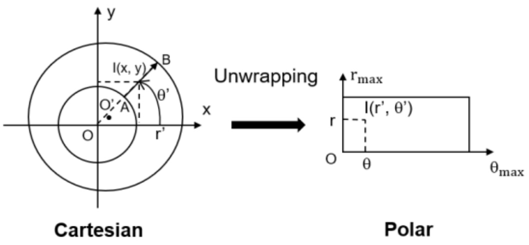


Figure 3: Cartesian-Polar coordinates system transformation map

In our experiment, we select a template size of 240x40 pixels, which is adequate to represent the most unique information of the iris. By solving mathematical problems related to this model, including the internal and external boundaries are usually not concentric, the formula for transformation I(x,y) → I(r, θ) is derived as follow:

$$\begin{cases} I(r,\theta) = I\big(x(r',\theta'), \quad y(r',\theta')\big) & (1) \\ x(r',\theta') = (R_{pupil} + r') \times \cos\theta' & (2) \\ y(r',\theta') = (R_{pupil} + r') \times \sin\theta' & (3) \end{cases}$$

with:

$$r' = |AB| \times \frac{r}{r_{max}} \qquad (4)$$

$$\theta' = 2\pi \times \frac{\theta}{\theta_{max}} \qquad (5)$$

$$|AB| = |\, dx\cos\theta' + dy\sin\theta' \pm \qquad (6)$$

$$\sqrt{{\mathrm{R}_{\mathrm{Iris}}}^{2} - (dx\sin\theta' - dy\cos\theta')^{2}} - \mathrm{R}_{\mathrm{pupil}} \mid$$

$$dx = \mathrm{x}_{0'} - \mathrm{x}_{0} \qquad (7)$$

$$dy = \mathrm{y}_{0'} - \mathrm{y}_{0} \qquad (8)$$

2.4.Feature Extraction

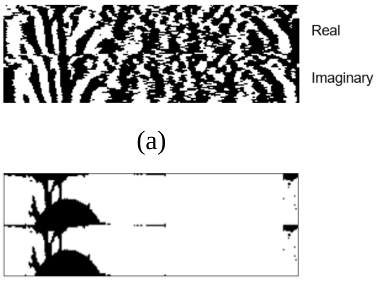
From the obtained normalized iristemplate, distinctive features of the iris need to be encoded to become the figure for comparison. Based on the suggestion given by Libor Masek[21] we choose to implement Gabor 1D Log filter for our model:

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right) \qquad (9)$$

We use the optimal parameters of center frequency f_0 = 1/18 and bandwidth σ = 1/36 fromMasek's work [21] for the filter. The filtration step is implemented in the frequency domain. 1D Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) are the techniques for converting forth and back every row of the iris sheet between spatial domain and spectral domain. The frequencies of the textures are represented in complex format, which is then convolved with the Gabor filter using the OpenCV function mulSpectrums(). After being transformed back to the spatial domain and binarized, the result is 2 matrices of real and imaginary components corresponding to the iris code of the input iris sample.

2.5.Iris Mask Generating

Because the iris region may be impeded by eyelids and eyelashes, it is necessary to mask these obstacles and eliminate them from matching two iris codes. We find the eyelashes by first applyingthe Dynamic Binarization technique [17] to the iris image in the Polar coordinates system. After also converting the eyelid mask derived from the segmentation stage to the Polar dimension and binarizing it, the mask for comparison is produced by combining the two previous masks using logical AND operation. *Fig.4* presents an example of iris code and comparison mask.



(a)

(b)

Figure 4: (a) Iris code, (b) Iris mask

2.6.Template Matching

A subject is qualified to be accepted or rejected by comparing his iris code with all iris samples in a model database to find the degree of dissimilarity.The Hamming Distance (HD) method[2] is chosen to compute the dissimilarity between two iris codes:

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

$$HD = \frac{||codeA \otimes codeB \cap maskA \cap maskB||}{||maskA \cap maskB||} \qquad (10)$$

The idea of this metric function is measuring the number of different bits between two code matrices on the total number of bits taken into comparison after masking. The more identical the two iris code are the smaller the HD score is.

In order to deal with the sample's rotational inconsistency caused be tilting the subject's head, one set of iris code and mask in the matching phase will be shifted to the left and right directions by maximum 20pixels each using a bit-wise operation. The shifting position resulting in the smallest HD value will be recognized as the comparison result of the pair.To further cut down on both execution time and error rate, we employ the TripleA solution proposed by Rathgeb et al. [22]. The optimal shifting size for the first step in the method is $s = \sqrt{K}/\sqrt{2} \approx 4$ [22], with $K = 12 * 2 + 1 = 25$ is the bit shifting range.

After comparing the subject's iris code with all available samples, giving recognition result is based on deciding a reject threshold. If the smallest HD result among all comparisons smaller than the threshold, the program returns a True value which means the subject is recognized as a genuine user and is accepted. Whereas, a False result when the minimum HD is greater than the threshold implies that the subject is rejected as an impostor.

3. Multi-Devices Architecture

For a scenario of multiple authentication devices working together, a communication protocol is required to control all of these devices from a single computer. Based on a comprehensive survey for IoT technologies conducted by Al-Fuqaha et al. [23], we choose to adopt Message Queue Telemetry Transport (MQTT),a Machine-to-Machine messaging protocol in low bandwidth environments. The basic principle is the publish/subscribe pattern in a network consisting of three components: Broker, subscriber, and publisher [23].

In our project, one host machine will be both the broker and the publisher. The broker's role is executed by utilizing "mosquito", a well-known broker service well-supporting Debian-based Linux platforms. The broker after being initialized will establish a topic dedicated tothe iris recognition system that it takes charge of. All of the recognition devices after being started will automatically subscribe to that topic through a LAN. The proposed architecture for a recognition device is illustrated in *Fig.5*.

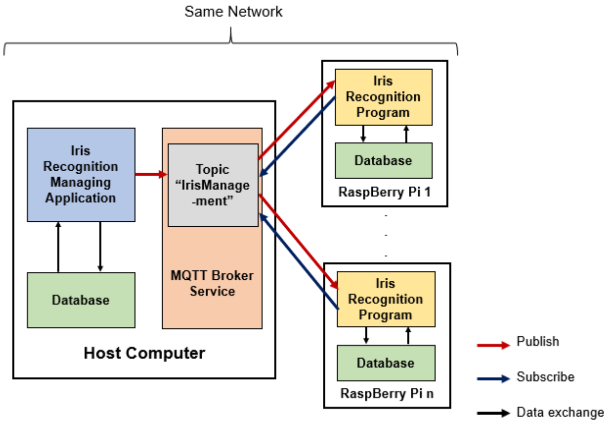


Figure 5: The architecture of the proposed system

Each time a new user requests for enrolling in the system, his iris sample will be acquired througha management softwareon the host machine. After that, the program will processthe sampleto obtain iris code and iris mask,then assign an ID to them and publish a message containing the obtained information to the broker.When the subscribers receive a message indicating that a new user needs to be registered, they will process the rest of the message then save the relevant data to the device's database. The procedure to withdraw a user from the system is also doneusing the same software. This time it will publish a message which contains the ID of the user chosen to be removed, whose relevant data will be deleted by the receiving devices.

IV. Experimentsand Results

1. Experiments Setting

Our device is examined with an iris dataset containing samples from 50 volunteers. The procedure for image acquisition is performed with our proposed mechanism. 1050 samples are collected in total with 21 entries per subject. Corresponding each volunteer we select a sample from his dataset for system training. The rest of the picture set is treated as testing samples.

The proposed recognizing algorithm is trained with the training set to find proper setting parameters. Subsequently, the fine-tuned software is then put into the examination with the training set is now treated as a database. Every sample is tested in two scenarios of the user's identityas a genuine subject and as an impostor. In total there are 2000 trials with 1000 cases per scenario. From the results, the system is evaluated based onaccuracy rate and execution time.

2. Iris Samples Classification Result

There are four types of outcome may appear in a recognition process: (1) True Accept (TA): The subject is in the database and is accepted, (2) False Reject (FR): The subject is in the database but is rejected, (3) True Reject (TR): The subject is not in the database and is rejected, and (4) False Accept (FA): The subject is not in the database but is accepted. The Receiver Operating Characteristic (ROC) curve which reflects the effectiveness of a classifying mechanism in terms of the tradeoff between False Reject Rate (FRR) and False Accept Rate (FAR) when adjusting the reject threshold [24] for our system is shown in *Fig.6*. Two attributes of the curve True Positive Rate (TPR) and False Positive Rate (FPR) are calculated as:

$$\text{TPR} = 1 - \text{FRR} = \frac{\text{No. of TA}}{\text{No. of TA} + \text{No. of FR}} \text{ x } 100\% \qquad (11)$$

$$\text{FPR} = \text{FAR} = \frac{\text{No. of FA}}{\text{No. of FA} + \text{No. of TR}} \text{ x } 100\% \qquad (12)$$

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

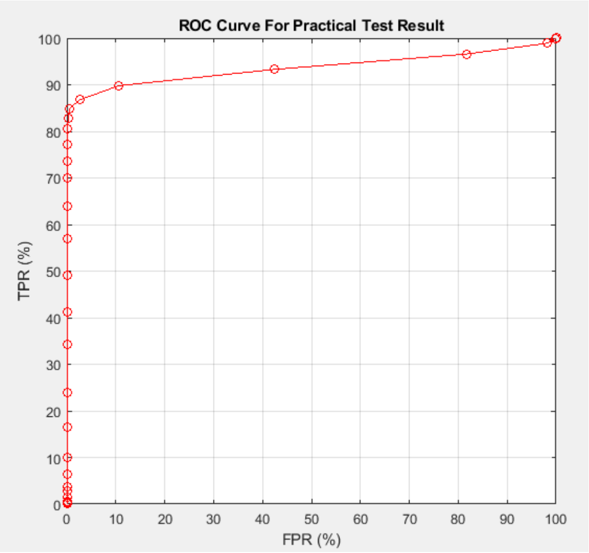


Figure 6: ROC Curve of the test result

The objective of a biometric identifying task is to reduce both the FRR and FAR as small as possible. In practice when FRR decreases to a certain rate the FAR also climbs up drastically. Due to inaccuracy in the iris processing process, there are cases where inter-class iris codes comparisons can result in Hamming distances as high as some instances in intra-class matching [24]. In contrast, the operation in template matching may coincidentally give an inter-class pair an HD score as low as that for the bit-wise shifting samples from the same class. The graph in *Fig.12* shows 2 domains of HD distribution in our experiments for all of the inter-class and intra-class comparisons. As can be seen clearly from thechart, there is an overlapping region between the two domains, which creates false reject and false accept results depending on the selected reject threshold.

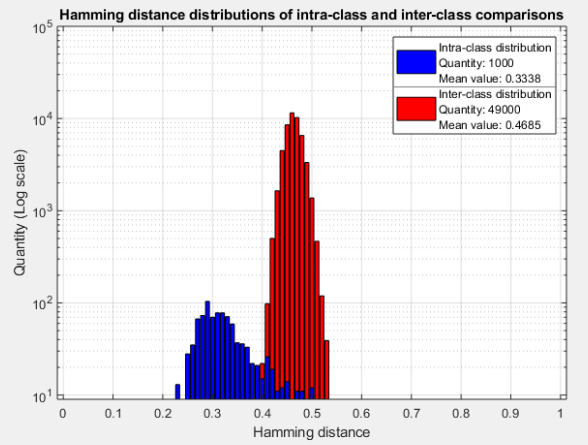


Figure 7: Hamming distance distribution

The fact that the observed shared region between inter-class and intraclass distributions is relatively small means this classifier can achieve a tolerable accuracy rate. Consequent result is the ROC curve starts to bend noticeably only after the TPR reaching around 85%, then it gradually draws toward the top right corner showing an adequate separation between the two classes. The main source that contributes to the misclassification is our algorithm suffers from several iris subjects superimposed by eyelids, eyelashes, or even having a lower image resolution than the model samples of the same class in the hypothetical database.

Deciding a reject threshold depends on the desire of the error rate tradeoff, which is varied between different security tasks. *Fig.8* illustrates the accuracy rate combined from TPR and TNR versus different reject thresholds.

The highest total accuracy value experienced in our experiments is 92.2% at a threshold of 0.40, corresponding to a TPR of about 85% and a TNR of approximately 100%, which is acceptable for practical applications.

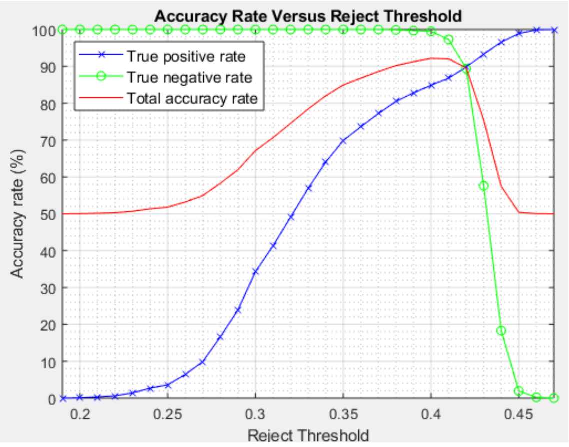


Figure 8: Recognition accuracy rate versus reject threshold

3. *Execution Time*

Finally, our recognition device is examined with the processing speed in different aspects. *Table 1* presents the testing results of the highest, smallest, and average values for iris code generating, template matching, and combined execution time. The mean execution time for the whole process is only about 0.9s, which is fairly quick and much faster than the result reported in work [14], where the minimum iris codes comparison time for a dataset of 233 images is above 13s in the same platform.Our average time for template matching occupies only about 0.13s since we apply the *TripleA* approach to the procedure.

Table 1: Execution time records of the recognition device

Factor	*Time (s)*
Minimum value for iris code generating time	*0.4516*
Maximum value for iris code generating time	*2.0049*
Mean value for iris code generating time	*0.7943*
Minimum value for iris codes comparing time	*0.1327*

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020

Maximum value for iris codes comparing time	*0.3148*
Mean value for iris codes comparing time	*0.1373*
Minimum value for total execution time	*0.5899*
Maximum value for total execution time	*2.1407*
Mean value for total execution time	***0.9317***

V. CONCLUSION

In this paper, we have proposed a practical iris recognition system based on Raspberry Pi 3B+, which brings benefits of affordable price and comfort in large scale deployment. The recognition device posses the ability to capture iris sample automatically, then generate iris code and carry out template matching through a refined process which overcomes the limitations of the platform. For controlling the remote machines,the MQTT protocolis employed.

Experimental results show that our algorithm can reach a maximum accuracy rate of over 92% and is particularly fit for security tasks where the system must maintain an absolute correct rejection rate. Feedback time is another strength of our device, with the average execution time for the overall process is not up to 1 second considering the single board computer is built with a core of only 1.4Ghz maximum processing speed. The key factors to improve the exactness for the system in the future are sample acquisition with stable high resolution, more accurate segmentation algorithm, and careful selection of parameters for iris pattern filter.

REFERENCES

1. F. Scotti and V. Piuri, “Adaptive Reflection Detection and Location in Iris Biometric Images by Using Computational Intelligence Techniques,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 7, pp. 1825 – 1833, 2010.
2. J. G. Daugman, “How Iris Recognition Works,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 21–30, 2004.
3. T. Mansfield, G. Kelly, D. Chandler, and J. Kane, “Biometric Product Testing Final Report,” National Physical Laboratory of UK, Issue 1.0, 2001.
4. J. G. Daugman, “Biometric Personal Identification System Based On Iris Analysis,” US5291560A, 1994.
5. T. Zhao, Y. Liu, G. Huo, and X. Zhu, “A Deep Learning Iris Recognition Method Based on Capsule Network Architecture,” *IEEE Access*, vol. 7, pp. 49691–49701, 2019.
6. K. Wang and A. Kumar, “Toward More Accurate Iris Recognition Using Dilated Residual Features,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 12, pp. 1670–1684, 2019.
7. K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, “Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective,” *IEEE Access*, vol. 6, pp. 18848–18855, 2018.
8. P. Lavrič, Ž. Emeršič, B. Meden, V. Štruc, and P. Peer, “Do it Yourself: Building a Low-Cost Iris Recognition System at Home Using Off-The-Shelf Components,” presented at the International Electrotechnical and Computer Science Conference ERK 2017, 2017.

9. P. J. Basford *et al.*, "Performance Analysis of Single Board Computer Clusters," *Future Gener. Comput. Syst.*, 2019.

10. R. Harshitha and M. H. S. Hussain, "Surveillance Robot Using Raspberry Pi and IoT," presented at the 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 2018.

11. A. Gangwar and A. Joshi, "DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition," presented at the 2016 IEEE International Conference on Image Processing (ICIP), 2016.

12. A. Monteiro, M. de Oliveira, R. de Oliveira, and T. da Silva, "Embedded application of convolutional neural networks on Raspberry Pi for SHM," *Electron. Lett.*, vol. 54, no. 11, pp. 680–682, 2018.

13. F. R. G. Cruz, C. C. Hortinela, B. E. Redosendo, and B. K. P. Asuncion, "Iris Recognition Using Daugman Algorithm on Raspberry Pi," presented at the 2016 IEEE Region 10 Conference (TENCON), 2016.

14. Z. Kunik, A. Bykowski, T. Marciniak, and A. Dąbrowski, "Raspberry Pi based complete embedded system for iris recognition," presented at the 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2017.

15. P. Shopa, N. Sumitha, and P. S. K. Patra, "Traffic Sign Detection and Recognition Using OpenCV," presented at the International Conference on Information Communication and Embedded Systems (ICICES2014), 2014.

16. S. Rayhan Kabir, M. Akhtaruzzaman, and R. Haque, "Performance Analysis of Different Feature Detection Techniques for Modern and Old Buildings," presented at the 3rd International Conference on Recent Trends and Applications in Computer Science and Information Technology, 2018.

17. J.-J. Huang, T. Yu, and Y.-J. Chen, "The Front Vehicle Distance and Time to Collision Warning System," *Int. J. Concept. Manag. Soc. Sci.*, vol. 2, no. 3, pp. 36–41, 2014.

18. N. A. Othman, M. U. Salur, M. Karakose, and I. Aydin, "An Embedded Real-Time Object Detection and Measurement of its Size," presented at the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018.

19. P. Lia, X. Liu, and N. Zhao, "Weighted Co-occurrence Phase Histogram for Iris Recognition," *Pattern Recognit. Lett.*, vol. 33, no. 8, pp. 1000–1005, 2012.

20. J.-J. Huang, J.-R. Wu, and Y.-J. Chen, "A Real-time Lane Departure Warning System Based on TI-DM6437," *Int. J. Concept. Manag. Soc. Sci.*, vol. 2, no. 3, pp. 31–35, 2014.

21. L. Masek, "Recognition of Human Iris Patterns for Biometric Identification," The University of Western Australia, Australia, 2003.

22. C. Rathgeb, H. Hofbauer, A. Uhl, and C. Busch, "TripleA: Accelerated Accuracy-preserving Alignment for Iris-Codes," presented at the 2016 International Conference on Biometrics (ICB), 2016.

23. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 4, pp. 2347–2376, 2015.

24. J. G. Daugman, "New Methods in Iris Recognition," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 37, no. 5, pp. 1167–1175, 2007

25. Yasameen K. Al-Majedy, Abdul Amir H. Kadhum, Ahmed A. Al-Amiery, Abu Bakar Mohamad. "Coumarins: The Antimicrobial agents." Systematic Reviews in Pharmacy 8.1 (2017), 62-70. Print. doi:10.5530/srp.2017.1.11

26. Kak, S.Probability constraints on the classical/quantum divide(2013) NeuroQuantology, 11 (4), pp. 600-606.

27. Grandpierre, A., Chopra, D., Doraiswamy, P.M., Tanzi, R., Kafatos, M.C.A multidisciplinary approach to mind and consciousness(2013) NeuroQuantology, 11 (4), pp. 607-617.

DOI: 10.37200/IJPR/V24I4/PR201035
Received: 22 Oct 2019 | Revised: 13 Sep 2019 | Accepted: 15 Jan 2020