A Comparative Analysis Between the Performance of the Extracted Features of JPEG and PNG on a Raspberry Pi Iris Recognition System

Gio Arturo S. Martinez[1], Reese Elijah B. Moralde[2], Noel B. Linsangan[3], Rayellee Myrtle Laire Ang[4]

School of Electrical, Electronics, and Computer Engineering, Mapua University
Intramuros, Manila, Philippines

[1]gasmartinez@mapua.edu.ph
[2]rebmoralde@mapua.edu.ph
[3]nblisangan@mapua.edu.ph
[4]rmlang@mymail.mapua.edu.ph

***Abstract*— The image file format used in iris recognition systems has been proven to affect the performance of iris authentication. PNG and JPEG, the image file formats that are among the most common, will be compared in this study by converting a single iris image to the two image file formats, which will then be evaluated using total and average processing time FRR and FAR. The two file formats differ in their used image compression algorithms. JPEG achieves a lossy image compression, which results in irreversible data loss after every compression, using DCT (Discrete Cosine Transform). On the other hand, PNG achieves lossless image compression using the DEFLATE algorithm. Both algorithms' motive is to compress data. The latter, however, executes this without losing data. The image file formats are compared on a Raspberry Pi 4B that runs the iris recognition system utilizing Daugman's algorithm. After conducting 80 trials of iris recognition in the system, the JPEG database resulted in a total and average processing time of 70.8301s and 1.7708s respectively, and a FRR of 0.05 and a FAR of 0.05. Meanwhile, the PNG database yielded a total and average processing time of 72.2576s and 1.8064s respectively, and a FRR of 0.10 and a FAR of 0.05. This comparative study identified that the extracted iris features from JPEG produced better recognition performance results in the implemented iris recognition system over PNG.**

Keywords*— *Iris recognition, JPEG, PNG, Daugman Algorithm, Raspberry Pi

INTRODUCTION

Biometrics can be described as a form of authentication that works on human attributes such as the iris, face, fingerprints, and voice [1]. Recent studies have even proposed authentication systems for tongue print [2] and palm veins [3]. In line with this, Biometrics are not limited to security purposes. A study by J. D. A. Villarama et al. developed finger biometrics that is automated to track attendance. Instead of the usual logging method by pen or keyboard for attendance, fingerprints would be scanned by a scanner, and the employees' payroll would be calculated.[4]. Among these, the iris biometric has both high universality and stability in biological individuality [5]. It has been stated that the general flow of image processing starts with image acquisition, enhancement, segmentation, feature extraction, and finally, representation and recognition [6], such as the Daugman Algorithm. The segmentation begins by locating the outmost diameter of the iris, and then a line is drawn for each point of the diameter; the drawn line should resemble a ring. The features are then extracted using Gabor filters onto the normalized iris images. Finally, encoded iris images are produced from feature extraction and will be compared using the Hamming Distance. [7]. Fundamentally, several studies have proposed improvements in the specific steps of iris recognition to upgrade the accuracy of an IR's (iris recognition) system. Among these, improvements in feature extraction stand out as it presents solutions to the problems of an iris recognition system, which is the extraction of redundant features. Redundancy increases memory consumption, and it also results to increasement in computational complexity [8]. Although there are various papers that cover the improvements in an iris recognition system itself, it is observed that there is a scarcity of studies that tackle the effectiveness of the type of image file format used in iris recognition. Several image file formats use either lossy or lossless compressions. Two of these are also considered the most prominent image file formats: Portable Network Graphics Format (PNG) and the Joint Photographic Experts Group (JPEG) [9]. PNG is a raster graphics file format, and it is commonly used for its support for lossless image compression. The compression process for PNG has two stages: First is the pre-compression which uses a single filtering method for prediction. Second is the compression using the non-patented lossless data compression algorithm known as DEFLATE [10]. DEFLATE is a compressed data algorithm that consists of a succession of blocks that works as input data blocks. A combination of both the LZ77 algorithm and the Huffman coding algorithm compresses each input block. It looks for redundant input strings (strings denote arbitrary byte sequences). Then, if the redundant string is found, a pointer that points to the preceding string replaces it that contains the distance and its length. The matched lengths and distances are each assigned to a Huffman tree for compression [11]. On the other hand, JPEG is also a popular compression standard, and it can adjust its compression, balancing image quality and file memory size. This standard uses the Discrete Cosine Transform (DCT) for acquiring lossy compression. The three core steps of JPEG lossy compression are DCT transformation, Coefficient Quantization, and Lossless compression. A discrete cosine transform (DCT) defines a set of fixed data points regarding sums of cosine functions fluctuating at different frequencies. It is a technique for separating a picture into sections or "spectral sub-bands" of varying relevance in terms of visual quality. To transfer signals to the frequency domain from the spatial domain, DCT can be used. [10], [12]. The JPEG lossy compression causes the reduction of file size while preserving image quality (at

least to human vision) by eliminating the redundancies and the least relevant image data. [9] [13]. The Iris recognition system developed in this study will be implemented on Raspberry Pi. There is much research regarding what systems Raspberry Pi can support. The single-board computer Raspberry Pi has been proven to be powerful enough to make computations for applications in image and video processing. Raspberry Pi has been used to process images for classifying Feline Epidermal Disease [14], identifying bacterial leaf blight from rice grains [15], classifying defects in Robusta Green coffee beans [16], and recognizing four species of snails [17]. It has also been used in video processing for recognizing Filipino Sign Language (FSL)[18], calculating vehicle speed [19], determining pupil diameter [20], and detecting Filipino Food and its distance from the camera source to help the visually impaired [21]. The studies mentioned have observed similarities with Iris recognition systems. All of which possess a camera module and a Raspberry Pi. With the studies mentioned, ample information regarding the capabilities of a Raspberry Pi has been deduced.

Many studies contribute to the improvement of iris recognition systems. Infrared light, indubitably, is a data acquisition tool if done under specific and correct conditions. NIR Spectroscopy and other techniques determined a fruit's sugar content in a non-destructive way [22]. Parts of iris images may contain Haar-like features; therefore, they can undergo Haar wavelet processing [23]. As such, a study by N. B. Linsangan et al. utilized a group of wavelets: Biorthogonal, Reverse-Bior, Haar, and Daubechies, for iris template encoding. Furthermore, the study's researchers determined that Haar was the most accurate algorithm [24]. Another study used a deep learning approach, in which case they proposed an iris recognition framework based on the transfer learning approach [25]. A study by N. B. Linsangan et al. developed an iris recognition device implemented on Raspberry Pi that uses the Daugman algorithm for biometric verification. One of the notable recommendations in the study is to have additional preprocessing methods for removing undesired features and enhancing salient features of the image [7]. Additionally, a study by H. Hofbauer et al. showed that the compression algorithm applied to an image is important because they have concluded that JPEG and JPEG200 are inferior candidates for image compression in iris recognition; BPG is more suitable. A similar study that focused on image compression of iris images by A. Paul et al. mentioned that compression removes redundancy (redundant signal sources) and omits irrelevant pixel values [26], which can be a factor in attaining higher accuracy for iris recognition systems. This supports the idea that the image file formats and the compression algorithm used on the image would make the iris recognition system's feature extraction and image processing more accurate.

Most studies on iris recognition and iris feature extraction discussed used only one database and one image file format. Along with this, the same studies disregarded the effect of the images' file format or compression technique on their iris recognition system. Although there exists a study that used BPG and compared them with JPEG and JPEG 2000, the use of BPG file format technology is quite uncommon. On the other hand, JPEG and PNG are among the image file formats patronized the most, which means that the image file formats will be universally supported. Furthermore, most previously mentioned papers [1], [8], [20] used recognition rate as their only performance metric or their studies, which only cover correctly recognized inputs, disregarding false positives and negatives. False Reject Rate or FRR and False Accept Rate or FAR are the two performance metrics to be used for the evaluation of the performance of JPEG and PNG in terms of iris feature extraction and recognition.

The study's purpose is to distinguish which PNG and JPEG produce better recognition performance in an IR system that uses Daugman's Algorithm, using metrics: False Acceptance Rate and False Rejection Rate. In line with this are the two specific objectives, namely: (1) to present JPEG and PNG's FAR and FRR difference; (2) To be able to deploy an IR (Iris Recognition) system utilizing Raspberry Pi computer that uses the Daugman algorithm. (3) To measure the iris authentication processing time difference between JPEG and PNG in the mentioned IR system implementation.

This study is beneficial for iris recognition researchers. This research aims to distinguish which image file format used in this paper is a better candidate as input data for feature extraction. Hence, the findings in this paper can help researchers with what iris database to use or what image file format is more appropriate for their study. Furthermore, this paper can serve as a reference for constructing a low-scale iris recognition system that has limited memory resources.

This study works on eyes that do not have any external physical deformation. The image file formats that will be compared in this study are only JPEG and PNG. False Reject Rate or FRR and False Accept Rate or FAR are the two performance metrics to be used to evaluate the study. The IR system's algorithm would be Daugman Algorithm, deployed in Raspberry Pi version 4B. The prototype's IIT Delhi database [27] and camera module are the two sources of images used in this study. IIT Delhi database iris images are originally in BMP file format. Conversion of these images to JPEG and PNG would occur so that there would be a database for each file format.

METHODOLOGY

Hardware Development

The system block diagram of the study is shown in Figure 1. All iris information will come from iris scans from the camera and the iris images from the IIT Delhi iris database. The Raspberry Pi will handle all processes, such as preprocessing, iris localization, normalization, feature extraction, encoding, and iris matching. Lastly, an LCD screen will display the iris authentication results.

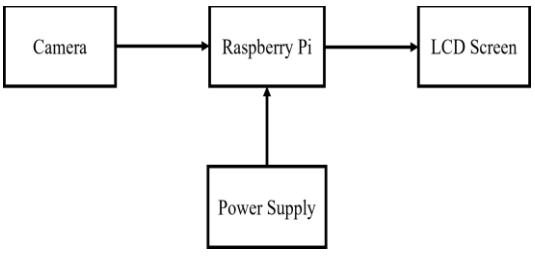

Fig. 1 System Block Diagram

System Flowchart

Figure 2 illustrates the sequential flow of the software for the study. The source of iris images may come from one of the two sources; it can be from a camera module or from the IIT Delhi database. The images captured from the camera will be saved first as BMP.

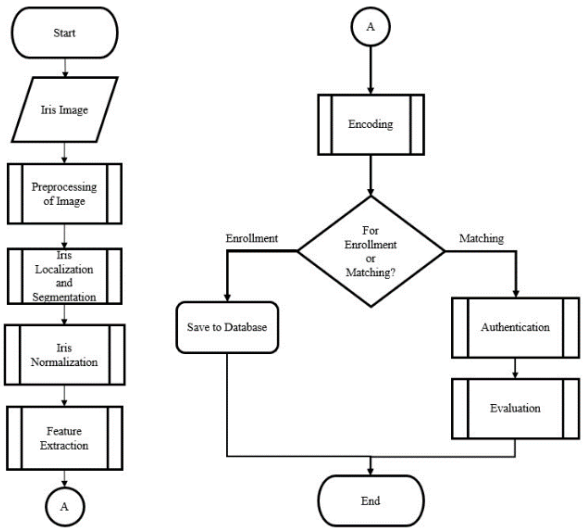

Fig. 2 Main System Flowchart

Figure 3 shows the main system's predefined process modules: Preprocessing, Iris Localization and Segmentation, Iris Normalization, Feature Extraction, Encoding, and Authentication.

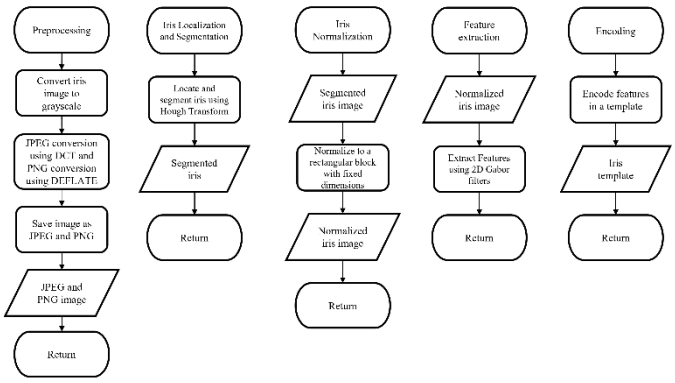

Fig. 3 Predefined Process Modules

The Preprocessing module converts the input iris image to its corresponding grayscale version for both file formats. The compression algorithm applied to the image will depend on the file type trial. On JPEG trials, the image is converted to JPEG, which uses DCT; On PNG trials, the image is converted to PNG, which uses DEFLATE.

In the Iris Localization and Segmentation module, only the regions of interest, the outer and inner iris, remain. This is done so by applying Hough Transform to find circular shapes and drawing them in the image so that the boundary between the region of interest and the rest of the image is drawn; in this context, two circular regions are expected to be detected, the iris and the pupil. After which, the iris region would be further segmented so that only the iris remains. Figure 4 shows what a localized and segmented iris image input looks like.

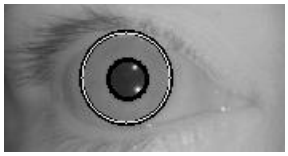

Fig. 4 Localization of Eye Input

The Iris Normalization module receives the segmented iris image as input, and it would be transformed into a rectangular strip that has fixed dimensions to remove inconsistencies that can be found in the image.

The normalized iris image is now fit for feature extraction. Gabor Filters are used to extract essential features from the iris. A bank of Gabor filters with varying parameters of the given equation would be convolved with the current normalized iris image. The extracted features will be collected and stored in a feature vector converted to binary form. Furthermore, if the current image in the system is set for enrollment, it would be saved onto the database after encoding.

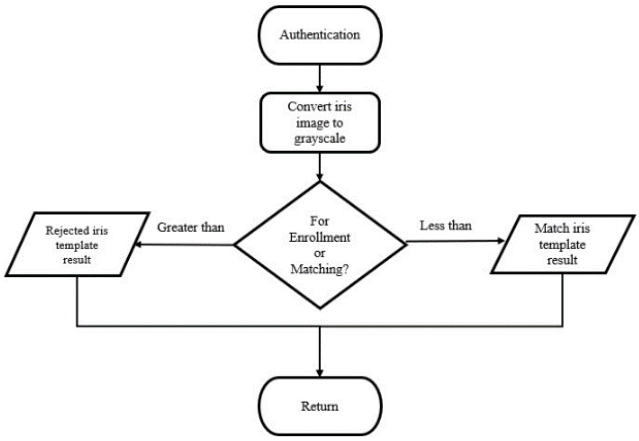

Fig. 5 Authentication Module

However, if the image is set for authentication, then the system proceeds to the authentication module, as shown in the figure above. Binary patterns of the enrolled iris image versus binary patterns of the iris image cued for authentication would be compared using Hamming Distance. The equation defines HD:

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j \oplus Y_j \qquad (1)$$

where,

- N is the total number of bits in the bit pattern
- X and Y are the two-bit patterns to be compared
- $\oplus$ is an exclusive OR operator

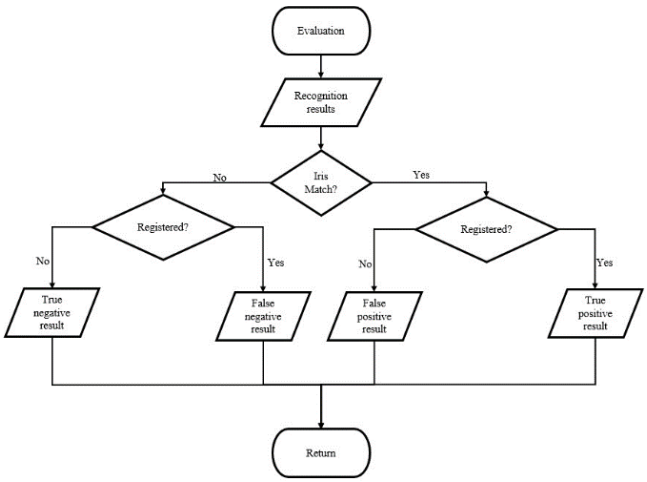

Fig. 6 Evaluation Module

Lastly, figure 6 shows the evaluation module. In this process, the iris recognition results will be evaluated to find out its result type. Recognition results from registered inputs or "authentic inputs" may be classified either into true positive (TP) or false positive (FP), while recognition outputs from unregistered inputs or "impostor inputs" may be classified either as false positive (FP) or true negative (TN).

Experimental Setup

The main components of the prototype include a Raspberry Pi NoIR Camera V2 to take the iris images, a Raspberry Pi 4 Model B+ to handle all processes, and a 7-inch Raspberry Pi LCD to display the output, all of which will be powered by a power supply. The camera is mounted on a fixed and stable stand. To help illuminate the subject's eye, an infrared LED is placed below the camera. A chin rest is also available for stable and consistent image acquisition. This overall setup can be seen in Figure 7.

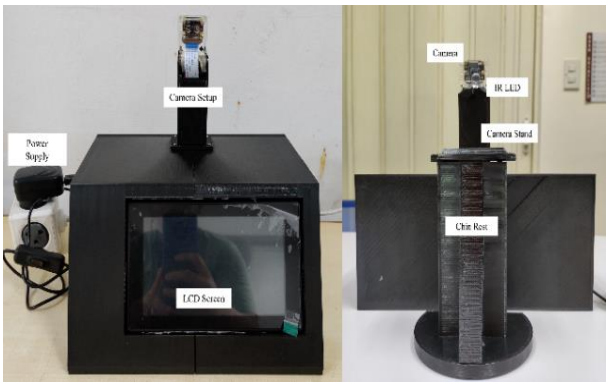

Fig. 7 Experimental Setup

Data Gathering Procedures

There would be 40 trials for each image file format, namely JPEG and PNG; overall, 80 trials would be conducted for this study. For each trial, an eye image will be inputted into the system to test whether or not the system will recognize the input.

Additionally, 20 eye images out of the 40 trials will already be enrolled in the system's database for each file format; These will be called authentic inputs. The remaining 20 trials are not enrolled in the system and will be called impostor inputs. Furthermore, the processing time for each trial will be recorded.

The scores obtained from the trials would then be tallied using confusion matrices. By obtaining the false positive (FP) and true negative (TN) values, the False Accept Rate (FAR) will be calculated. By obtaining the false negative (FN) and true positive (TP) values, the False Reject Rate (FRR) will be calculated.

There are two groups to group the data produced in the study, JPEG inputs and PNG inputs. The two groups were both subdivided into authentic and impostor inputs. Trial numbers 1-10 and 21-30 for every trial group are images from the IIT Delhi database. Trial numbers 11-20 and 31-40 are images acquired from the camera module of the system. Each table shows the trial number, the input type (authentic or impostor), the trial's recognition result, and the result type.

RESULTS AND DISCUSSION

Data Gathered

TABLE I. JPEG INPUTS RECOGNITION RESULTS

Trial No.	JPEG Inputs Recognition Results			
	Input Type (Actual)	*Recognition Result*	*Result Type*	*Processing time (s)*
1	Authentic	Match	TP	2.934
2	Authentic	Match	TP	2.9133
3	Authentic	Match	TP	2.7654
4	Authentic	Match	TP	2.6804
5	Authentic	Match	TP	3.0432
6	Authentic	Match	TP	2.7632
7	Authentic	Match	TP	2.6417
8	Authentic	Match	TP	3.1802
9	Authentic	Match	TP	2.9701
10	Authentic	Match	TP	2.8707
11	Authentic	Match	TP	0.8202
12	Authentic	Match	TP	0.8347
13	Authentic	Match	TP	0.8072
14	Authentic	Match	TP	0.833
15	Authentic	Match	TP	0.8531
16	Authentic	Reject	FN	0.8322
17	Authentic	Match	TP	0.7403
18	Authentic	Match	TP	0.8199
19	Authentic	Match	TP	0.8862
20	Authentic	Match	TP	0.8834
21	Impostor	Reject	TN	2.6632
22	Impostor	Reject	TN	2.7931
23	Impostor	Reject	TN	2.7867
24	Impostor	Reject	TN	3.1037
25	Impostor	Reject	TN	2.6891
26	Impostor	Reject	TN	2.646
27	Impostor	Reject	TN	2.8557
28	Impostor	Match	FP	0.8691
29	Impostor	Reject	TN	2.6112
30	Impostor	Reject	TN	2.6273
31	Impostor	Reject	TN	0.8196
32	Impostor	Reject	TN	0.7655
33	Impostor	Reject	TN	0.8208
34	Impostor	Reject	TN	0.8183
35	Impostor	Reject	TN	0.8167
36	Impostor	Reject	TN	0.9194
37	Impostor	Reject	TN	0.7744
38	Impostor	Reject	TN	0.7899
39	Impostor	Reject	TN	0.8421
40	Impostor	Reject	TN	0.7459

Table I shows the recognition results for JPEG inputs. Trials 1-20 show authentic inputs. Among these, 19 resulted in a "match" (TP), and one (1) resulted in a "reject" (FN). Meanwhile, trials 21-40 show impostor inputs where 19 resulted in a "reject" (TN) and 1 resulted in a "match" (FP).

TABLE II. PNG INPUTS RECOGNITION RESULTS

Trial No.	PNG Inputs Recognition Results			
	Input Type (Actual)	*Recognition Result*	*Result Type*	*Processing time (s)*
1	Authentic	Match	TP	2.9547
2	Authentic	Match	TP	2.9403
3	Authentic	Reject	FN	3.2105
4	Authentic	Match	TP	2.7893
5	Authentic	Match	TP	3.1667
6	Authentic	Match	TP	2.7994
7	Authentic	Match	TP	2.7283
8	Authentic	Match	TP	3.2355
9	Authentic	Match	TP	2.9646
10	Authentic	Match	TP	2.9063
11	Authentic	Match	TP	0.9014
12	Authentic	Match	TP	0.8566
13	Authentic	Match	TP	0.7835
14	Authentic	Match	TP	0.8064
15	Authentic	Match	TP	0.8059
16	Authentic	Reject	FN	0.8638
17	Authentic	Match	TP	0.7708
18	Authentic	Match	TP	0.8308
19	Authentic	Match	TP	0.9189
20	Authentic	Match	TP	0.9038
21	Impostor	Reject	TN	2.7258
22	Impostor	Reject	TN	2.8009
23	Impostor	Reject	TN	2.7821
24	Impostor	Reject	TN	3.1234
25	Impostor	Reject	TN	2.7179
26	Impostor	Reject	TN	2.6545
27	Impostor	Reject	TN	2.8705
28	Impostor	Match	FP	0.8832
29	Impostor	Reject	TN	2.6435
30	Impostor	Reject	TN	2.675
31	Impostor	Reject	TN	0.6771
32	Impostor	Reject	TN	0.7953
33	Impostor	Reject	TN	0.8429
34	Impostor	Reject	TN	0.8428
35	Impostor	Reject	TN	0.84
36	Impostor	Reject	TN	0.9617
37	Impostor	Reject	TN	0.8038
38	Impostor	Reject	TN	0.8048
39	Impostor	Reject	TN	0.857
40	Impostor	Reject	TN	0.8179

Table II shows the recognition results for PNG inputs. Trials 1-20 show authentic inputs. Among these, 18 resulted in a "match" (TP), and two (2) resulted in a "reject" (FN). Meanwhile, trials 21-40 show impostor inputs where 19 resulted in a "reject" (TN) and 1 resulted in a "match" (FP).

Statistical Treatment

Evaluation of the performance of the JPEG and PNG involves the computation of FAR and FRR of the recognition results. Two 2x2 confusion matrices show the tally of the result types.

TABLE III. 2x2 CONFUSION MATRIX FOR JPEG TRIAL RESULTS

		Predicted	
		Authentic	Impostor
Actual	Authentic	19	1
	Impostor	1	19

Table III shows the confusion matrix for the JPEG trial results. It shows that trials that used the JPEG image file format had one (1) falsely rejected authentic and one (1) falsely accepted impostor.

TABLE IV. 2x2 CONFUSION MATRIX FOR PNG TRIAL RESULTS

		Predicted	
		Authentic	Impostor
Actual	Authentic	18	2
	Impostor	1	19

Table IV shows the confusion matrices for the PNG trial results. It shows that trials that used the PNG image file format had two (2) falsely rejected authentic and one (1) falsely accepted impostor.

The accumulative FP and TN for all "imposter" trials are summed together and would serve as the denominator for the same tallied FP score of all "imposter" trials, which finally yields the FAR.

$$FAR = \frac{FP}{FP + TN} \quad (2)$$

On the other hand, the accumulative TP and FN for all “authentic” trials are summed together. They would be the denominator for the same tallied FN score of all "authentic" trials, yielding the FRR.

$$FRR = \frac{FN}{FN + TP} \quad (3)$$

TABLE V. PROCESSING TIME SUMMARY AND FRR & FAR RESULTS

Image File Type	Processing Time Summary and FRR & FAR Results			
	Total Processing Time (s)	*Average Time (s)*	FRR	FAR
JPEG	70.8301	1.7708	0.05	0.05
PNG	72.2576	1.8064	0.10	0.05

With the processing time recorded for each trial and the tallied results from the confusion matrices, the total and average processing time, and the FRR and FAR results on the used iris recognition system for each file format were obtained. Table V shows that JPEG had a total and average processing time of 70.8301s and 1.7708s, respectively. Meanwhile, PNG had a total and average processing time of 72.2576s and 1.8064s, respectively. For the FRR and FAR results, JPEG resulted in an FRR of 0.05 and a FAR of 0.05. Meanwhile, PNG resulted in an FRR of 0.10 and a FAR of 0.05.

CONCLUSION

Implementation of the iris recognition system in Raspberry Pi computer version 4B is a success by the researchers. Using the system’s camera and the IIT Delhi Database, the researchers were able to create two eye image databases with different file formats, namely JPEG and PNG. With the iris recognition system, the researchers were able to show the difference in the processing time and the FAR and FRR results between JPEG and PNG. This comparative study identified that the extracted iris features from JPEG produced better recognition performance than PNG using total and average processing time, and using FRR and FRR in an IR system that uses Daugman’s Algorithm.

RECOMMENDATIONS

The researchers recommend using a better camera and lighting equipment to optimize image acquisition which is a crucial factor for the success of iris recognition. To improve the study, the researchers recommend the inclusion of other image file formats to determine which performs best in an iris recognition system implemented on Raspberry Pi that uses the Daugman algorithm. As this study is limited to the usage of a sole algorithm — the Daugman Algorithm, using a different algorithm is endorsed by the researchers to determine which of the extracted iris features between JPEG and PNG would perform better with that specific iris recognition algorithm.

REFERENCES

[1] S. S. Jauro and R. Yadav, “Review on Iris Recognition Research Directions- A Brief Study,” *Internatonal J. Appl. Eng. Res. ISSN 0973-4562*, vol. 13, no. 10, pp. 8728–8735, 2018.

[2] M. V. C. Caya, J. P. H. Durias, N. B. Linsangan, and W. Y. Chung, “Recognition of tongue print biometrie using binary robust independent elementary features,” *HNICEM 2017 - 9th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag.*, vol. 2018-Janua, pp. 1–4, 2017.

[3] A. P. I. D. Magadia, R. F. G. L. Zamora, N. B. Linsangan, and H. L. P. Angelia, “Bimodal Hand Vein Recognition System using Support Vector Machine,” *2020 IEEE 12th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2020*, pp. 2–6, 2020.

[4] J. D. A. Villarama, J. Paul, R. O. Gernale, D. A. N. Ocampo, and J. F. Villaverde, “Wireless Biometric Attendance Management and Payroll System,” no. 1, pp. 3–7, 2017.

[5] S. I. Manzoor and A. Selwal, “An analysis of biometric based security systems,” *PDGC 2018 -*

2018 5th Int. Conf. Parallel, Distrib. Grid Comput., pp. 306–311, 2018.

[6] S. H. and S. Malisuwan, "A Study of Image Enhancement for Iris Recognition," *J. Ind. Intell. Inf.*, vol. 3, no. 1, pp. 61–64, 2014.

[7] N. B. Linsangan *et al.*, "Iris Recognition using Daugman algorithm on Raspberry Pi," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 2126–2129, 2017.

[8] R. Gagan and S. Lalitha, "Elliptical Sector Based DCT Feature Extraction for Iris Recognition," *Proc. 2015 IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2015*, pp. 1–5, 2015.

[9] J. Hu, S. Song, and Y. Gong, "Comparative performance analysis of web image compression," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-Janua, pp. 1–5, 2018.

[10] N. Saraswat and H. Ghosh, "A Study on Size Optimization of Scanned Textual Documents BT - Image and Video Technology," 2016, pp. 75–86.

[11] S. Oswal, A. Singh, and K. Kumari, "Deflate compression algorithm," *Int. J. Eng. Res. Gen. Sci. 2016*, vol. 4, no. 1, pp. 430–436, 2016.

[12] M. Singh, S. Kumar, S. Singh, and M. Shrivastava, "Various Image Compression Techniques: Lossy and Lossless," *Int. J. Comput. Appl.*, vol. 142, no. 6, pp. 23–26, 2016.

[13] M. Abuzaher and A.-A. Jamil, "JPEG Based Compression Algorithm," *Int. J. Eng. Appl. Sci.*, vol. 4, no. 4, p. 257481, 2017.

[14] B. J. Andujar, N. J. Ferranco, and J. F. Villaverde, "Recognition of Feline Epidermal Disease using Raspberry-Pi based Gray Level Co-occurrence Matrix and Support Vector Machine," *2021 IEEE 13th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2021*, 2021.

[15] A. N. Yumang, J. F. Villaverde, M. H. C. Tan, and J. K. D. Tulfo, "Bacterial Leaf Blight Identification of Rice Fields Using Tiny YOLOv3," *4th IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2022*, pp. 1–5, 2022.

[16] V. A. M. Luis, M. V. T. Quinones, and A. N. Yumang, "Classification of Defects in Robusta Green Coffee Beans Using YOLO," *4th IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2022*, pp. 2–7, 2022.

[17] J. R. I. Borreta, J. A. Bautista, and A. N. Yumang, "Snail Recognition Using YOLO," *4th IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2022*, pp. 1–6, 2022.

[18] M. C. Ang, K. R. C. Taguibao, and C. O. Manlises, "Hand Gesture Recognition for Filipino Sign Language Under Different Backgrounds," 2022.

[19] A. N. Yumang, C. C. Paglinawan, L. C. M. Andrada, E. C. Garcia, and J. M. F. Hernandez, "Optimization of vehicle speed calculation on raspberry pi using sparse random projection," *2018 IEEE 10th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2018*, 2019.

[20] M. V. C. Caya, C. J. P. Rapisura, and R. R. B. Despabiladeras, "Development of Pupil Diameter Determination using Tiny-YOLO Algorithm," *2022 IEEE 14th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2022*, pp. 1–6, 2022.

[21] D. E. S. Banguilan, C. K. S. Veneracion, and A. N. Yumang, "Raspberry PI based Food Recognition for Visually Impaired using YOLO Algorithm," pp. 165–169, 2021.

[22] A. S. Borras, R. A. B. Ganotisi, N. B. Linsangan, and R. A. Juanatas, "Non-destructive Determination of Sweetness of Philippine Fruits using NIR Technology," no. 1, 2022.

[23] C. O. Manlises, J. M. Martinez, J. L. Belenzo, C. K. Perez, and M. K. T. A. Postrero, "Real-time integrated CCTV using face and pedestrian detection image processing algorithm for automatic traffic light transitions," in *8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2015*, 2016.

[24] N. B. Linsangan *et al.*, "Real-time iris recognition system for non-ideal iris images," *ACM Int. Conf. Proceeding Ser.*, pp. 32–36, 2019.

[25] S. Minaee and A. Abdolrashidi, "DeepIris: Iris Recognition Using A Deep Learning Approach," *arXiv Prepr. arXiv1907.09380*, 2019.

[26] A. Paul, T. Z. Khan, P. Podder, R. Ahmed, M. M. Rahman, and M. H. Khan, "Iris image compression using wavelets transform coding," *2nd Int. Conf. Signal Process. Integr. Networks, SPIN 2015*, pp. 544–548, 2015.

[27] "IIT Delhi Iris Database (Version 1.0)," 2007. [Online]. Available: http://web.iitd.ac.in/~biometrics/Database_Iris.html.