

Robust embedded access control system based on face and encrypted QR with RPi4

Samir Marwan Hammami¹, Muhammad Alhammami²

¹Department of Management Information Systems, Dhofar University, Salalah, Oman

²Department of Communication, Higher Institute for Applied Sciences and Technology, Damascus, Syria

Article Info

Article history:

Received Jan 26, 2024

Revised May 3, 2024

Accepted May 17, 2024

Keywords:

Access control

Deep metric learning

Face recognition

QR codes

Raspberry Pi 4

ABSTRACT

Facial-based recognition systems are commonly used for building access control, with the accuracy and computing requirements still being improved. On the other hand, QR codes are gaining rising attention as an input interface to many embedded applications. This paper proposes an embedded access control system that customises both previous techniques to be implemented on the CPU of a low-cost Raspberry Pi 4 computer. The achieved system works smoothly with a frame rate of 8.27 FPS, increasing the accessing control's robustness compared to a system based on face recognition only. It also offers the ability to control the access of unknown faces. In tandem with integration, this strengthens security measures, improves user experience, and outperforms conventional access control approaches, creating an attractive offer for many businesses.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muhammad Alhammami

Department of Communication, Higher Institute for Applied Sciences and Technology

P.O. Box 31983, Damascus, Syria

Email: Muhammad.Alhammami@hiast.edu.sy

1. INTRODUCTION

One of the most important aspects of building security is access control. Conventional systems generally depend on passwords, keys, or cards that can be misplaced, stolen, or forgotten. Facial recognition technologies have been employed more lately and provide a more user-friendly and secure option. Nonetheless, there are still issues with these systems' accuracy and processing demands. Moreover, they are unable to restrict access for unidentified faces.

Numerous researches have looked on the application of face recognition to access control. These efforts have helped to lower the processing requirements and increase the accuracy of facial recognition systems. They do not, however, address the problem of limiting access for unknown persons and instead concentrate mostly on well-known faces. However, there may be a workaround for this issue as QR codes have been employed as an input interface for a number of embedded programs. A detachable color two-dimensional code design is presented [1] in which the authors presented a state-of-the-art technique for producing two-dimensional codes with distinct color components. Information on the potential benefits and applications of this design is provided by the research.

Research by Yang *et al.* [2] introduces a two-dimensional code hierarchical encryption algorithm based on attribute encryption. The authors propose attribute-based encryption as an encryption method to increase the security of two-dimensional codes. The study emphasizes the advantages of using this hierarchical encryption method to protect sensitive information. A logistic information privacy protection system based on encrypted QR code is described [3]. They provide a method that protects the confidentiality

of logistical data by using encrypted QR codes. The system design and its possible uses in protecting sensitive data are described by the authors.

Research by Liang *et al.* [4] offer a design of system and key algorithms for 2D code recognition with various security levels. The authors provide a method that combines two-dimensional code recognition with varying security levels. They highlight the significance of customized security levels while going over the system's architectural concerns and important algorithms. Research by Xu *et al.* [5] detail the creation and use of a system for identifying two-dimensional codes that runs on a mobile phone. The system architecture and its possible uses in a number of domains are covered in the paper.

Research by Zhu *et al.* [6] talk about the application of QR code recognition technology in express sign. The use of QR code recognition technology in express sign systems is examined by the writers. They draw attention to the benefits and real-world applications of utilizing QR codes for effective express sign procedures.

Research by Hong [1] presents the paper "design of an intelligent access control system based on DES encrypted QR code" at the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications. The author suggests using DES-encrypted QR codes in an intelligent access control system. The system architecture, implementation specifics, and possible uses in access control scenarios are presented in this study.

Even with these developments, access control technology may still be improved. It is possible to improve facial recognition systems' accuracy and computing efficiency. Furthermore, limiting access for unfamiliar faces is a serious problem that has not received enough attention in the literature.

This paper suggests a novel fix for these issues. It describes an embedded access control system that runs on the CPU of an inexpensive Raspberry Pi 4 computer and integrates face recognition and QR codes. This method provides access control for unknown faces in addition to improving the robustness of access control over a facial recognition-only system. The system operates according to the following procedures and can be reconfigured to adapt to different access control requirements or scenarios: i) the faces of all employees are recorded; ii) each employee is provided with a secure QR code; iii) features are extracted offline for each employee and stored in the system. These features will be used for recognition in real time; iv) a database defining all employee permissions is built offline using structured query language (SQLite); v) Any employee entering or exiting the control system must show his/her own secure QR code besides being recognised by his/her face. The employee's QR code must match his face, plus he has permission to pass the control system; vi) guests are provided with a direct one time password (OTP) based QR code, which allows them to access once during a specific time; and vii) the system records all the logs of all functions.

The following sections of this manuscript will detail the design and implementation of our proposed system. We will present experimental results demonstrating the system's performance, including its smooth operation at a frame rate of 8.27 FPS. We will also discuss how our system strengthens security measures, improves user experience, and outperforms conventional access control approaches, making it an attractive option for many businesses.

2. RESEARCH METHOD

2.1. Functionalities of face recognition

The goal is to construct an edge-operating facial recognition system using low-cost hardware like the Raspberry Pi4. This means that it must be approached face recognition as a one-time learning job, which is ideal for deep metric learning (DML). DML trains a neural network to learn an embedding of a data input, allowing distance or similarity between data points to be measured. This means that, in the context of face recognition, one must learn a function that can convert an input image of a face into a point in a high-dimensional space where images of the same face are widely separated from one another.

2.1.1. Creating a dataset

There are a few important procedures and things to keep in mind while compiling a dataset of the authorized individuals for this application. Usually, the procedure begins with data gathering, in which webcam photographs of faces are taken. A broad range of changes in stance, face expression, lighting, occlusion, and resolution should ideally be covered by these photographs. The pertinent details about each face are then tagged into the gathered photos. Because it supplies the ground truth labels required for machine learning model training, this annotation procedure is essential.

2.1.2. Face detection

The computer vision problem of face detection is detecting and recognizing human faces in digital photos or movies. All facial analysis algorithms, such as those for face alignment, recognition, verification,

and parsing, start with this phase. The main objective of face detection given a digital image is to find out if the image contains any faces.

There are several methods for face detection [7]–[9], each with its strengths and weaknesses. Some of the most common techniques include:

- Haar-based technique: this method is speedy and requires low computational resources, making it suitable for resource-constrained devices. However, it is highly prone to false-positive detections and requires manual tuning.
- Multi-task cascaded convolutional neural networks (MTCNN): is a deep learning-based method that has achieved state-of-the-art results on standard benchmark face detection datasets. It is more complex and computationally intensive than the haar-based technique but offers higher accuracy.
- Histogram of oriented gradients (HoG)+support vector machine (SVM): This method is also widely used in face detection. It involves extracting HoG features from the image and using an SVM for classification.

Standards for face detection often involve ensuring the quality and diversity of the images. The National Institute of Standards and Technology (NIST) provides facial recognition technology guidelines [10], which include face detection. These standards ensure the technology is accurate, reliable, and can be used seamlessly across devices.

The benefits of face detection are numerous. They enhance the accuracy of facial recognition systems by allowing the model to learn diverse features of human faces. This improves performance in recognising different individuals [11]–[13]. Furthermore, face detection is an essential tool for facial recognition technology, providing the requisite training materials for accurately identifying individuals' faces in real-world scenarios.

In terms of computational needs, different face detection methods have different requirements. For instance, Haar-based techniques are less computationally intensive and can be run on embedded devices. In contrast, deep learning-based methods like MTCNN require more computational resources but offer higher accuracy. To use a face detection method, it is typically necessary to load the face detection model into the environment, preprocess the images (such as resizing and normalisation), and then feed them into the model for face detection. Once the faces are detected, they can be used for further processing or analysis.

Histogram of oriented gradients method "HoG", published in 2005 [14], is used because of its speed, high detection accuracy and low false positive rate. The input of this stage is a grey image, and the outputs are areas where faces are located, which will be the inputs of the next stage. For each pixel in the grey image, HoG first looks at the surrounding pixels, compares the darkness of the pixel with its surroundings and draws an arrow towards the darkest area. Repeating this process on all the image pixels gives a set of arrows called a gradient that indicates the flow from the brightest region to the darkest region. Next, HoG divides the image into 16×16 pixel squares and places only one arrow in the direction of the dominant arrows in that square, as shown in Figure. 1.

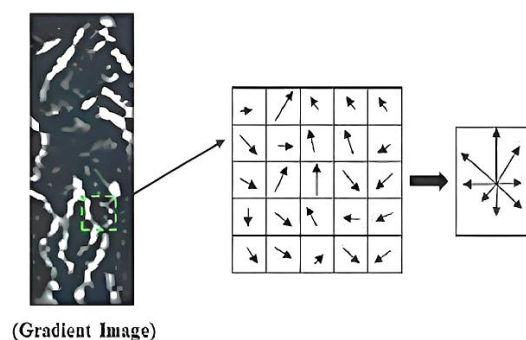


Figure 1. Gradient map

2.1.3. Face detection

Face repositioning is a technology used in computer vision, facial identification, and medical operations. It is often referred to as face reshaping or face transformation. It entails physically modifying a face's structure or rearranging the orientation or placement of facial characteristics in a picture.

Face repositioning has several advantages. By guaranteeing that faces are positioned consistently, it enhances the efficiency of facial recognition algorithms in computer vision. This facilitates the algorithm's ability to match a face in a database with an input face. In the medical industry, a patient's look and self-esteem can be greatly enhanced by face realignment. Additionally, it can address functional problems including trouble breathing or eating. There are several uses for face repositioning. It is used to increase the precision of face detection and recognition algorithms in computer vision and facial recognition. Before putting faces into a recognition system, face repositioning, for example, might normalize them to a standard posture. Face repositioning is used in medicine to improve facial cosmetics or address functional problems during treatments like facelifts and orthognathic surgery.

For face repositioning, many methods are employed [15]. Faces in pictures can be repositioned in computer vision using methods including thin-plate spline transformations, projective transformations, and affine transformations. These methods convert an input image's face characteristics to a reference image's standard set of facial features. In the medical industry, algorithms are used to organize surgical treatments based on the specific facial anatomy of the patient and the intended result. Repositioning facial features refers to shifting their direction or location. This can involve scaling features to make them bigger or smaller, rotating features, and moving elements in both horizontal and vertical directions. The particular algorithm or process that is employed determines the precise characteristics of the relocated face.

Face repositioning is typically performed using a software library that implements the necessary algorithms. The user provides an input image and a reference image, and the software repositions the face in the input image to match the face in the reference image. In the medical field, face repositioning is performed by a trained surgeon. The surgeon uses medical imaging technology to plan and perform the necessary surgical steps.

In this project, post to isolating the face region in the previous stage and in order to deal with the rotation, centring and scaling of the faces, the "face landmark estimation" algorithm by Kazemi and Sullivan [16] is used to find 68 face landmarks shown in Figure 2 where the landmarks are shown on the face in Figure 2(a) and the names of each component and its point region in Figure 2(b). Examples of these landmarks are the top of the chin, the edges of the eye, the edges of the eyebrow, and others.

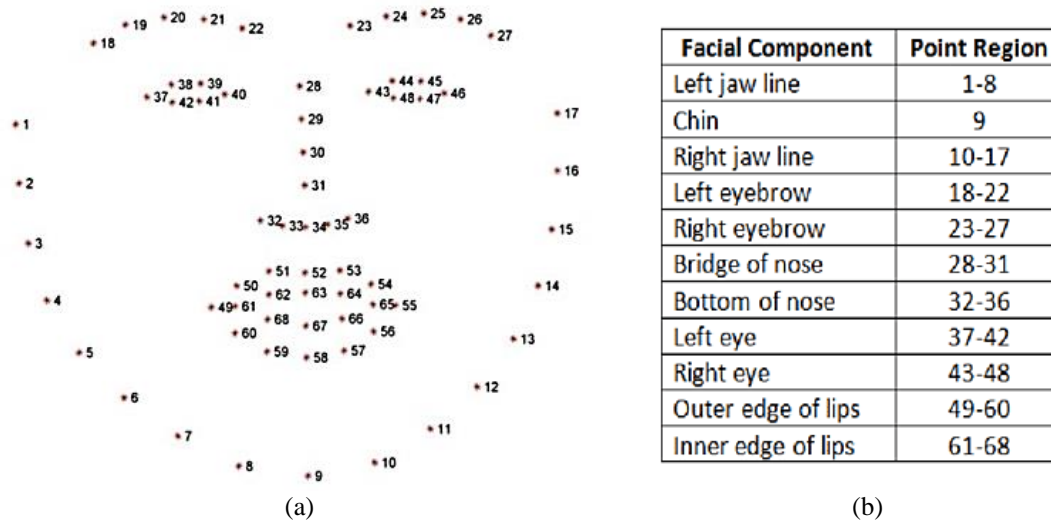


Figure 2. Face landmark estimation (a) algorithm by Kazemi and Sullivan [16] on the face and (b) name of each component and its point region

2.1.4. Extraction of facial features vector

A branch of machine learning called DML is concerned with acquiring distance measurements for the purpose of comparing and analyzing data points in high-dimensional spaces [17]. It seeks to comprehend a similarity measure that uses samples to calculate how similar or different two or more items are. Semantic textual similarity, 3D form retrieval, speaker verification, face recognition, face verification, and person re-identification are just a few of the fields in which DML has been used. It is a fundamental component of many computer vision applications and is applied to a variety of tasks, including information retrieval, clustering, and k-Nearest Neighbor (k-NN) classification.

The characteristics of a distance measure serve as the foundation for DML standards. It must possess triangle inequality, symmetry, the identity of indiscernibility, and non-negativity. Additionally, network architecture, sampling policy, and suitable distance measure provide difficulties for researchers looking to enhance network model performance.

DML has several advantages. It uses activation functions to give a better solution for nonlinear data. It enables computers to learn well without needing to be explicitly taught, providing effective answers to challenging issues and massive volumes of data. Additionally, it enhances the learnt metric's discriminating capacity by putting forth creative sampling plans or loss functions.

DML uses machine learning to automatically create task-specific distance measures from (weakly) supervised data. Then, a variety of activities may be carried out using the acquired distance metric. From the input domain, it learns a mapping to an embedding space where different things are situated far away from one another and semantically related ones are located adjacent.

It is required to extract features from each picture and produce a modified embedding in order to employ DML algorithms. A variety of learning algorithms are available, including triplet mining, triplet sampling, and Siamese networks and related loss functions. Deep discriminant analysis techniques based on Fisher's discriminant analysis are also an option. Lastly, multi-modal DML, few-shot metric learning, and geometric metric learning using neural networks may all be investigated.

Deep neural network-based models achieve state-of-the-art performance in computer vision tasks, including face recognition. However, there are two obstacles in this application. The first one is that Raspberry Pi has limited computational resources, and the second obstacle is that the number of training image examples per face is relatively small to learn a neural network. Such a problem is called a one-shot learning problem. DML generates close feature vectors corresponding to faces belonging to the same person while increasing the distance between those belonging to different faces. In this work, the DML approach is based on a pre-trained ResNet-34 neural network [18] to generate feature vectors of 128 integer lengths for all employees' faces in the dataset.

2.1.5. Recognizing the face

As this application uses the DML technique; so the last stage is identifying the person with an image which is not in the original dataset by finding the nearest features vector from the former stage. K-mean function is used to measure the similarity or dissimilarity between pairs of vectors. Hence, to effectively measure the similarity between different faces. The recognition result will give the employee's name and this takes only milliseconds to be executed.

2.2. Using QR codes

There are two main objectives of using QR codes in this system. The first one is to increase the reliability of identifying a person by giving him his own dynamic QR code generator as a client app on his smartphone based on its international mobile equipment identity (IMEI) identifier based on the advanced encryption standard (AES) encryption algorithm, besides recognising his face when entering and exiting, doing as a second authentication factor will prevent the intrusion of the access control system by using a fake picture of an employee. The dynamic QR code generator app is built using a Flutter environment but is out of the scope of this paper. The second objective is to deal with people not in the database (such as visitors) or who are not usually allowed to access certain places. In this case, they are given a unique code used only once and during a specific time.

2.2.1. QR structure

The QR code's structure is made up of black and white checkered pixel patterns that are separated into several particular regions 1, 9, 8. Each area serves a distinct role. Figure 3 is an example of a version 3 QR code construction.

- Firstly, "finder pattern/positioning marker (black areas, 1)": three identical areas at the QR code's corners. Each pattern is represented as a NxN matrix. "finder patterns" aid the decoder in detecting the QR code and determining the right orientation.
- Secondly, "data (pink regions, 2)": blocks of bits containing real data.
- Thirdly, format information (green regions, 3): these bits hold information about the error correction level and masking pattern utilized.
- Fourthly, "separators (yellow areas, 4)": these areas are white, which implies they contain exactly zero bits. The width of each white separator is one pixel. "Separator areas" enhance the recognizability of "finder patterns" by separating them from the real data.

- Fifthly, "timing pattern (orange regions, 5)": the width of a single square is determined by alternating between black and white bits.
- "Error correction (purple regions, 6)": the amount of error correction words depends on the code version and error severity.
- "Remainder bits (dark blue, 7)": this area follows the error correcting zone and is made up of empty bits that are not always zero.
- "Sixthly, alignment patterns (bright blue, 8)": the decoder uses them to correct for mild visual distortions. Depending on the edition, these patterns might be more than one or zero.
- Finally, to identify it from its surroundings, the QR code is surrounded by a "Silent zone" (a region with just zero bits).

The decoding mechanism is carried out according to the flowchart in Figure 4. However, for detailed information about the structure of the QR code and the process of coding and decoding it, one can get back to [19], [20]. These sources provide comprehensive information on the QR code's format information, data masking patterns, and the mathematical principles that govern the error correction capabilities. They also deeply elaborate on the decoding process.



Figure 3. Structure of a QR code

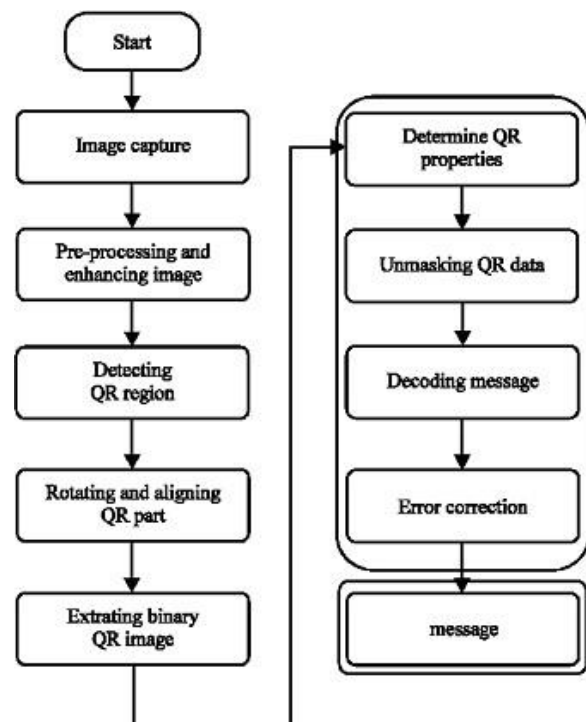


Figure 4. Flowchart of decoding QR codes

2.2.2. Reading QR codes

QR codes have a complex reading process, requiring a long computation time. However, this processing time is still acceptable for many applications. All existing QR decoders implemented are software-based, either in mobile phone apps or other microprocessor-based embedded systems such as Raspberry Pi; nevertheless, for more advanced applications, such as data decryption for data security, where extra real-time and complex functions will occupy most of the computational resources of the processor. Figure 4 clearly illustrates the calculation work for each of the three primary steps of the QR decoding process: picture capture, image preprocessing, and QR decoding.

2.3. Complementary functions

2.3.1. Generating a one-time-password and encoding it with QR

One-time-password based on the AES encryption algorithm gives direct permissions regardless of facial recognition but only for a specified period. This allows the system to deal with visitors whose photos are not in the database. This password is converted to QR encoding in line with the system's working mechanism. The generated code is printed and given to the visitor to pass on to the digital camera used in the

system. PyOTP is a Python package that we use to create and validate one-time passwords. It is applicable to online applications and other systems that need users to log in, where two-factor (2FA) or multi-factor (MFA) authentication mechanisms can be used.

RFC 4226 (HOTP: an HMAC-based one-time password algorithm) and RFC 6238 (TOTP: time-based one-time password algorithm) establish the open MFA standards. Both of these protocols have server-side functionality implemented by PyOTP. Client-side support can be enabled by telling users to use Google authenticator, authy, or another compatible software for TOTP, or by delivering authentication codes to users by email or SMS (HOTP). By utilizing the camera on their phone to scan the QR codes offered by PyOTP, users may quickly set up auth tokens in their apps. We adhered to the relevant RFCs' sections on TOTP security concerns and HOTP security criteria, which state, at the very least:

- Using HTTPS to ensure transport secrecy.
- Maintaining the secrecy of HOTP/TOTP secrets by keeping them in a database with restricted access.
- Preventing replay attacks by refusing client-supplied one-time passwords (this necessitates keeping the most recent authenticated timestamp, OTP, or OTP hash in the database and refusing the OTP upon finding a match).
- Reducing the frequency of brute-force attempts to access the application's login features

2.3.2. Defining the permissions database

One integrated SQL database engine that doesn't need a separate server process is called SQLite [21]. It directly writes and reads from standard disk files. An whole SQL database with several tables, indices, triggers, and views can be found on a single disk file. Since the Raspberry Pi 4 has limited resources, SQLite is a great option because it is lightweight and tiny [22].

SQLite is self-contained, meaning no external dependencies are required. It's serverless, so it will not consume the Raspberry Pi's random access memory (RAM) and central processing unit (CPU) when not being utilised. SQLite has zero configuration, making it easy to use right out of the box. It's also cross-platform, available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT). In this project, SQLite defines and manages each employee's permissions. The permissions table contains the following columns (unique identifier, employee's ID, place ID, value 0 or 1 to block or allow access to this place).

2.3.3. Keeping logs of actions

The following data is saved continuously: i) employee's ID, date, time, place and entry/exit of persons using (face recognition +QR code); ii) the used OTP, the date, time, place and entry/exit of people (direct OTP); iii) all single-use passwords with their number and their generating date/time; iv) all unauthorised entry attempts because they do not match the permission definition table, and v) all unauthorised entry attempts have expired passwords.

3. RESULTS AND DISCUSSION

All the previous functions are implemented using a Raspberry Pi 4 nano computer using Python environment 3.6. Many open-source libraries were used in the implementation. Multi-threading and other parallel techniques are used so the system can work as fast as possible. In the tests, the system achieved 8.27 FPS and complete robustness. During the tests, using cascade filters in face detection lets the system achieve 12.23 FPS but with a lower detection rate. Using convolutional neural network (CNN) makes the system run at only 1.7 FPS but with a higher detection rate. Hence, HoG was ultimately chosen as it is a good compromise between speed and detection rate. As shown in Figure 5, the final prototype is built as a standalone system in an aluminium case and can be used using either a webcam or a Raspi Cam. It also has a built-in screen but can be connected to an external screen. It also has four keys for controlling the flow of the functions.

Like encrypted QR codes and advanced face recognition algorithms, this powerful synergy, orchestrated by the versatile Raspberry Pi4 platform, presents a robust and convenient solution for organisations seeking to elevate their security posture and user experience. The landscape of access control systems is undergoing a transformative shift driven by the convergence of cutting-edge technologies. The convergence of cryptographic QR codes and advanced facial recognition technology on the Raspberry Pi4 platform is revolutionising access control systems; it presents a robust and convenient solution for organisations seeking to elevate their security posture and user experience. This fusion creates a powerful tool for organizations to outperform conventional access control techniques. These benefits can be embedded in other safety applications like [23]–[25]. Another impact is the that this is an economical solution

comparing to conventional hardware-intensive solutions. Using Raspberry Pi4 makes our solution more flexible and affordable option.

However; there are still potential considerations to be always addressed. Firstly, privacy concerns like how to allay worries about facial recognition technology, it is imperative to put in place clear data security procedures and abide with applicable privacy laws. Secondly, lighting and environmental circumstances where accurate face recognition performances under a range of lighting and environmental circumstances depend on proper camera positioning and setup modifications. Thirdly, technical dependencies by putting in place backup plans and redundant power supplies, operational continuity is ensured by reducing the risk of disruptions from technical problems or power outages.



Figure 5. The final prototype

4. CONCLUSION

This research described the integration of facial recognition with encrypted QR codes to create a strong access control system. Although the database is now kept on the same system, moving it to the cloud would be preferable as it would make database modifications more dependable across various instances of the system. Although the system is running smoothly on Raspberry Pi 4, it was not feasible to implement it using an earlier version like Raspberry Pi 3.

The synergistic fusion of encrypted QR codes and facial recognition technology, powered by the Raspberry Pi4, represents a paradigm shift in access control systems. By prioritising robust security, streamlined user experience, and cost-effectiveness, this innovative approach caters to the evolving needs of organisations seeking to secure their environments while enhancing user convenience. However, a proactive approach to address potential privacy concerns, environmental influences, and technical dependencies is crucial for successful implementation and maximising the system's benefits.





REFERENCES

- [1] Y. Hong, "Design of intelligent access control system based on des encrypted QR code," in *Proceedings of 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2020*, IEEE, Aug. 2020, pp. 1005–1008, doi: 10.1109/AEECA49918.2020.9213475.
- [2] K. Yang, H. Yuan, and Y. Guo, "Two-dimensional code hierarchical encryption algorithm based on attribute encryption," *Computer Engineering*, vol. 44, no. 06, pp. 136–140, 2018, doi: 10.19678/j.issn.1000-3428.0046392
- [3] X. Zhang, H. Li, Y. Yang, G. Sun, and G. Chen, "LIPPS: logistics information privacy protection system based on encrypted QR code," in *Proceedings - 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 10th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Symposium on Parallel and Distributed Processing with Applications, IEEE TrustCom/BigDataSE/ISPA 2016*, IEEE, Aug. 2016, pp. 996–1000, doi: 10.1109/TrustCom.2016.0167.
- [4] R. Liang, Y. Wang, and X. Li, "Design of system and key algorithms for 2D code recognition with different security levels," *Computer Engineering and Design*, vol. 38, no. 11, pp. 2903–2908, 2017.
- [5] L. Xu, X. Jiang, and J. Zhang, "Design and implementation of two-dimensional code recognition system in mobile phone," *Journal of Computer Applications*, vol. 32, no. 5, pp. 1474–1476, Apr. 2013, doi: 10.3724/sp.j.1087.2012.01474.
- [6] J. Zhu, Y. Chang, and C. Zhu, "Application of qr code recognition technology in modern power logistics management," in *2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City*, M. Atiquzzaman, N. Yen, and Z. Xu, Eds. Singapore: Springer Singapore, 2022, pp. 351–357, doi: 10.1007/978-981-16-7469-3_40.
- [7] G. Yang and T. S. Huang, "Human face detection in a complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, Jan. 1994, doi: 10.1016/0031-3203(94)90017-5.
- [8] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.





- [9] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993, doi: 10.1109/34.254061.
- [10] P. J. Grother, P. J. Grother, M. Ngan, and K. Hanaoka, "Face recognition vendor test (FRVT)," US Department of Commerce, National Institute of Standards and Technology, Tech. Rep., 2014, doi: 10.6028/NIST.IR.8009.
- [11] M. Mahjeed, G. Thamilarasu, N. Johnson, and C. Alfonso, "A deep learning approach for ECG authentication on implantable medical devices," in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, IEEE, Jul. 2023, pp. 1–7, doi: 10.1109/ICCCN58024.2023.10230198.
- [12] H. Wang *et al.*, "Joint Biological ID : a secure and efficient lightweight biometric authentication scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2578–2592, 2023, doi: 10.1109/TDSC.2022.3186999.
- [13] D. Banerjee and K. Yu, "3D face authentication software test automation," *IEEE Access*, vol. 8, pp. 46546–46558, 2020, doi: 10.1109/ACCESS.2020.2978899.
- [14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, IEEE, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.
- [15] S. Ghali, *Introduction to geometric computing*. London: Springer London, 2008, doi: 10.1007/978-1-84800-115-2.
- [16] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2014, pp. 1867–1874, doi: 10.1109/CVPR.2014.241.
- [17] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1–2, pp. 305–307, Jun. 2018, doi: 10.1007/s10710-017-9314-z.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [19] M. Alhammami, C. P. Ooi, W. H. Tan, and S. Nyeanchong, "FPGA hardware implementation for accelerating QR decoding," *Journal of Engineering and Applied Sciences*, vol. 11, no. 14, pp. 3273–3278, 2016, doi: jeasci.2016.3273.3278.
- [20] H. S. Al-Khalifa, "Utilizing QR code and mobile phones for blinds and visually impaired people," in *Computers Helping People with Special Needs: 11th International Conference, ICCHP 2008, Linz, Austria, July 9-11, 2008. Proceedings 11*, Springer Berlin Heidelberg, 2008, pp. 1065–1069.
- [21] K. P. Gaffney, D. R. Hipp, M. Prammer, D. Kennedy, L. Brasfield, and J. M. Patel, "SQLite: past, present, and future," *Proceedings of the VLDB Endowment*, vol. 15, no. 12, pp. 3535–3547, Aug. 2022, doi: 10.14778/3554821.3554842.
- [22] S. Balarezo, X. Arias, and K. Espin, "Remote manipulation of a robotic Arm with 6 DOF via IBSV using a Raspberry Pi and Machine Vision," *Lecture Notes in Networks and Systems*, pp. 845–854, 2022, doi: 10.1007/978-3-031-10464-0_58.
- [23] M. Alhammami and S. M. Hammami, "An FPGA-based IP for recognizing violence against children," *MethodsX*, vol. 8, p. 101378, 2021, doi: 10.1016/j.mex.2021.101378.
- [24] M. Alhammami, S. M. Hammami, C. P. Ooi, and W. H. Tan, "Optimised ML-based system model for adult-child actions recognition," *KSI Transactions on Internet and Information Systems*, vol. 13, no. 2, pp. 929–944, Feb. 2019, doi: 10.3837/tiis.2019.02.024.
- [25] S. M. Hammami and M. Alhammami, "Vision-based system model for detecting violence against children," *MethodsX*, vol. 7, pp. 104–108, 2020, doi: 10.1016/j.mex.2019.11.017.

BIOGRAPHIES OF AUTHORS



Samir Marwan Hammami     is an associate professor in management information systems at Dhofar University, and he serves on different administration positions and committees at Dhofar University. He has many research papers in the field. His research interests include management information systems, knowledge management, computers for society, computer vision, and digital entrepreneurship. He received several research funds in Oman. He is also a member of the editorial board/scientific committees of several peer-reviewed research journals and conferences. He can be contacted at email: samir@du.edu.om.



Muhammad Alhammami     is a researcher and lecturer at the Higher Institute for Applied Sciences and Technology in Damascus, Syria. His research interests include communication, SoC-FPGA and embedded systems, artificial intelligence and machine learning, image and signal processing, and uPC. He has published several articles in journals and conferences, such as *Advances in Visual Informatics*, *Data in Brief*, *MethodsX*, and *Journal of Asian Finance, Economics and Business*. Muhammad Alhammami's CV shows that he has made significant contributions to his field of study. He can be contacted at email: Muhammad.Alhammami@hiast.edu.sy or dr.mhammami@outlook.com.