BAREKENG: Journal of Mathematics and Its Applications
December 2026 Volume 20 Issue 4 Page 3489-3500
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss4pp3489-3500

DESIGN AND IMPLEMENTATION OF A SMART OUTPASS MANAGEMENT SYSTEM USING FACE RECOGNITION

S. Aasha Nandhini[1*], G. M. Deyanesh Krishna[2], Malathy Batumalay[3]

[1,2]*Department of Electronics and Communication Engineering, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, 603110, India.*

[3]*Faculty of Information Technology, INTI International University*
[3]*Centre for Data Science and Sustainable Technologies, INTI International University Nilai, N. Sembilan, 71800, Malaysia*

*Corresponding author's e-mail: * **aashanandhinis@ssn.edu.in***

Article Info

Article History:

Received: 05th November 2025
Revised: 30th April 2026
Accepted: 07th June 2026
Available online: 24th August 2026

Keywords:

Deep learning;
Energy–accuracy tradeoff;
Face recognition;
IoT;
Mathematical optimization;
Raspberry Pi.

ABSTRACT

In educational institutions, manual outpass verification commonly fails because it is inefficient, relies on people, and is prone to counterfeit. This leads to delays in administration and security breaches. Standard digital substitutes that use barcodes or RFID technology only partially automate the process and don't include biometric validation. To solve these problems, an intelligent web-enabled framework is being built that uses deep learning-based face recognition and connects to the Internet of Things infrastructure. The system uses OptimizedEdgeNet, a new, lightweight convolutional neural network that is developed for embedded edge devices like the Raspberry Pi. A mathematical optimization approach is proposed that takes into account recognition accuracy, decision threshold, computing latency, and energy use. The proposed methodology employs Gaussian distance modelling of facial embeddings to analytically find the ideal matching threshold (τ^*) that lowers the total recognition error. Additionally, equations for latency and energy are developed as factors of operation count, processor frequency, and hardware efficiency, creating a multi-objective optimization function that enhances recognition performance while reducing delay and power consumption. The analytical model is confirmed by experimental testing, which shows that the Raspberry Pi 4B can recognize 96.2% of the time with an average delay of 1.64 seconds and an energy use of 2.4 J per inference. The suggested deep learning architecture and mathematical formulation offer a thorough, real-time, and resource-efficient approach for automating secure digital outpasses in regulated settings.

How to cite this article:

S. A. Nandhini, G. M. D. Krishna and M. Batumalay., "DESIGN AND IMPLEMENTATION OF A SMART OUTPASS MANAGEMENT SYSTEM USING FACE RECOGNITION", *BAREKENG: J. Math. & App.*, vol. 20, no. 4, pp. 3489-3500, Dec, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

In educational institutions, outpass management systems have always been manual, paper-based operations that depend on staff validation and retaining physical records. These kinds of systems take a lot of time and are also easy to abuse, fabricate, and make mistakes. As more and more administrative tasks are done digitally, there is a pressing need for safe, automated, and open ways to handle student movement authorisation. There have been a few digital techniques suggested, like RFID-based and QR code-based systems, however these methods are limited because they rely on outside tokens and can't validate identify biometrically [1], [2]. Facial recognition is a non-intrusive and very accurate way to identify people [3], [4]. It could be a good way to automate identity-based tasks in locations like colleges, hostels, and companies.

A lot of the time, Internet of Things (IoT)-enabled facial recognition systems focus on implementation instead of analytical optimisation [5], [6]. Most of them use pre-trained architectures like Dlib or MobileNet for inference. These work well, but they aren't made for hardware that doesn't have a lot of processing power, like the Raspberry Pi. This makes it hard to get real-time performance while yet keeping high accuracy and low energy use. Also, traditional systems don't have a way to connect recognition accuracy, decision threshold, processing delay, and energy consumption in a quantitative way. These are all important for using embedded intelligence in edge computing contexts.

Numerous researchers have enhanced facial recognition on embedded devices [7]. The authors in [8] created a face detection system utilising Haar and LBP algorithms on a Raspberry Pi. The system was moderately accurate but not very robust when the lighting changed. In [1] the authors suggested a contactless attendance system utilising Raspberry Pi and artificial intelligence, whereas in [9] the authors offered a multitask learning framework for real-time facial recognition on Raspberry Pi. Concentrated on model performance without considering energy or latency trade-offs. The authors in [10] created a door lock system that uses deep learning and a Raspberry Pi. The authors in [11] used Dlib to develop a door lock system that showed that facial verification could be used for safe access control. The authors in [12] created a system for managing automatic gate passes that uses the web and the IoT. Concentrated on application process devoid of biometric authentication. Nikhil et al. suggested a smartphone app for managing hostels that would use IoT services for students. Showed how to maintain digital records that are important to campus systems [13]. The authors in [14] were the first to use the Haar cascade classifier for real-time object detection. It is a classic example of a modern deep-learning-based recognition system. While these investigations demonstrated functional viability, they lacked a mathematical performance characterisation or optimisation model to analytically adjust system parameters for embedded operation.

This study provides a web-enabled outpass management framework that combines IoT-based automation with a new lightweight deep learning model called OptimizedEdgeNet (QEN). This model was made for edge devices like the Raspberry Pi. The suggested methodology differs from typical implementations by integrating a mathematically optimised deep learning architecture with an analytical performance model that encompasses identification accuracy, threshold tuning, latency, and energy efficiency within a cohesive optimisation framework. Gaussian distance modelling of embeddings is used to find the best recognition threshold (τ^*) that reduces classification mistakes. Analytical formulae for processing delay and energy consumption show how efficient the system is based on the computing load and hardware frequency.

This work makes four important contributions. First, a web-enabled outpass verification system that works in real time is created. It uses deep learning-based face recognition to automate authorisation. Second, QEN is a new lightweight Convolutional Neural Network (CNN) architecture that is meant to work well even when there are limits on how much computing power is available. Third, a mathematical optimisation model is created to find the best balance between accuracy, latency, and energy costs using closed-form analytical expressions. Finally, the suggested models are tested on a Raspberry Pi platform, showing that they are quite accurate, have shorter inference times, and that the theoretical predictions and experimental outcomes are very similar. This integrated framework turns traditional IoT-based identification systems into embedded solutions that are mathematically optimised and aware of resources. This makes it a scalable, safe, and effective way to administer a digital campus and other real-time access control applications.

Facial recognition is used for identification as it is less invasive, non-contact, and user-friendly compared to other methods such as fingerprints or iris scanning. Moreover, facial recognition technology does not need any contact with the human body; therefore, it is well-suited for access control systems that have variable scenarios in real-time applications. As mentioned above, Raspberry Pi 4B is selected as the deployment platform due to its low cost, portability, and ability to handle light-weight deep learning algorithms. The device is well-suited as an edge computing device for designing real-time intelligent systems as it can handle camera modules and is easy to implement in IoT applications.

2. RESEARCH METHODS

2.1 System Overview

The suggested solution uses IoT, web technologies, and deep learning to automate the process of managing outpasses in education institutions. The architecture is based on a three-tier client-server approach that includes a web interface, an embedded processing unit, and a centralized database. The web server that runs on Flask serves as the middleware that lets users talk to the embedded device. Students send in requests for digital outpasses through a secure login site, and staff members approve these requests through their own dashboards. A central MySQL database keeps track of approved records. At the entrance to the school, a Raspberry Pi 4B with a Pi Camera Version 2 takes a picture of the student's face. Using QEN, a lightweight CNN made for limited spaces, the embedded processor runs the deep learning-based facial recognition algorithm. The system compares facial embeddings to the items in the database and then produces either "Access Granted" or "Access Denied" based on how similar the two are. This modular design as depicted in Fig. 1 allows for real-time verification, reduces the need for human interaction, and keeps data safe by making sure that databases talk to each other at the same time.

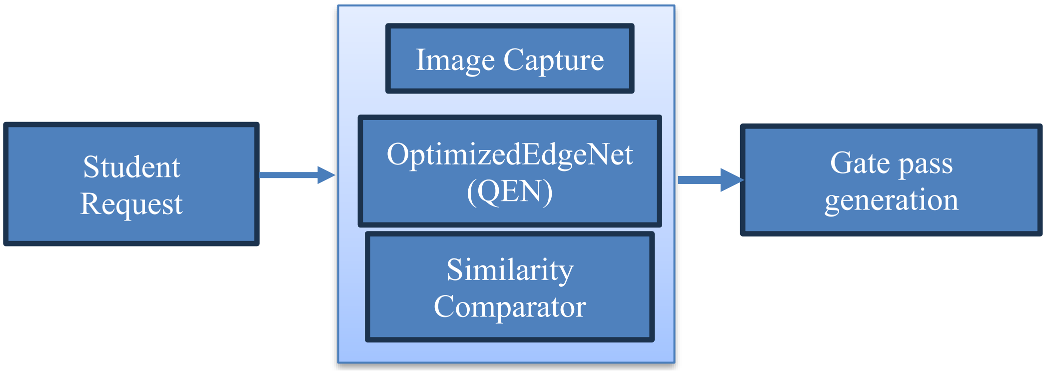


Figure 1. Overall Block Diagram of Smart Gate Pass Management System using QEN

2.2 OptimizedEdgeNet

A novel deep learning model called OptimizedEdgeNet (QEN) was built to allow real-time inference on devices with restricted resources. The model is based on lightweight architectures like MobileNet and SqueezeNet, however it has been changed to work better on Raspberry Pi [15], [16], [17]. Compared to regular convolutional networks, QEN uses depthwise separable convolutions to cut down on the number of parameters and floating-point operations by around 70%. The network architecture consists of the following layers:

1. Input layer: 128 × 128 grayscale image.
2. Convolution block 1:32 filters, 3×3 kernel, ReLU6 activation, followed by batch normalization.
3. Depthwise convolution block 2:64 filters, stride 2, quantized to 8-bit precision.
4. Convolution block 3:128 filters, ReLU6 activation.
5. Global average pooling layer to minimize feature dimensionality.
6. Fully connected dense layer: 128 outputs representing the facial embedding vector.

The use of ReLU6 activation ensures better gradient control on low-power processors, while 8-bit quantization reduces memory usage during inference [18], [19]. Each facial embedding vector represents an individual in a 128-dimensional feature space. Verification between two embeddings x_i and x_j is performed using Euclidean distance as shown in Eq. (1).

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{128} (x_{ik} - x_{jk})^2}. \tag{1}$$

A decision is made using a similarity threshold τsuch that in Eq. (2).

$$\hat{y} = \begin{cases} 1, & d(x_i, x_j) \leq \tau \\ 0, & d(x_i, x_j) > \tau \end{cases}. \tag{2}$$

The proposed QEN algorithm is different from other lightweight neural networks such as MobileNet and SqueezeNet as its design principle mainly concentrates on edge deployment, which includes architecture and hardware optimization. Although MobileNet also uses depthwise-separable convolutions, QEN has made the neural network more efficient by lowering the number of computations through minimizing the number of parameters, greyscale inputs, and 8-bit quantization. Hence, it is more efficient when used on Raspberry Pi. On the other hand, QEN tries to find the optimal point between the representation and inference, whereas SqueezeNet is all about on reducing the number of parameters using fire modules. One of the main differences between this proposed system and typical deep learning-based recognition systems is that it incorporates analytical threshold optimization, which provides a mathematically based way to optimize the trade-off between accuracy, latency, and energy consumption. These design decisions together will result in an efficient recognition system that is more appropriate for real-time edge-based face recognition applications.

2.3 Mathematical Modelling and Derivations

The facial embedding distances for genuine pairs and impostor pairs are modelled as Gaussian random variables as shown in Eq. (3).

$$d_s \sim N(\mu_s, \sigma_s^2), d_d \sim N(\mu_d, \sigma_d^2). \tag{3}$$

The probabilities of false acceptance $P_{FA}$and false rejection $P_{FR}$are expressed using Eqs. (4) and (5).

$$P_{FA}(\tau) = 1 - \Phi\left(\frac{\tau - \mu_d}{\sigma_d}\right), \tag{4}$$

$$P_{FR}(\tau) = 1 - \Phi\left(\frac{\tau - \mu_s}{\sigma_s}\right), \tag{5}$$

where $\Phi(\cdot)$denotes the standard normal cumulative distribution function.

The optimal threshold $\tau^*$is derived by minimizing the total classification error as shown in Eq. (6).

$$\tau^* = \arg\min[\alpha P_{FA}(\tau) + (1 - \alpha) P_{FR}(\tau)]. \tag{6}$$

Solving equilibrium between both error components yields the closed-form expression as shown in Eq. (7).

$$\tau^* = \frac{\sigma_d \mu_s + \sigma_s \mu_d}{\sigma_d + \sigma_s}. \tag{7}$$

The overall recognition accuracy $A(\tau)$as a function of threshold $\tau$is given by Eq. (8).

$$A(\tau) = 1 - [P_{FA}(\tau) + P_{FR}(\tau)]. \tag{8}$$

Here, α represents the system's security weighting coefficient higher values of α prioritize minimizing false acceptance at the cost of longer verification time.

2.4 System Delay and Energy Models

The total system delay T_{total} consists of four main components: image capture time (T_{cap}), neural network processing time (T_{proc}), database query time (T_{DB}), and communication latency (T_{comm}), given in Eq. (9).

$$T_{total} = T_{cap} + T_{proc} + T_{DB} + T_{comm}. \tag{9}$$

The processing delay of the deep learning model as shown in Eq. (10) depends on the number of arithmetic operations N_{ops}, the processor frequency f, and hardware efficiency η:

$$T_{proc} = \frac{N_{ops}}{f \times \eta}. \tag{10}$$

Substituting Eq. (10) into Eq. (9) gives Eq. (11)

$$T_{total} = T_0 \frac{N_{ops}}{f \times \eta}. \tag{11}$$

The energy consumption per inference is derived from the power–frequency relationship of the processor. The instantaneous dynamic energy is expressed using Eq. (12)

$$E = V^2 \times C \times f \times T_{proc}. \tag{12}$$

Substituting Eq. (10) into Eq. (12) gives Eq. (13)

$$E = V^2 \times C \times \frac{N_{ops}}{\eta}. \tag{13}$$

These equations form the analytical foundation for evaluating the computational performance and energy trade-offs of the embedded recognition system.

2.5 Multi-Objective Optimization Framework

The overall system efficiency is defined through a multi-objective optimization function that balances recognition accuracy ($A(\tau)$), latency ($T_{total}(f)$), and energy consumption ($E(f)$). The objective function is expressed using Eq. (14).

$$J(\tau, f) = w_1 A(\tau) - w_2 T_{total}(f) - w_3 E(f). \tag{14}$$

where $w_1, w_2, w_3$are the weighting coefficients representing the relative importance of accuracy, delay, and power efficiency.

The optimization is performed within the feasible domain of τ and f as expressed in Eq. (15).

$$\tau \in [\mu_s, \mu_d], f \in [f_{min}, f_{max}]. \tag{15}$$

The optimal configuration $(\tau^*, f^*)$is obtained by maximizing $J(\tau, f)$, which yields the most efficient trade-off between accuracy, speed, and power on the Raspberry Pi hardware.

3. RESULTS AND DISCUSSION

The suggested web-based outpass management system was successfully set up and tested on the Raspberry Pi 4B platform. Both the analytical models and the experimental findings were analyzed to see if the mathematical formulas worked for accuracy, latency, and energy optimization. The following discussion compares projected and measured system performance.

3.1 Experimental Setup

The dataset used for testing is composed of resources obtained from users' faces in a controlled campus setting. Approximately 1000 images were captured to obtain faces and some of these images were taken under different lighting conditions (daytime, nighttime) and using different poses and expressions. Each of the images was taken with a Raspberry Pi Camera Module at a resolution of 128 × 128 and were converted into grayscale images before being processed. The training and testing images were split into different training and test sets by a 80:20 ratio so that the images could be fairly evaluated. In order to ensure the results of the test were based on completed images, the images received face detection and were aligned as part of the pre-processing process. Additionally, the samples were normalized to ensure uniformity across the dataset. The system is intended to process each frame of video in real-time rather than as continuous video streams. The analytical models were evaluated using the parameter values $\mu_s = 0.35$, $\mu_d = 0.65$, $\sigma_s = 0.07$, $\sigma_d = 0.08$, and $\alpha = 0.7$. The optimal threshold (τ*) computed from the mathematical formulation was found to be 0.47, which corresponds to the point of minimum total classification error. The experimental findings validated this prediction, demonstrating that the system attained maximum accuracy at the identical threshold value. Table 1 shows a side-by-side comparison of the analytical and experimental data. The average difference between the theoretical and measured values stayed around 5%, which means that the model and real-world behaviour were very similar.

Table 1. Comparison Between Analytical Predictions and Experimental Results of the Proposed QEN Based System.

Parameter	Analytical Model	Experimental Value	Deviation (%)
Optimal threshold (τ*)	0.47	0.47	0
Recognition accuracy	96.5%	96.2%	0.3
Average latency	1.61 s	1.64 s	1.8
Energy per inference	2.3 J	2.4 J	4.3

3.2 Performance Evaluation Metrics

To provide a comprehensive evaluation of the proposed system, additional performance metrics are considered beyond overall accuracy. These include precision, recall, F1-score, confusion matrix analysis, and error-based measures such as False Acceptance Rate (FAR) and False Rejection Rate (FRR). These metrics provide deeper insight into system reliability under real-world operating conditions.

3.2.1 Confusion Matrix Analysis

The performance of the face recognition system is analysed using a confusion matrix consisting of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Here, TP represents correctly authenticated users, FP indicates unauthorized users incorrectly granted access, FN represents valid users incorrectly denied access, and TN denotes correctly rejected unauthorized users.

Table 2. Confusion Matrix of the Proposed System

	Predicted Positive	**Predicted Negative**
Actual Positive	TP = 481	FN = 19
Actual Negative	FP = 14	TN = 486

3.2.2 Performance Metrics

The standard evaluation metrics are computed using Eqs. (16)-(18).

$$Precision = \frac{TP}{TP+FP}, \tag{16}$$

$$Recall = \frac{TP}{TP+FN}, \tag{17}$$

$$F1 - Score = \frac{2. Precision. Recall}{Precision + Recall}. \tag{18}$$

It is observed from Table 2, the system achieves a precision of 97.17%, a recall of 96.20%, and an F1-score of 96.68%. These results indicate that the proposed system maintains a strong balance between correctly identifying authorized users and minimizing incorrect access decisions, thereby ensuring reliable performance in real-time access control scenarios.

3.2.3 Error Rate Analysis (FAR and FRR)

To further evaluate system reliability, the FAR and FRR are computed using Eqs. (19) and (20).

$$FAR = \frac{FP}{FP+TN}, \tag{19}$$

$$FRR = \frac{FN}{TP+FN}. \tag{20}$$

From the observed results, the FAR is 2.80% and the FRR is 3.80%. These error rates are directly influenced by the decision threshold (τ). While accuracy provides an overall measure, precision and recall offer deeper insights into classification behaviour. The low FAR and FRR values further confirm that the proposed system achieves reliable and secure identity verification suitable for real-time access control applications.

3.3 Accuracy–Threshold Relationship

The accuracy of recognition changes in a bell-shaped pattern depending on the similarity threshold, with the highest point at $\tau = 0.47$. When the criteria are lower, the system is too stringent, which causes more erroneous rejections. On the other hand, larger thresholds make the decision boundary less strict, which leads to more false acceptances. The analytical model effectively encapsulates this trade-off and forecasts the ideal operational zone where both error probabilities are equilibrated. Fig. 2 shows that accuracy goes up quickly when τ gets closer to its best value and then slowly goes down after that. This proves that the threshold obtained from the Gaussian distance model is the ideal setting for the recognition system to work in. This behavior occurs because lower threshold values impose stricter matching conditions, increasing false rejections, while higher threshold values relax the decision boundary, leading to higher false acceptance. The optimal threshold balances these competing effects, resulting in maximum recognition accuracy.

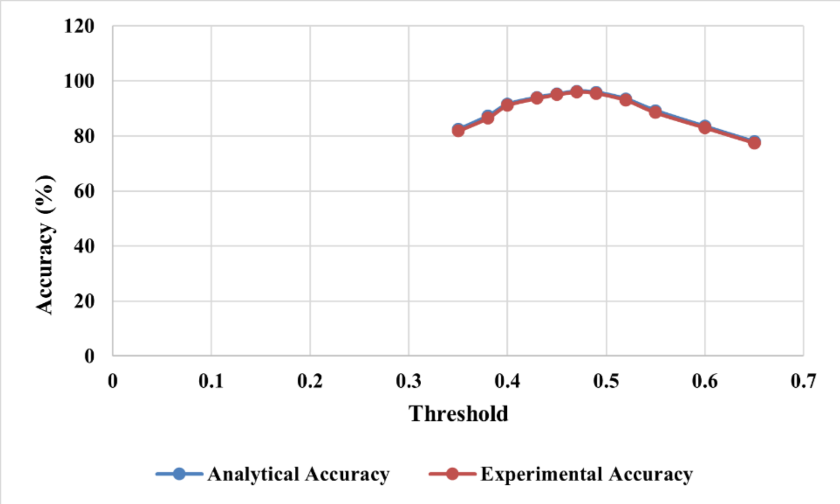


Figure 2. Recognition accuracy versus similarity threshold showing the analytical–experimental correlation and optimal threshold (τ = 0.47)*

3.4 Latency–Frequency Relationship

The total delay for a range of processor frequencies, from 1.0 GHz to 1.5 GHz was assessed and the results are depicted in Fig. 3. The trend that was seen reveals that latency and frequency are related in the opposite way, which is what the analytical formulation says. The model processes image embeddings faster at higher clock speeds because each operation takes less time to compute. But the rate of improvement slows down after 1.5 GHz since the system is getting close to hardware saturation. This graph shows this relationship, with latency dropping quickly up to 1.3 GHz and then levelling out slowly. The testing results show that going above 1.5 GHz doesn't give much of a benefit and uses more energy, hence 1.5 GHz is the best frequency for real-time use.

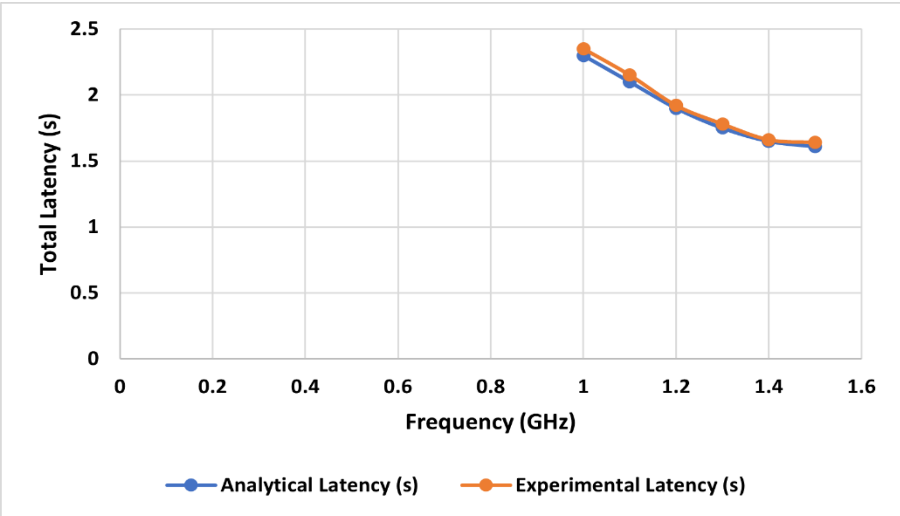


Figure 3. Variation of Total System Latency with Processor Frequency (GHz) Demonstrating Inverse Proportionality and Hardware Saturation Beyond 1.5 GHz

3.5 Energy vs Frequency Relationship

The same frequency band was used to measure energy use. The increase in energy consumption with frequency is attributed to the proportional relationship between dynamic power and operating frequency. This highlights the trade-off between faster processing and higher energy usage, reinforcing the need for optimization. The results show that energy increases almost linearly with clock frequency, which is what you would expect since dynamic power and operating frequency are proportionate. Fig. 4 shows that energy use goes up from about 1.9 J at 1.0 GHz to 2.4 J at 1.5 GHz. Higher frequencies lower delay, but they also use more energy. This shows how important the proposed optimization approach is, which looks at both energy and time trade-offs to get the best results. The experimental data corroborates analytical expectations, validating that energy and delay may be efficiently balanced via mathematical optimization.

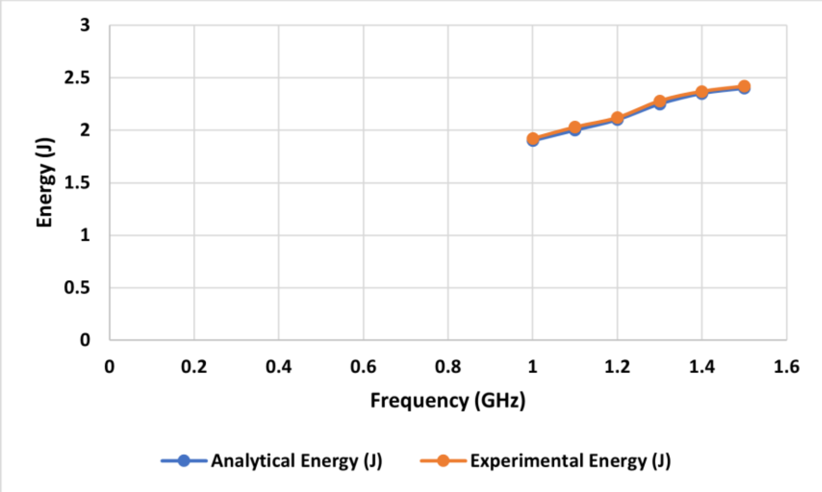


Figure 4. Energy Consumption per Inference versus Processor Frequency Illustrating Linear Growth due to Dynamic Power Dependence.

3.6 Optimization and Trade-off Analysis

The optimal operating point demonstrates that achieving maximum accuracy does not necessarily correspond to minimum energy consumption, thereby emphasizing the importance of multi-objective optimization. By evaluating the multi-objective function for accuracy, latency, and energy, the overall system performance index (J) was maximized for $\tau = 0.47$ and $f = 1.5$ GHz. These parameters provide the best balance between computational efficiency and recognition reliability. The experimentally validated results achieving Recognition accuracy: 96.2%, Average latency: 1.64 seconds and Energy per inference: 2.4 joules outperform existing work reported in [20].

3.7 Comparison with Existing Methods

To validate the effectiveness of the proposed QEN model, its performance is compared with existing face recognition approaches commonly used in embedded systems. The comparison focuses on key performance indicators such as recognition accuracy, inference latency, and computational efficiency.

Table 3. Comparison with Existing Models

Method	Accuracy (%)	Latency (s)	Energy per Inference (J)
MobileNet	94.0	1.85	2.80
SqueezeNet	93.2	1.92	2.95
Proposed QEN	96.2	1.64	2.40

The proposed QEN model is compared with widely used lightweight architectures such as MobileNet and SqueezeNet, which are commonly deployed in edge-based vision systems. As shown in Table 3, the proposed model achieves higher recognition accuracy while maintaining lower latency and energy consumption. This improvement is primarily due to the use of depthwise separable convolutions, quantization, and architecture optimization specifically tailored for embedded platforms. These results demonstrate that the proposed model provides a better trade-off between accuracy and computational efficiency compared to existing edge-based approaches.

3.8 Implementation Results

The Web Enabled Outpass Management System using face recognition has been implemented, successfully showcasing key functionalities of the web application. As shown in Fig. 5, The login page designed for both students and staff, allows secure authentication and directs users to their respective dashboards. As shown in Figure 6, The student dashboard provides an interface to view the status of outpass requests and submit new requests through a user-friendly request form. On the other hand, the staff dashboard enables staff members to view, approve, or reject requests efficiently.

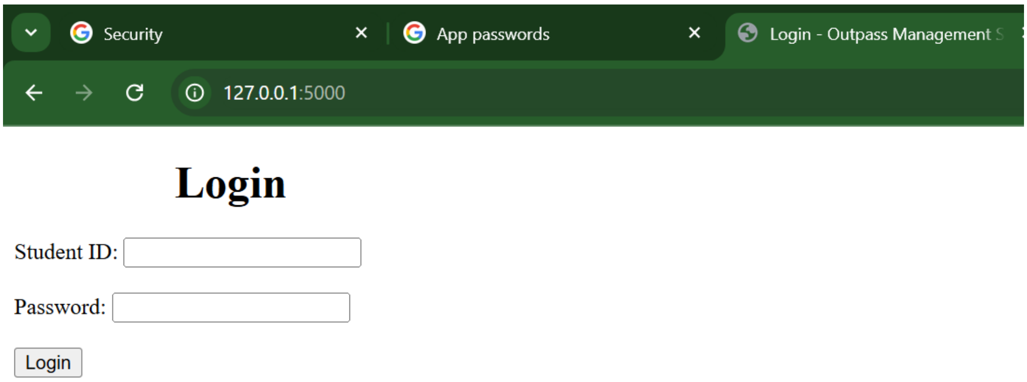


Figure 5. Web Application Login Interface for Students and Staff Users.

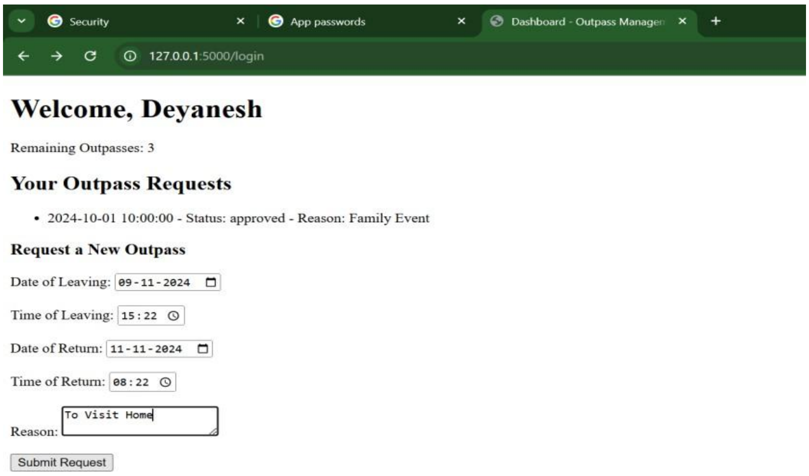
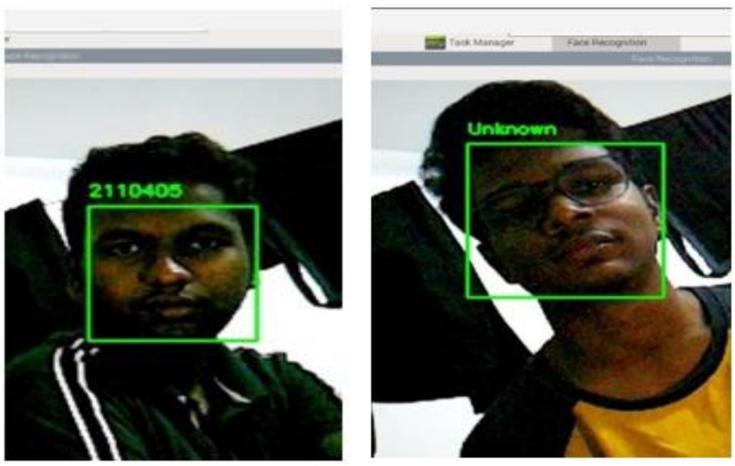


Figure 6. Student dashboard for submitting and tracking outpass



(a) (b)

Figure 7 (a) Authorizing the student and (b) Access denial for unregistered or unauthorized user

Fig. 7 shows the test results that includes recognizing a student (ID: 2110405) and denying access to an unknown person respectively, demonstrating the system's ability to differentiate authorized users from unauthorized ones.

4. CONCLUSION

In this study a mathematically-optimized and empirically validated web-enabled outpass management system is presented that integrates an IoT infrastructure with a deep learning-based facial recognition engine. The presented solution aims to overcome the issues of manual verification by providing a secure, automated, real-time solution for managing gate level identity. It is facilitated through a novel, light-weight QEN architecture that enable the accuracy of facial recognition at resource constrained devices like Raspberry Pi with lower power and latency constraints. The mathematical model for the accuracy, latency and power consumed was developed and was found to effectively predict and facilitate the optimization of system parameters. The derived analytical calculation for optimal similarity threshold provide a robust methodology to decrease the classification error, while the multi-objective function balances the accuracy-latency-power consumption trade off. An experimental validation shows that the proposed model predicted system parameters with considerable accuracy with 96.2 % recognition rate and average delay of 1.64 s with power consumed of 2.4 Joules per inference approximately matching the model. It appears that a synergy of deep learning and mathematical modeling can improve the predictability and performance of an embedded biometric system. Hence a normal IoT application becomes a smart, measurable and flexible system capable of efficient execution even on a constrained hardware. The further improvement is possible by making system scalable for widespread applications, reducing computation requirement by edge-cloud hybrid processing and to reduce the computational requirement on edge device by making adaptive threshold learning from live input. Also considering the possibility of FPGA or TPU acceleration can improve the processing capability significantly and can add the predictive analytics for surveillance of activities and analysis of access patterns.

Author Contributions

G. M. Deyanesh Krishna: Conceptualization, Software Development, Methodology Design, System Implementation, Data Collection, Formal Analysis, Validation, and Original Draft Preparation. S. Aasha Nandhini: Supervision, Project Administration, Technical Guidance, Formulation of Mathematical Models, Performance Analysis, And Critical Review and Editing of The Manuscript. R. Malathy: Data Curation, Validation of Analytical Models, Experimental Verification, Literature Review Support, and Manuscript Refinement During Revision. All authors discussed the results, contributed to the interpretation of findings, and approved the final version of the manuscript for publication.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declarations

The authors declare that there are no competing interests regarding the publication of this manuscript.

Declaration of Generative Ai and Ai-Assisted Technologies

The authors used generative AI (ChatGPT) only to assist with language polishing and formatting consistency (e.g., improving wording and ensuring uniform terminology). No AI was used to generate research content, perform analyses, or create/modify figures and tables. The authors reviewed the manuscript in full and remain responsible for its content.

REFERENCES

[1] Khushi Verma, Khushi Gupta, Divya Kumawat, and Gajendra Singh Rajput, "CONTACTLESS ATTENDANCE MANAGEMENT USING AI-BASED FACE RECOGNITION," International Journal on Science and Technology, vol. 16, no. 2, Jun. 2025. doi: https://doi.org/10.71097/IJSAT.v16.i2.5162.

[2] C.-W. Lien and S. Vhaduri, "CHALLENGES AND OPPORTUNITIES OF BIOMETRIC USER AUTHENTICATION IN THE AGE OF IOT: A SURVEY," *ACM Comput Surv*, vol. 56, no. 1, pp. 1–37, Jan. 2024. doi: https://doi.org/10.1145/3603705.

[3] M. Masud *et al.*, "DEEP LEARNING-BASED INTELLIGENT FACE RECOGNITION IN IOT-CLOUD ENVIRONMENT," *Comput Commun*, vol. 152, pp. 215–222, Feb. 2020. doi: https://doi.org/10.1016/j.comcom.2020.01.050.

[4] H. Ge, Z. Zhu, Y. Dai, B. Wang, and X. Wu, "FACIAL EXPRESSION RECOGNITION BASED ON DEEP LEARNING," *Comput Methods Programs Biomed*, vol. 215, p. 106621, Mar. 2022. doi: https://doi.org/10.1016/j.cmpb.2022.106621.

[5] Y. Xie, P. Li, N. Nedjah, B. B. Gupta, D. Taniar, and J. Zhang, "PRIVACY PROTECTION FRAMEWORK FOR FACE RECOGNITION IN EDGE-BASED INTERNET OF THINGS," *Cluster Comput*, vol. 26, no. 5, pp. 3017–3035, Oct. 2023. doi: https://doi.org/10.1007/s10586-022-03808-8.

[6] W. Yang, S. Wang, N. M. Sahri, N. M. Karie, M. Ahmed, and C. Valli, "BIOMETRICS FOR INTERNET-OF-THINGS SECURITY: A REVIEW," *Sensors*, vol. 21, no. 18, p. 6163, Sep. 2021. doi: https://doi.org/10.3390/s21186163.

[7] X. Lv, M. Su, and Z. Wang, "APPLICATION OF FACE RECOGNITION METHOD UNDER DEEP LEARNING ALGORITHM IN EMBEDDED SYSTEMS," *Microprocess Microsyst*, p. 104034, Jan. 2021. doi: https://doi.org/10.1016/j.micpro.2021.104034.

[8] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, "FACIAL RECOGNITION USING HAAR CASCADE AND LBP CLASSIFIERS," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330–335, Nov. 2021. doi: https://doi.org/10.1016/j.gltp.2021.08.044.

[9] A. A. A. Aboluhom and I. Kandilli, "REAL-TIME FACIAL RECOGNITION VIA MULTITASK LEARNING ON RASPBERRY PI," *Sci Rep*, vol. 15, no. 1, p. 28467, Aug. 2025. doi: https://doi.org/10.1038/s41598-025-97490-6.

[10] S. S. F. A. Elnozahy, S. C. Pari, and L. C. Liang, "RASPBERRY PI-BASED FACE RECOGNITION DOOR LOCK SYSTEM," *IoT*, vol. 6, no. 2, p. 31, May 2025. doi: https://doi.org/10.3390/iot6020031.

[11] T. G. Reddy, S. C. Sai, B. P. Kumar, R. T. Venkatesh, K. Sathwik, and K. Singh, "FACE RECOGNITION DOOR LOCK SYSTEM USING RASPBERRY PI," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, May 2023, pp. 218–221. doi: https://doi.org/10.1109/ICSCCC58608.2023.10176434.

[12] Vaddeboyina Sri Manvith, S. Madhunala, and B. V. Kiranmayee, "GATEZEE—AN AUTOMATED GATE PASS MANAGEMENT SYSTEM," 2022, pp. 385–395. doi: https://doi.org/10.1007/978-981-16-7389-4_36.

[13] L. S. Nikhil, P. Lavanyakumari, S. Balakrishna, Ch. Kiran, and Ch. A. Reddy, "THE ULTIMATE GUIDE TO EXPLORE YOUR HOSTEL THROUGH HOSTEL APP," 2025, p. 020049. doi: https://doi.org/10.1063/5.0265791.

[14] P. Viola and M. Jones, "RAPID OBJECT DETECTION USING A BOOSTED CASCADE OF SIMPLE FEATURES," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE Comput. Soc, pp. I-511-I–518. doi: https://doi.org/10.1109/CVPR.2001.990517.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "RETHINKING THE INCEPTION ARCHITECTURE FOR COMPUTER VISION." in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2818–2826. doi: https://doi.org/10.1109/CVPR.2016.308.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MOBILENETV2: INVERTED RESIDUALS AND LINEAR BOTTLENECKS," Mar. 2019, [Online]. Available: http://arxiv.org/abs/1801.04381 . doi: https://doi.org/10.1109/CVPR.2018.00474.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS," *Commun ACM*, vol. 60, no. 6, pp. 84–90, May 2017. doi: https://doi.org/10.1145/3065386.

[18] S. Ioffe and C. Szegedy, "BATCH NORMALIZATION: ACCELERATING DEEP NETWORK TRAINING BY REDUCING INTERNAL COVARIATE SHIFT," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., in Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, Nov. 2015, pp. 448–456. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html

[19] M. Segu, A. Tonioni, and F. Tombari, "BATCH NORMALIZATION EMBEDDINGS FOR DEEP DOMAIN GENERALIZATION," *Pattern Recognit*, vol. 135, p. 109115, Mar. 2023. doi: https://doi.org/10.1016/j.patcog.2022.109115.

[20] S. Bhutada, S. Mekala, M. Gandham, R. Bhat, and R. Upadhyayula, "FACE RECOGNITION BASED GATE PASS SYSTEM," JETIR, vol. 9, no. 12, pp. 102–106, 2022.