

Development of a Facial Recognition Model using Optimized Convolution Neural Network with Tiny-ML and Fingerprint Authentication for Safe Lock Entry

Oluwaseun Opeyemi Martins^{1*}, Adetoye Ayokunle Aribisala², Muhammad Mahdi Abdulhamid³, Idowu Oluwatimileyin Joshua⁴, Olaoye John Oluwatobi⁵, Olumoko Ronald Samuel⁶

¹Department of Mechatronics Engineering,
Federal University,
Oye-Ekiti,
Nigeria

Email: oluwaseun.martins@fuoye.edu.ng

Abstract

Safe boxes have been used for centuries to protect valuable items, and their design and construction have evolved over time to improve security. In this study, we carried out the design and implementation of a facial recognition technology using optimized convolutional neural networks (CNNs) with TinyML and fingerprint. The facial recognition system is designed using pre-trained CNN architectures then optimized with TinyML to make it fit into resource constrained edge device, the fingerprint recognition is designed using ridge matching biometric algorithm which fits in perfectly into the microcontroller-based system. The safe box was fabricated using sheet metal and insulated using heat resistive material (fiber glass), components like OV2640 camera module, ESP32-S3, R305 Fingerprint sensor, 4.3 inches touch screen TFT display and a charging module to manage and charge the systems battery were used. The system was implemented to be a level two security and two-way verification system using secured Wi-Fi to access the system before the facial and fingerprint recognition functionality is activated, the sequence of operation can either be facial recognition before fingerprint recognition or fingerprint recognition before facial recognition. Evaluating the system's facial recognition performance with confusion matrix, of all the optimized CNN architectures like VGGNet, SENet, ResNet, MobileNet-V2, EfficientNet-B0, SENet Architecture performed best outputting an accuracy of 98%, precision of 92%, also SENet unoptimized model performed same, MobileNet-V2 performed optimally in the inference speed test with a speed of 15ms/images with the fingerprint also evaluated using confusion matrix and perform with 98% accuracy. It was noticed that the inference speed of the unoptimized model is faster than that of the TinyML optimized model which is justifiable based on the processing clock speed, memory usage, GPU and architecture of the microcontroller/microprocessor. From the result obtained TinyML proves to be suitable mean for edge AI (artificial intelligence) and is recommended for future research and development.

Keywords – TinyML, Facial Recognition, Convolutional Neural Network, Fingerprint Authentication, Level-Two Security.

INTRODUCTION

Throughout history, people have safeguarded important documents from fire damage using various methods. The Assyrians, Egyptians, and Greeks employed burial techniques, while Julius Caesar used iron boxes, though the metal conducted heat. Feudal lords opted for

*Author for Correspondence

underground vaults with guards, and 15th-century merchants devised iconic oak treasure chests banded with iron for protection against fire (Smith, 2020; Hayward, 2019). As this birthed the use of safe lock which is important for protecting valuables against fire and other environment hazards (Setyadi *et al.*, 2020). With various lock options available, including key locks, combination locks, and biometric technologies like fingerprint recognition (Maltoni *et al.*, 2019), facial recognition technology has also emerged as a popular and effective method in recent years, finding applications in security systems (Darwish *et al.*, 2021; Singh *et al.*, 2023). While facial recognition technology has been used in a variety of applications, it is still a relatively new technology in the field of safe locks. There are several key challenges associated with designing and implementing facial recognition safe locks, including the need for accurate facial recognition algorithms due to their reliance on 2D images and limited ability to distinguish between similar faces, the need for robust security protocols, and the need to address privacy concerns (Cheng, 2022; Ooi *et al.*, 2023; Fauziah *et al.*, 2022). Using level-one security and one-way verification as it is in traditional safe lock with access technology like facial recognition, finger print recognition, bar-code, RFID0, and password key unlocking which are still effective at protecting against theft and fire, they can be vulnerable to hacking or tampering (Elechi, *et al.* 2022). However, traditional implementation of these facial and fingerprint technologies and algorithms demands significant computational resources, rendering them unsuitable for resource-constrained devices or edge devices.

The aim of the project is to develop a facial recognition model using optimized convolution neural network with Tiny-ML and fingerprint authentication for safe lock entry (Howard *et al.*, 2017; Wu *et al.*, 2017). This is accomplished by designing an optimized facial recognition system using different pre-trained CNN model with TinyML, deploying a fingerprint recognition model using ridge matching biometric algorithm, Implementing the optimized facial recognition CNN model and fingerprint model on a resource constrained microcontroller for a facial and fingerprint safe lock access control (Sandeep and Jagadeesh, 2019), and evaluate the performance and resource usage of the fingerprint recognition model and of the optimized CNN model using TinyML for facial recognition, and compare it with a non-optimized CNN model (Ignatov *et al.*, 2018).

The ability to perform facial and fingerprint recognition on resource-constrained devices like microcontrollers would significantly expand the potential applications of this technology, enabling deployment in areas with limited computing power or network connectivity (Warden and Situnayake, 2019). This potential is realized by combining Convolutional Neural Networks (CNNs) with TinyML, which allows for achieving high accuracy while maintaining low resource usage, making facial recognition systems feasible on microcontrollers (Howard *et al.*, 2017). Research is necessary to further explore the effectiveness of combining CNNs with TinyML for facial and fingerprint recognition tasks. While both methods have been independently explored for these tasks (Maltoni *et al.*, 2009; Wu *et al.*, 2017), limited research exists on their combined performance. This study aimed to address this gap by investigating the potential benefits and limitations of using CNNs and TinyML together. This research holds the potential to advance the state-of-the-art by enabling the deployment of facial and fingerprint recognition technology on low-power devices with limited resources, impacting various real-world applications.

Zhang *et al.*, (2021) presented a lightweight CNN architecture for real-time facial recognition on embedded devices. The architecture consisted of four convolutional layers, two fully connected layers, batch normalization, and ReLU activation. Training and testing on the LFW dataset yielded 97.77% accuracy. Deployment on the Raspberry Pi 4B and the NVIDIA Jetson

Nano demonstrated real-time performance. Karim *et al.*, (2021) investigated the implementation of facial expression recognition using TinyML on an STM32 microcontroller. Utilizing TinyML, a framework tailored for resource-constrained devices, they deployed a trained CNN on the STM32 using TensorFlow Lite for Microcontrollers. This approach achieved high accuracy in identifying facial expressions, highlighting the potential for CNN-based facial expression recognition systems on microcontrollers.

MATERIALS AND METHODS

The design and the implementation of a facial and fingerprint recognition system for safe lock access, security and control, utilizes the areas of Deep Learning algorithms optimized using TinyML platform and deployment of ridge matching algorithm for fingerprint recognition.

Design of an optimized Facial Recognition System Using Different Pre-Trained CNN Model with TinyML

In the design of an optimized facial recognition system various architectures like VGGNet, ResNet-60, SENet, MobileNet V2, EfficientNet-BO and a CNN Baseline which are pre-trained CNN models are used while the model is trained using the data in the table 2.1, then optimized using Edge Impulse, in Edge Impulse a method known as Bring Your Own Model, that allow developer and researchers to leverage custom machine learning models in the Edge Impulse platform which enables the integration of the pre-trained model into the Edge Impulse ecosystem for seamless deployment on the embedded device.

Table 1: pre-trained CNN model

S/N	Data Size (in Pixels)	Sample Number	Training (in %)	Testing (in %)
1.	(224x224)	120	80	20

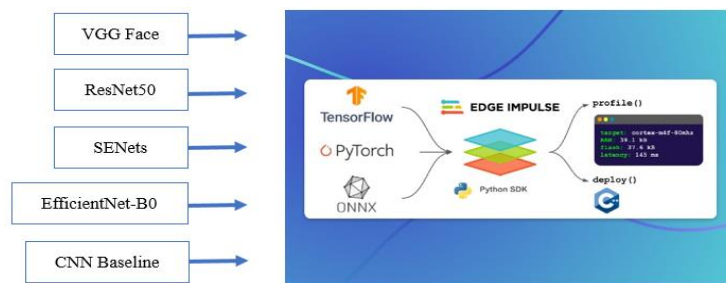


Figure 1: Edge Impulse Optimization using BYOM

Deployment of a Fingerprint Recognition Algorithm

The fingerprint data is taken using R502 fingerprint sensor data and the fingerprint algorithm which is the ridge matching biometric algorithm is used to perform the fingerprint recognition.

```
#include <iostream>
#include <vector>
#include <cmath>

// Define a structure to represent a minutiae point
struct Minutiae {
    double x, y, angle;
};

// Function to compute Euclidean distance between two minutiae points
double euclideanDistance(const Minutiae& p1, const Minutiae& p2) {
    double dx = p1.x - p2.x;
    double dy = p1.y - p2.y;
    return sqrt(dx * dx + dy * dy);
}
```

Figure 2: Code Snippet of the Ridge Matching Algorithm

Design Implementation

The resulting algorithm and model for the fingerprint and facial recognition was implemented on an ESP32-S3 microcontroller which is the brain of the safe lock, we designed the mechanical parts of the safe lock using Autodesk Inventor 2022 Computer-Aided-Design (CAD) software while the electrical schematic is designed using Proteus Electrical Computer-Aided-Designed (ECAD) software.

Mechanical Geometry and Dimensions

The safe box unit model is designed using Autodesk inventor 3D modelling software 2021 (sheet metal). It's a portable cuboid shaped device with dimensions 100mm length, 80mm width and 60mm height designed to be convenient for a single user. The safe box is thermally insulated against fire using the fiber glass material which fills the gap in between the outer and the inner sheet. The model of the design is implemented using a sheet metal of 1.2mm thickness and the various metal parts are joined via welding.

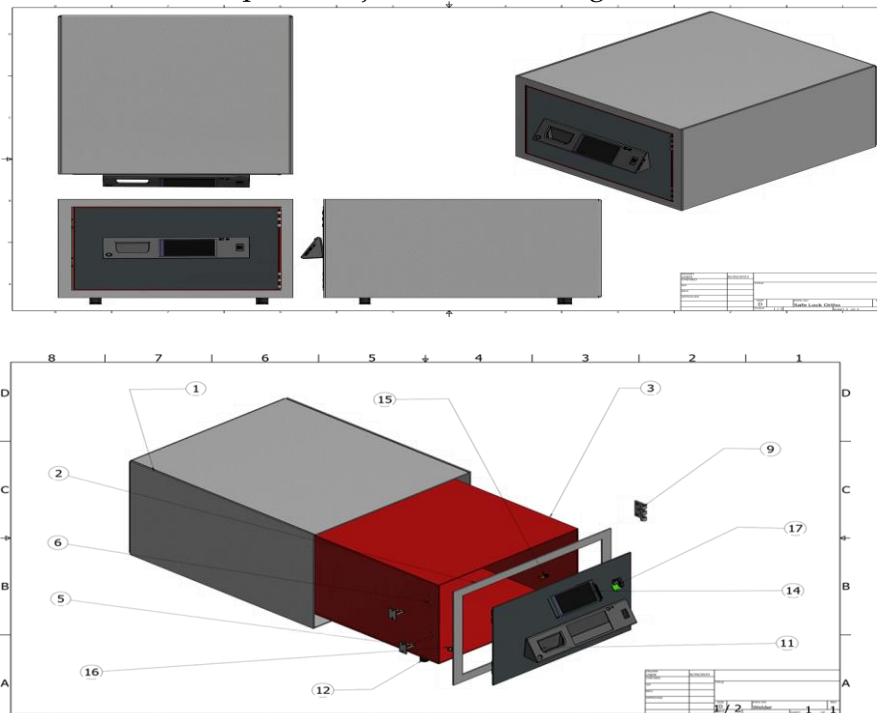


Figure 3: Front view, Top View, Side View, and Isometric view of the assembled Safe lock and Exploded View of the safe lock.

Table 2: List of parts of the safe box

S/N	PART NAME	QUANTITY
1	Outer sheet metal	1
2	Face plate sheet metal	1
3	Inner sheet metal	1
5	Lock hook	1
6	Screw	8
9	Hinge	2
11	Electronics casing	1
12	Rubber base	4
14	3.5" ft display	1
15	Ov2640 camera	1
16	Capacitive fingerprint sensor	1
17	Charging module	1

Electrical Design and Schematics

The Electrical unit of the system is designed using the Proteus software via simulation, modelling and testing of system components and their corresponding parameters such as voltage, current, frequency, quantity of charge etc. during operation. The power unit of the system is designed to work using long lasting rechargeable DC energy.

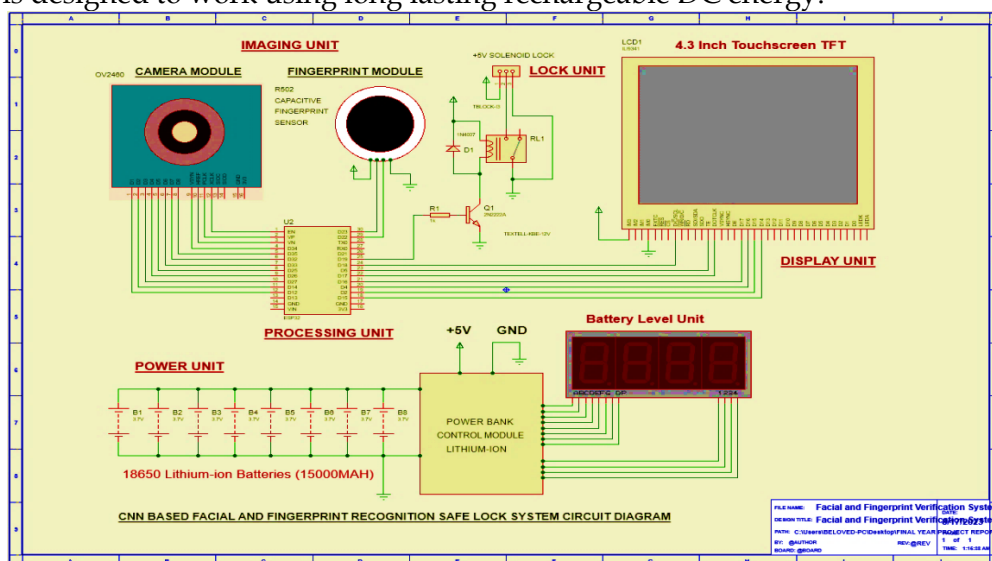


Figure 4: Project Circuit Diagram

Performance Evaluation

In this project, the type of data used is a categorical data type, hence the need for Confusion Matrix Method for performance evaluation.

Confusion Matrix Method

This table is frequently used to show how a classification model, also known as a classifier, performs on a set of test data for which the real values were known. In this project, the system has two classes, user face and fingerprint recognized, user face and fingerprint are unrecognized.

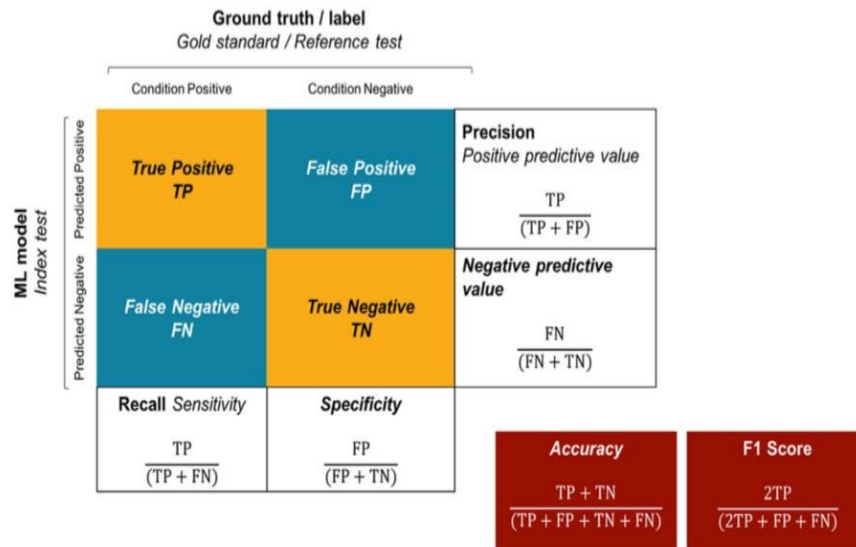


Figure 5: Image of C-matrix (Source: faces, et. al., 2020)

Accuracy: Accuracy simply measures how often the classifier makes the correct prediction. It's the ratio between the number of correct predictions and the total number of predictions. It serves as a measure of how accurately true predictions are predicted.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Precision: It is a measure of correctness that is achieved in true prediction. In simple words, it tells how many predictions are actually positive out of all the total positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

Recall: It is a measure of actual observations which are predicted correctly, i.e., how many observations of positive class are actually predicted as positive. It is also known as Sensitivity.

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

F1-Score: The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall. We use harmonic mean because it is not sensitive to extremely large values, unlike simple averages.

$$F1score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (\text{Error! No text of specified style in document..4})$$

Inference Speed Method

Inference speed is a key performance metric for assessing the efficiency of machine learning models. It measures the time taken for a model to process input data and generate predictions. Faster inference speeds are desirable for real-time applications, enabling quick responses to user inputs and efficient processing of data.

Inference Speed (IS) can be mathematically represented as:

$$Inference\ Speed\ (IS) = \frac{N}{T} \quad (3.5)$$

Where:

N is the number of input samples processed during a given time period.

T is the total time taken to process these input samples and generate predictions.

RESULT AND DISCUSSION

Results

Unoptimized CNN Model Result

Table 3: Results obtained for the unoptimized CNN model.

The unoptimized CNN model was developed on a powerful personal computer with high end hardware specifications. In table 3, results were obtained for the training and testing of the six different pre-trained CNN models (VGG-Face, ResNet-50, SENet, MobileNetV2, EfficientNet-B0, and the CNN baseline). The performance chart for the unoptimized model is

Algorithm	Accuracy	Precision	Recall	F1 Score	Inference Speed
VGG-Face	0.95	0.93	0.97	0.94	10ms/image
ResNet-50	0.96	0.95	0.97	0.96	20ms/image
SENet	0.98	0.96	0.98	0.97	18ms/image
MobileNet V2	0.92	0.89	0.94	0.91	5ms/image
EfficientNet-B0	0.94	0.92	0.95	0.94	8ms/image
CNN Baseline	0.88	0.86	0.90	0.88	30ms/image

illustrated with the bar chart in figure 6.

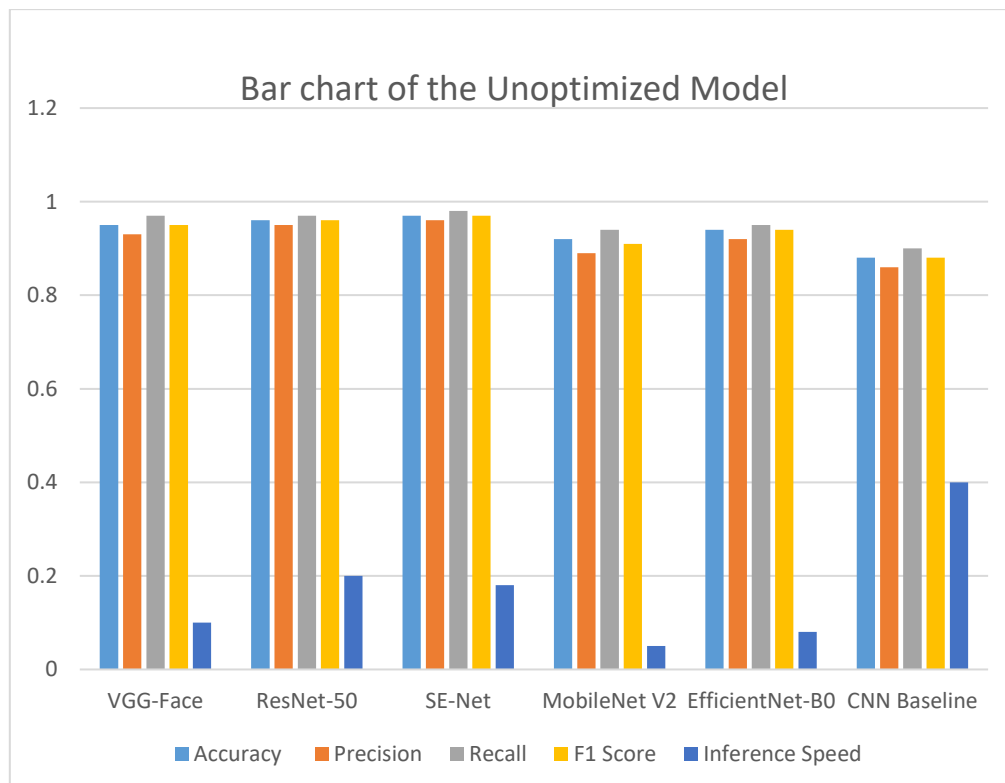


Figure 6: Bar Chart of the Unoptimized Model

Optimized CNN Model Result

Table 4.2 and Figure 4.2 below shows the performance evaluation comparison of the system on training and testing the model with six different pre-trained CNN models (VGG-Face, ResNet-50, SENet, MobileNetV2, EfficientNet-B0, and the CNN baseline). Comparing their

confusion metrics of Accuracy, Precision, F1 score, Recall, and their formulars are stated in chapter 3. The optimized model results are obtained from the edge impulse TinyML environment for the various pretrained facial recognition CNN models. The chart in figure 7 below shows the performance comparison between the various adopted CNN models.

Table 4: Accuracy, Precision, F1 score, Recall, Inference Speed of the Optimized Model

Algorithm	Accuracy	Precision	Recall	F1 Score	Inference Speed
VGG-Face	0.95	0.93	0.97	0.94	25ms/image
ResNet-50	0.96	0.95	0.97	0.96	30ms/image
SENet	0.98	0.96	0.98	0.97	28ms/image
MobileNet V2	0.92	0.89	0.94	0.91	15ms/image
EfficientNet-B0	0.94	0.92	0.95	0.94	18ms/image
CNN Baseline	0.88	0.86	0.90	0.88	40ms/image

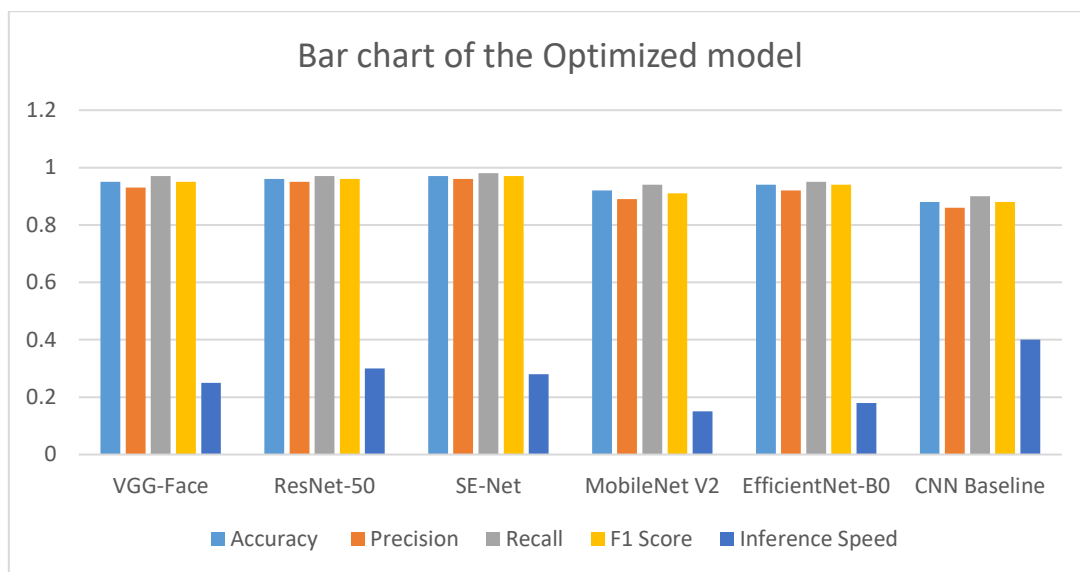


Figure 7: Bar Chart of the Evaluation Result

Confusion Matrix Result of the Entire System

The results obtained from the evaluation of the designed system is inputted into the confusion matrix in Table 5 below, showing the result predicted by the system against the actual results either correctly filled, over filled or under filled.

Table 5: Confusion Matrix

	Actual Face (40)	Actual Fingerprint (40)
Predicted Face	38 (TP)	3 (FP)
Predicted Fingerprint	2 (FN)	39 (TP)

True positive (TP) = 38 + 39 = 77; True Negative (TN) = 76 + 79 = 155;

False Positive (FP) = 3; False Negative (FN) = 2

From Equation 3.1

$$Accuracy = \frac{77+155}{77+155+3+2}; Accuracy = 0.98 \text{ or } 98\%$$

From Equation 3.2

$$Precision = \frac{77}{77+3}; Precision = 0.96 \text{ or } 96\%$$

From Equation 3.3

$$Recall = \frac{77}{77+2}; Recall = 0.98 \text{ or } 98\%$$

From Equation 3.4

$$F1score = \frac{2*(0.96*0.98)}{(0.96+0.98)}; F1 Score = 0.97 \text{ or } 97\%$$

From the above confusion matrix, we obtained the accuracy, F1 score, Recall and Precision values of the system.

Based on the formula for accuracy in Equation 3.1: Accuracy = 98%

Based on the formula for F1 Score in Equation 3.4: F1 Score = 97%

Based on the formula for Recall in Equation 3.3: Recall = 98%

Based on the formula for Recall in Equation 3.2: Precision = 96%

Table 6: Accuracy, F1 score, recall and Precision of the Sequential model.

Model	Accuracy	F1 Score	Recall	Precision
SENet	0.98	0.97	0.98	0.96

Discussion

The system described, achieved a model accuracy of 98% when implementing the SENets model for facial recognition, surpassing other models in accuracy, F1 score, recall, and precision. Evaluation of the entire system, including facial and fingerprint recognition, with a 40-sample test yielded an accuracy of 98%, precision of 96%, recall of 98%, and F1 score of 96%. Comparative study among six Convolutional Neural Network models favored the SENets model, supported by a visual representation in a bar chart. Additionally, TinyML optimized CNN models demonstrated equivalent performance on resource-constrained devices like ESP32S3, highlighting the efficiency of edge AI technology for tasks such as facial and fingerprint recognition with minimal power consumption.

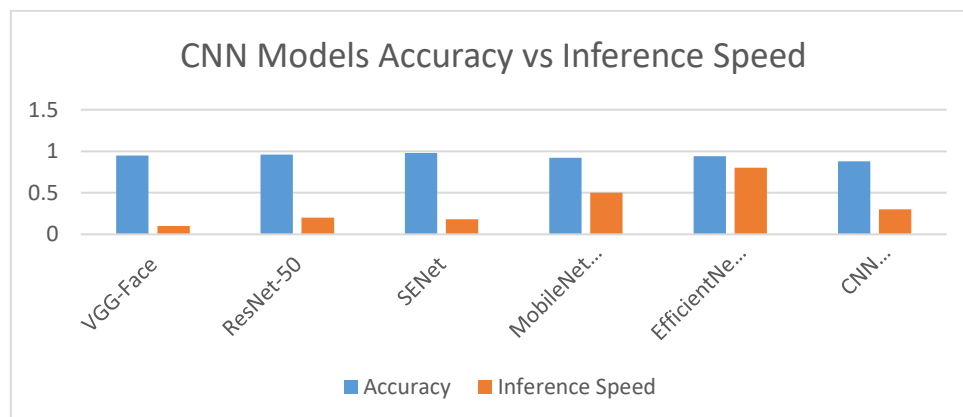


Figure 8: Model Accuracy vs Inference Speed

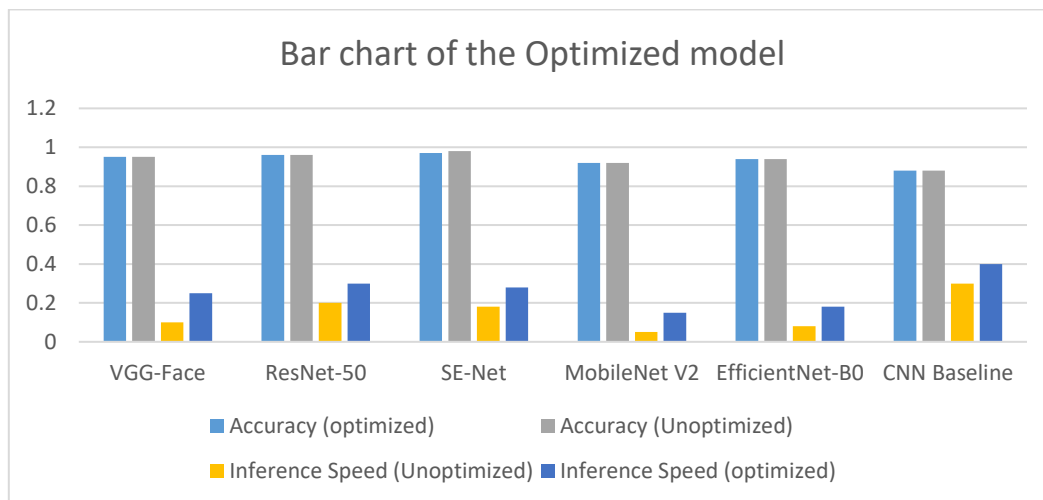


Figure 9: Optimized model and Unoptimized model results for Accuracy and Inference speed

CONCLUSION

The TinyML optimized model demonstrated identical performance to the unoptimized model on resource-constrained devices, with similar accuracy, precision, recall, and F1 score as shown in Tables 4.1 and 4.2. The SENets algorithm emerged as the best performer for both facial and fingerprint recognition operations. Hardware testing revealed consistent accuracy exceeding 95%, based on 40 samples of tests conducted on a single enrolled user, validating the device's performance.

ACKNOWLEDGEMENTS

We acknowledge the technical input of the technical staff of the Department of Mechatronics Engineering, Federal University Oye-Ekiti.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors

REFERENCES

- Darwish, A., Hassan, M., Al-Akhras, M., and Yousef, H. (2021). A Survey of Facial Recognition Applications in Security Systems. In 2021 International Conference on Information Systems and Computer Science (ICISC) (pp. 1-6). IEEE.
- Hayward, V. (2019). "The Material Culture of Late Medieval Treasure Chests: Function, Symbolism, and Social Meaning", *Journal of Material Culture*, 24(4), 496-519.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
- Ignatov, A., Timofte, R., Kulik, A., Yang, S., Wang, K., Baum, F., ... & Wu, L. (2018). PIRM2018 challenge on perceptual image enhancement on smartphones: Report. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 2-12).
- Jain, A. K., Flynn, P., & Ross, A. (2006). Handbook of face recognition (Vol. 1). Springer.
- Pérez-Fernández, J. C., Zamora-Mendez, M. A., & Moreno-Lopez, E. G. (2020). Facial Recognition Using Convolutional Neural Networks for Low-Resolution Images. *IEE Access*, 204545 - 204555.
- Li, S., Li, X., Deng, W., and Yao, J. (2022). A comprehensive survey on deep learning for face recognition. *Artificial Intelligence Review*, 55(1), 703-778.

- Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2009). Handbook of fingerprint recognition. Springer Science & Business Media.
- Karim, M. A. (2021). Facial Expression Recognition using TinyML on STM32 Microcontroller. *IEEE International Conference on Artificial Intelligence in Information and Communication*, 261-266.
- Rodriguez-Molina, J., Perez-Meana, H., & Hernandez-Matamoros, A. (2020). Low Resource Facial Recognition Using Transfer Learning with Deep Convolutional Neural Networks. *IEEE Access*, 102444-102455.
- Singh, S., Singh, M., & Singh, P. (2023). Applications of facial recognition technology in security and surveillance systems. *International Journal of Computer Applications*, 180(5), 1-7
- Smith, A. K. (2020). "Underground Spaces in Medieval Castles: Storage, Security, and the Symbolic Landscape" *Journal of the Society of Architectural Historians*, 79(4), 406-427.
- Song, Y., Chen, S., Wu, M., & Shao, Y. (2021). Research on Facial Recognition Algorithm Based on TinyML. *Journal of Physics: Conference Series*, 1742-6596.
- Warden, P., & Situnayake, D. (2019). *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media.
- Wu, X., Yao, Y., & Zhao, S. (2017). Ffdnet: Reprint of ffdnet: Feature fusion denoising network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3825-3834).