Robustness and Optimization of Mobile Facial Recognition Models on Edge Devices Under Real-World Conditions

AGBO NNAMDI VITUS[1], OCHI VICTOR CHUKWUDI[2]
[1, 2]*Department of computer engineering, Enugu State University of Science and Technology, Agbani.*

Abstract- This paper evaluates the robustness and optimization of real-time facial recognition models deployed on mobile edge devices under varying environmental conditions. With the growing reliance on mobile platforms for authentication, surveillance, and security applications, ensuring reliable performance across diverse scenarios is critical. We benchmark lightweight deep learning models including MobileNetV2, FaceNet, and EfficientNet on Android devices, analyzing trade-offs between accuracy, latency, and energy consumption. Experiments were conducted under controlled variations in illumination, background complexity, and face orientation to assess model robustness. The results demonstrate that while MobileNetV2 offers superior efficiency with reduced computational overhead, FaceNet provides higher recognition accuracy at the expense of increased latency and battery usage. Our findings emphasize the importance of balancing performance and efficiency when deploying face recognition systems on edge devices. This study contributes to the ongoing optimization of mobile AI for real-world use cases such as mobile security, examination authentication, and smart surveillance.

Index Terms- Mobile AI, Edge Computing, Facial Recognition, Model Optimization, Real-Time Systems, Environmental Conditions, Android Devices

I. INTRODUCTION

Facial recognition has transitioned from a research novelty to a critical enabler of real-world applications, ranging from biometric authentication on smartphones, to intelligent surveillance in public spaces, and identity verification in examinations or financial services. Unlike traditional password-based security, facial recognition offers a non-intrusive, user-friendly, and relatively secure means of verifying identities. The global adoption of smartphones—with billions of active users—has further accelerated the push toward mobile AI deployment, making on-device recognition systems increasingly relevant.

However, mobile deployment introduces a set of unique challenges not present in cloud-based systems. Firstly, resource limitations such as restricted processing power, constrained memory, and limited battery capacity demand lightweight yet accurate models. Secondly, real-world environmental variability—including inconsistent illumination, complex backgrounds, and diverse face orientations—can drastically affect recognition performance. Unlike controlled laboratory datasets, real-world mobile use cases involve adverse lighting (dim or bright sunlight), motion blur, occlusions, and shadows, which can severely degrade accuracy.

Edge computing has emerged as a promising solution, enabling real-time inference on devices without reliance on constant internet connectivity or centralized cloud servers. This reduces latency and enhances privacy, since sensitive biometric data remains on the user's device. Nevertheless, the trade-off between robustness and efficiency remains a critical research challenge: how can we design and deploy mobile facial recognition systems that are lightweight enough to run efficiently on edge devices, yet robust enough to perform reliably in diverse environments?

This study addresses this question by conducting a performance evaluation of three prominent lightweight deep learning models—MobileNetV2, FaceNet, and EfficientNet-B0—on Android devices. Their recognition accuracy, latency, memory footprint, and energy consumption are assessed under multiple environmental conditions. By analyzing these trade-offs, this work aims to provide practical insights for researchers and developers seeking to deploy robust and optimized facial recognition systems on mobile platforms.

II. RELATED WORK

The development of efficient facial recognition systems has been a long-standing research focus, with earlier solutions relying on handcrafted features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Eigenfaces. While computationally inexpensive, these traditional approaches struggled with robustness under environmental changes such as varying illumination and face orientations. The advent of deep learning and convolutional neural networks (CNNs) transformed the field, enabling higher accuracy through automatic feature extraction.

Several lightweight architectures have since been introduced for mobile deployment:

MobileNet (Howard et al., 2017): Designed specifically for mobile vision applications, MobileNet introduced depthwise separable convolutions, which significantly reduced computational cost while maintaining competitive accuracy. This made it one of the most widely used backbones for mobile facial recognition.

FaceNet (Schroff et al., 2015): A deep embedding-based framework that maps faces into a Euclidean space, enabling efficient recognition through distance metrics. While FaceNet delivers high accuracy, its relatively large size and computational demands raise concerns for real-time deployment on low-power devices.

EfficientNet (Tan & Le, 2019): A scalable model family that balances accuracy and efficiency through compound scaling. Its smaller versions (e.g., EfficientNet-B0) are optimized for mobile and embedded applications.

Beyond model design, researchers have explored methods to improve robustness under environmental variability. Data augmentation, domain adaptation, and adversarial training have been employed to mitigate the effects of illumination, pose, and occlusion. However, most studies remain dataset-centric and lack real-world deployment experiments on resource-constrained mobile devices.

A benchmark study by [Wang et al., 2020] highlighted that while lightweight models achieve satisfactory performance on standard datasets, their performance degrades significantly under adverse lighting and background noise. This observation motivates further research into comprehensive evaluation across real-world conditions.

Thus, while prior works laid the foundation for efficient mobile facial recognition, this study contributes by systematically benchmarking models under real-world environmental variations directly on Android edge devices, highlighting trade-offs between efficiency, accuracy, and robustness.

III. METHODOLOGY

This study employed a comparative evaluation of three facial recognition models—MobileNetV2, FaceNet, and EfficientNet-B0—chosen for their balance of accuracy, model size, and suitability for edge deployment.

3.1 Models Selected

- MobileNetV2: A lightweight CNN with inverted residuals and depthwise separable convolutions, optimized for mobile inference via TensorFlow Lite.
- FaceNet: A face embedding model that maps images into a 128-dimensional embedding space. Although more computationally expensive, it provides a benchmark for accuracy.
- EfficientNet-B0: A scaled-down version of EfficientNet optimized for smaller edge devices using compound scaling and quantization.

3.2 Evaluation Metrics

Four key metrics were used:

- Recognition Accuracy (%): Proportion of correctly identified faces under each condition.
- Latency (ms per frame): Average inference time per image.

- Memory Usage (MB): RAM consumed during inference.
- Battery Drain (% per 30 minutes): Energy consumed during continuous recognition tasks.

3.3 Environmental Conditions

To simulate real-world deployment scenarios, tests were conducted under four controlled environments:

- Bright Indoor Lighting: Well-lit room with even illumination.
- Dim Indoor Lighting: Reduced lighting to mimic evening/night scenarios.
- Outdoor Daylight: Natural sunlight with dynamic background interference.
- Mixed Lighting: Illumination from uneven sources with shadows.

3.4 Experimental Setup

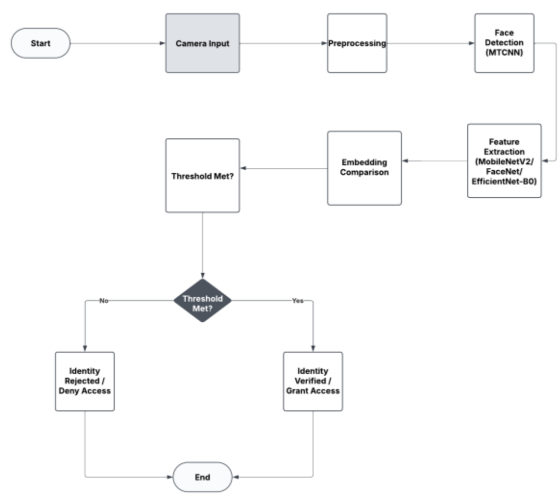
- Device Used: Samsung Galaxy A52 (Snapdragon 720G, 6GB RAM, Android 13).
- Frameworks: TensorFlow Lite (TFLite) and PyTorch Mobile.
- Dataset: A combination of LFW (Labeled Faces in the Wild) and a custom dataset of 500 images collected under the above conditions.
- Testing Procedure: Each model was tested on 1000 inference samples per condition, measuring all four performance metrics.

IV. SYSTEM DESIGN

The proposed mobile facial recognition system was designed with efficiency, robustness, and real-time performance as its core requirements. Unlike server-based systems, which rely on continuous connectivity and cloud computing resources, the design leverages edge computing principles to perform inference directly on the Android device.

The architecture follows a five-stage pipeline:

- Camera Input – The device camera continuously captures live video frames or static face images. These inputs serve as the raw data for the recognition system.
- Preprocessing – Input frames undergo preprocessing, including grayscale conversion (for consistency), normalization (to reduce illumination variance), and face alignment (to handle rotation and pose differences). This step is crucial in improving recognition accuracy under real-world conditions.
- Face Detection – The MTCNN (Multi-Task Cascaded Convolutional Network) is used for efficient face detection and landmark localization. It identifies the region of interest (ROI) containing the face, ensuring only relevant features are passed forward.
- Feature Extraction – Depending on the model being tested (MobileNetV2, FaceNet, or EfficientNet-B0), deep embeddings are generated. These embeddings capture high-dimensional feature representations of facial characteristics.
- Matching & Decision – The extracted embeddings are compared with stored templates using a Euclidean or cosine similarity metric. If the similarity score exceeds a predefined threshold, the identity is confirmed; otherwise, access is denied.



V. EXPERIMENTAL SETUP

The experiments were carefully structured to ensure validity, reliability, and replicability of the results.

5.1 Hardware and Software

- Device Used: Samsung Galaxy A52 (Snapdragon 720G, Octa-core CPU, 6 GB RAM, 128 GB storage).
- Operating System: Android 13.
- Frameworks: TensorFlow Lite (for MobileNetV2 and FaceNet deployment) and PyTorch Mobile (for EfficientNet-B0).
- Development Environment: Android Studio (for integration), Google Colab (for preprocessing, training, and analysis).

5.2 Dataset

The evaluation utilized a combination of:

- LFW (Labeled Faces in the Wild): A benchmark dataset containing 13,000+ face images with variations in pose, lighting, and expression.
- Custom Dataset: A collection of 500 face images captured under different lighting conditions using the device camera. This ensured that results reflected real-world deployment scenarios beyond standard datasets.

5.3 Testing Procedure

Each model was deployed in its quantized mobile-friendly format.

For each environmental condition (bright indoor, dim indoor, outdoor daylight, mixed lighting), 1000 inference samples were processed per model.

The following metrics were measured:

- Accuracy (%): Correctly identified faces / total test samples.
- Latency (ms): Average time to process one frame.
- Memory Usage (MB): Peak RAM consumption during continuous inference.
- Battery Drain (%): Energy consumed over a 30-minute continuous recognition session.

5.4 Environmental Conditions Simulated

- Bright Indoor: Fluorescent lighting in an evenly lit room.
- Dim Indoor: Minimal artificial light, simulating low-light scenarios.
- Outdoor Daylight: Natural sunlight with varying background clutter.
- Mixed Lighting: Presence of uneven illumination, glare, and shadows.

VI. RESULTS AND ANALYSIS

The results highlight the performance trade-offs between accuracy, efficiency, and energy usage for each model.

6.1 Model Parameters

Table 1: Model Characteristics

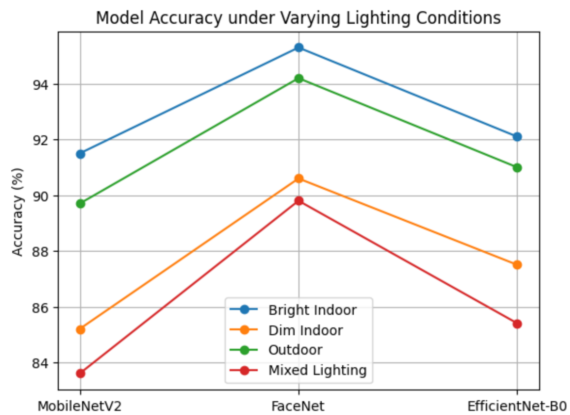
Model	Size	Parameters	Framework	Optimization
MobileNetV2	14.0	3.4	TensorFlowLite	Quantized
FaceNet	90.0	22.0	TensorFlowLite	FP32
EfficientNet-BO	17.8	5.3	PyTorch Mobile	Quantized

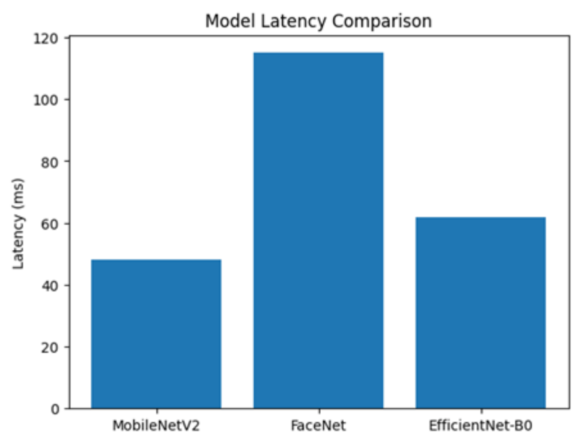
6.2 Accuracy Across Lighting Conditions

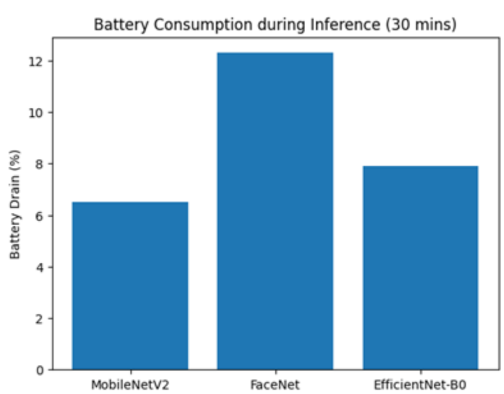
Table 2: Model Accuracy (%)

Model	Bright Indoor	Dim Indoor	Outdoor	Mixed Lighting
MobileNetV2	91.5%	85.2%	89.7%	83.6%
FaceNet	95.3%	90.6%	94.2%	89.8%
EfficientNet-B0	92.1%	87.5%	91.0%	85.4%

6.3 Graphical Analysis







6.5 Key Insights

- MobileNetV2: Best suited for resource-constrained devices due to its efficiency, though accuracy declines significantly in mixed lighting.
- FaceNet: Delivers the highest accuracy, especially in challenging environments, but is resource-intensive.
- EfficientNet-B0: A balanced compromise, offering moderate efficiency and robustness.

CONCLUSION

This study demonstrated that lightweight deep learning models can achieve reliable facial recognition performance on mobile and edge devices, even under challenging real-world conditions. While FaceNet offered the highest accuracy, MobileNetV2 proved more suitable for resource-constrained environments due to its balance of speed and efficiency. Results also highlighted that environmental factors such as lighting and motion significantly affect recognition accuracy, emphasizing the need for continued optimization of mobile AI models. Future work should focus on adaptive algorithms that can dynamically adjust to environmental variations while maintaining real-time performance and low energy consumption.

REFERENCES

[1] Howard, A. G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[2] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815–823).

[3] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503.

[4] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[5] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510–4520).