Real-Time Deepfake Detection on Embedded Systems: A Portable Device Architecture

Alwin[1], Pavitra Premanand[2], Vijay S P[3], Kanishka S[4], Anjana R[5], N Vishnu Venkatesh[6]
[1,2,3,4,5,6]*Department of Forensic Science JAIN (Deemed-to-be University), Bengaluru, Karnataka*

***Abstract*—The rapid proliferation of hyper-realistic deepfake technology poses severe threats to digital security, public trust, and individual privacy. While high-performance server-side detection models exist, there is a critical gap in localized, real-time verification tools for field investigators. This paper proposes a portable deepfake detection framework optimized for embedded mobile hardware. Utilizing a MobileNetV2 architecture fine-tuned via a dual-phase transfer learning regimen on the Deepfake Detection Challenge (DFDC), Celeb-DF, and FaceForensics++ datasets, the system focuses on identifying high-frequency boundary artifacts and spatial inconsistencies. To ensure efficiency on resource-constrained devices, the model was converted to an INT8-quantized TensorFlow Lite (TFLite) format, reducing the model size from 9MB to 2.3MB and peak RAM usage to 12MB. The resulting system, deployed on a Samsung Galaxy A31, achieves an ROC AUC of 0.8718 with an inference latency of 100–200ms per frame. The framework consists of a native Android application for immediate field verification and a Python-based interface for detailed forensic analysis, effectively bridging the gap between theoretical deep learning and practical digital forensic requirements.**

***Index Terms*—Deepfake Detection, MobileNetV2, Embedded Systems, INT8 Quantization, Digital Forensics, TensorFlow Lite.**

I. INTRODUCTION

The evolution of synthetic media has entered an era of "hyper-realism." The emergence of Generative Adversarial Networks (GANs) and more recent Diffusion Models has enabled the creation of manipulated audio and video content that is virtually indistinguishable from authentic media to the human eye. These "deepfakes" are no longer mere academic curiosities; they are active tools for misinformation, corporate espionage, and sophisticated social engineering attacks(Venkatesh et al., 2023).

Traditional detection methods often rely on manual physiological cues—such as unnatural blinking or skin-tone inconsistencies. However, as generative models evolve, they are becoming adept at synthesizing these very features, rendering manual inspection obsolete. Consequently, the development of automated, high-accuracy detection systems has become a priority for the digital forensics community. A significant limitation of current state-of-the-art (SOTA) detectors is their reliance on heavy computational clusters. Most forensic analysis is currently performed on high-performance servers, which introduces two primary risks:

1. Latency:
The time required to upload large video files and wait for server-side processing prevents immediate verification.

2. Privacy:
Transmitting sensitive evidence to a cloud server introduces potential data breaches and chain-of-custody concerns.
There is, therefore, a pressing need for "edge-compatible" solutions—forensic tools that reside entirely on the investigator's device. This research focuses on the development of a portable device architecture capable of executing deepfake detection locally.

1.1. The Mechanics of Deepfakes

To detect a deepfake, one must understand its synthesis. Most deepfakes utilize a GAN structure consisting of two competing networks: the Generator (G) and the Discriminator (D). The Generator attempts to create a facial manifold that matches the target distribution, while the Discriminator attempts to distinguish the synthetic image from a real one. As this

zero-sum game progresses, the Generator becomes expert at creating photorealistic textures. However, this process often leaves microscopic "artifacts"—spatial inconsistencies in the frequency domain, boundary-layer noise, or specular reflection errors in the pupils. Our research targets these high-frequency discrepancies(Kamalesh & Venkatesh, 2025).

1.2. Dataset Diversity

A detector is only as robust as its training data. To avoid overfitting to a single generation tool, we utilized a diversified pipeline:

- Deepfake Detection Challenge (DFDC): A massive dataset of over 100,000 videos provided by Meta, focusing on wide-scale diversity.
- Celeb-DF: A "second generation" dataset characterized by higher visual quality and more subtle manipulations.
- FaceForensics++: A comprehensive benchmark developed by the Technical University of Munich, focusing on specific manipulation methods like Face2Face and NeuralTextures.

By training on this heterogeneous corpus, the proposed model is engineered to generalize across various generation tools and environmental conditions.

II. REVIEW OF LITERATURE

The trajectory of facial forensics has shifted from the identification of biological "glitches" to the deployment of lightweight neural architectures.

2.1. From Manual Cues to Deep Learning

Early research focused on physiological inconsistencies. Li et al. (2020), in *"In Ictu Oculi,"* demonstrated that early deepfakes failed to replicate natural eye-blinking patterns because the training sets primarily contained images of people with open eyes. While groundbreaking, this method proved fragile as GANs began incorporating temporal consistency. This transition underscored the necessity of moving toward Convolutional Neural Networks (CNNs), which can detect non-linear spatial artifacts that human observers cannot perceive.(Shenoy et al., 2025)

2.2. The Role of Efficient Architectures

The use of XceptionNet by Rossler et al. (2019) proved that deep learning could achieve near-perfect accuracy in controlled environments. However, XceptionNet's massive parameter count made it unsuitable for mobile deployment. This led to the adoption of depth-wise separable convolutions, as proposed by Chollet (2017). By decoupling spatial and channel-wise correlations, it became possible to maintain high accuracy while drastically reducing floating-point operations (FLOPs). This theoretical foundation justifies our selection of MobileNetV2 as the backbone for our portable system.

2.3. Edge-Compatible Forensics

The shift toward "edge AI" was furthered by Afchar et al. (2018, 2025) with the development of MesoNet. MesoNet proved that a compact network focusing on "mesoscopic" properties (intermediate-level features) could identify forgeries without the need for extreme depth. This research informs our objective: reducing the computational footprint to allow a Samsung Galaxy A31 to perform tasks previously reserved for GPUs.

2.4. Hybrid and Temporal Approaches

Recent studies by Guera and Delp (2019) suggest that combining CNNs with Long Short-Term Memory (LSTM) units can detect "frame-to-frame" flickering. While our current implementation focuses on static facial crops for maximum speed, the work of Alzurfi and Altaei (2025) suggests that hybridizing CNN features with statistical classifiers (like SVM or Logistic Regression) can sharpen the decision boundary between "Real" and "Fake."

III. METHODOLOGY

The proposed system is built on a four-phase framework: Forensic Data Engineering, Neural Architecture Design, Multi-stage Optimization, and Hardware-Aware Quantization.

3.1. Forensic Data Engineering

The quality of the input determines the reliability of the verdict. We implemented a rigorous preprocessing pipeline:

1. Stochastic Temporal Sampling:

To prevent overfitting to redundant frames, we sampled $N=5$ frames per video using the formula: $f_i = \lfloor i \cdot \frac{T}{N+1} \rfloor, \forall i \in \{1,2, \dots, N\}$ This ensures the model

sees the face from different angles and lighting conditions.(Shukla et al., 2023)

2. Boundary-Preserving Localization:
Using a Haar-cascade classifier, we localized faces but applied a 20% spatial margin. The bounding box $B = [x, y, w, h] $ was expanded to: $C = [x - 0.1w, y - 0.1h, 1.2w, 1.2h]$ This is critical because deepfake artifacts often occur at the "blend zone" where the synthetic mask meets the original skin.

3. Standardization:
Images were resized to 224×224 pixels via bicubic interpolation and normalized to a range of $[-1, 1] $ to stabilize backpropagation.

3.2. Neural Architecture Design
We selected MobileNetV2 due to its efficiency. The architecture utilizes:

- Inverted Residuals:

Shortcut connections between thin bottleneck layers maintain information flow while reducing parameters.

- Depth-wise Separable Convolutions:

This splits a standard convolution into a depth-wise and a pointwise step, drastically lowering the computational cost for the MediaTek Helio P65 processor.

3.3. Multi-stage Optimization (Transfer Learning)
A two-phase training process was employed:

- Phase I (Linear Probe):

The MobileNetV2 backbone was frozen. A custom head consisting of a Global Average Pooling (GAP) layer and a Dropout layer ($p=0.2$) was added. This was optimized for 12 epochs ($\eta = 10^ {-3} $).

- Phase II (Fine-Grained Adjustment):

The top 50 layers were unfrozen, and the learning rate was dropped to $\eta = 10^ {-5} $ for 15 additional epochs. This allowed the model to specialize in detecting "micro-artifacts" like specular reflection inconsistencies in pupils and frequency aliasing in skin pores.

3.4. Hardware-Aware Quantization
To move from Python to Android, we used Post-Training Quantization (PTQ). We mapped 32-bit floating-point weights (r) to 8-bit integers (q): $q = \text{round}\left(\frac{r}{s} + Z\right)$ This reduced the model footprint from 9MB to 2.3MB, optimizing cache utilization and reducing memory bandwidth bottlenecks.

IV. SYSTEM IMPLEMENTATION AND DEPLOYMENT

The transition from a laboratory model to a field tool required an architectural bridge that respects the limitations of mobile hardware.

4.1. Hardware Selection and Integration
The Samsung Galaxy A31 (MediaTek Helio P65) was selected to demonstrate that forensic AI does not require flagship hardware. We utilized the Android Neural Network API (NNAPI), allowing the TFLite interpreter to communicate directly with the octa-core CPU for accelerated inference.

4.2. Engineering the Android Application
The app, developed in Kotlin, implements several memory-saving strategies:

1. MappedByteBuffers: To avoid the "Low Memory Killer" (LMK) from shutting down the app, we used memory-mapped files. Instead of loading the entire 2.3MB model into active RAM, the app reads the model directly from the disk as needed.
2. Asynchronous Inference Pipeline: To keep the UI responsive, the following sequence is handled on a background thread:

- *Image Selection* $\rightarrow$ *Bitmap Decoding* $\rightarrow$ *Channel Normalization* $\rightarrow$ *TFLite Interpreter Execution* $\rightarrow$ *Verdict Delivery*.

4.3. Memory and Power Efficiency
Quantization reduced peak memory usage from 45MB to 12MB. This is vital for field investigators who may be running other evidence-collection apps simultaneously. The system operates entirely offline, ensuring that no data leaves the device.

V. RESULTS AND ANALYSIS

5.1. Evaluation Framework
The model was tested on a held-out validation set from the DFDC dataset (1,003 samples: 917 Real, 86 Fake).

Given the class imbalance, raw accuracy was deemed insufficient; therefore, we utilized ROC AUC, F1-Score, and the Matthews Correlation Coefficient (MCC).

5.2. The "Label Inversion" Incident

During Evaluation 1, the model showed an accuracy of 83.95% and an ROC AUC of 0.8718. However, the confusion matrix revealed that the model was tagging almost everything as "Fake."

Investigation revealed a Label Inversion: the mapping between class indices (0 and 1) had flipped during preprocessing. High scores were being interpreted as "Real" instead of "Fake." This was mathematically confirmed in Evaluation 2: after correcting the labels, the ROC AUC shifted from 0.8718 to 0.1282 ($1 - 0.8718 = 0.1282$). This "mirror-image" relationship proved that the model's discriminative power was intact, but the output interpretation was reversed.

5.3. Quantitative Performance (Corrected)

The following table summarizes the final performance metrics:

Metric	Keras Float32	TFLite INT8
ROC AUC	0.8718	0.8686
Accuracy	83.95%	83.05%
F1 Score	0.9068	0.9053
Model Size	9 MB	2.3 MB
Peak RAM	45 MB	12 MB

The negligible gap between the Float32 and INT8 models confirms that quantization did not significantly degrade the classification integrity.

5.4. Threshold Sensitivity and Calibration

A threshold sweep (0.01 to 0.99) revealed that the optimal operating point depends on the use case:

- Low Threshold (0.15): Maximizes Fake Recall (catches almost all deepfakes but increases False Positives).
- High Threshold: Maximizes Real Recall (ensures genuine videos aren't flagged, but may miss subtle fakes).

Calibration curves showed that the model is slightly overconfident—assigning extreme scores (0.0 or 1.0) more often than the empirical accuracy justifies. Applying temperature scaling would be a recommended future improvement.

5.5. Field Testing

Three real-world samples were processed on the Galaxy A31. The Haar-cascade detector successfully localized faces, and the TFLite interpreter provided verdicts within 100–200ms per frame. A limitation was noted: faces occupying less than 5% of the frame area were occasionally missed by the detector.

VI. FINDINGS AND DISCUSSION

6.1. Achievements and Practicality

The project successfully demonstrated that a capable deepfake detector can be deployed on mid-range hardware. An ROC AUC of 0.87 on the DFDC dataset is competitive with specialized lightweight models like MesoNet, while offering a complete end-to-end mobile deployment stack.

6.2. The Lesson of Metric Auditing

The label inversion incident serves as a critical case study in AI forensics. It highlights that high accuracy can be a "false signal" if the evaluation pipeline is not audited. The project now mandates a manual "sanity check" (running known fakes through the model) before any automated evaluation.

6.3. The Threshold Trade-off

The research confirms that there is no "universal" threshold. In a journalistic triage scenario, a low threshold is preferable to avoid missing a fake. In a legal/courtroom scenario, a high threshold is required to avoid falsely accusing someone of forgery.

6.4. Generalization Limits

While the model was trained on three datasets, evaluation was limited to DFDC. Deepfake detectors often struggle with "cross-dataset generalization." For example, a model trained on GAN-based swaps may fail against Diffusion-based generation. This remains a primary area for future research.

VII. CONCLUSION

This research has bridged the gap between high-compute deep learning and portable digital forensics. By optimizing a MobileNetV2 architecture through INT8 quantization and a dual-phase transfer learning regimen, we developed a system that provides real-time deepfake detection on the Samsung Galaxy A31.

The system achieved a 74% reduction in model size and a significant decrease in RAM consumption without compromising the discriminative power (ROC AUC 0.87). The identification and correction of label inversion underscored the necessity of rigorous auditing in forensic AI pipelines. This portable architecture empowers field investigators to perform immediate, privacy-preserving verification of synthetic media, providing a scalable defense against the growing threat of digital misinformation.

REFERENCES

[1] X. Chu, Y. Chen, M. C. Stamm, and K. J. R. Liu, "Information theoretical limit of media forensics: The forensicability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 774–788, 2015.

[2] Z. Huang, J. Hu, X. Li, Y. He, X. Zhao, B. Peng, B. Wu, X. Huang, and G. Cheng, "SIDA: Social media image deepfake detection, localization and explanation with large multimodal model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 28831–28841, 2025.

[3] X. Fu, Z. Yan, T. Yao, S. Chen, and X. Li, "Exploring unbiased deepfake detection via token-level shuffling and mixing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, pp. 3040–3048, 2025.

[4] C. Tan, R. Tao, H. Liu, G. Gu, B. Wu, Y. Zhao, and Y. Wei, "C2P-CLIP: Injecting category common prompt in CLIP to enhance generalization in deepfake detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, pp. 7184–7192, 2025.

[5] C. Wang, Z. He, X. Hu, W. Guan, W. Wang, and Z. Fu, "MAP-Mamba: Multi-artifacts perception Mamba for generalizable face forgery detection," *IEEE Trans. Inf. Forensics Security*, 2026.

[6] Z. Yang, J. Liang, Y. Xu, X.-Y. Zhang, and R. He, "Masked relation learning for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1696–1708, 2023.

[7] Z. Wang, Y. Chen, Y. Yao, M. Han, W. Xing, and M. Li, "IDCNet: Image decomposition and cross-view distillation for generalizable deepfake detection," *IEEE Trans. Inf. Forensics Security*, 2025.

[8] R. Xia, D. Liu, J. Li, L. Yuan, N. Wang, and X. Gao, "MMNet: Multi-collaboration and multi-supervision network for sequential deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3409–3422, 2024.

[9] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang, "ISTVT: Interpretable spatial-temporal video transformer for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1335–1348, 2023.

[10] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 6105–6114, 2019.

[11] K. Vidya, P. Ramesh, H. Viknesh, and S. Devanand, "Compressed deepfake detection using spatiotemporal approach with model pruning," *Procedia Comput. Sci.*, vol. 230, pp. 436–444, 2023.

[12] M. N. V. Venkatesh, D. A. Rajiv, M. P. Das, and M. S. Warrier, "Vantage point recreation: A novel approach in endpoint security for smart homes," *Int. J. Innov. Res. Technol. (IJIRT)*, 2026. DOI: 10.64643/IJIRTV12I8-191180-459.

[13] M. Shukla, V. Srivastav, M. D. Khare, and N. V. Venkatesh, "IoT-driven solutions for VANET trustworthiness: Examining misconduct and position security challenges," *Multidisciplinary Reviews*, vol. 6, Art. no. 2023ss059, 2024. DOI: 10.31893/multirev.2023ss059.

[14] S. S. Shenoy and N. V. Venkatesh, "A predictive framework for real-time courtroom assistance using AI-based mock legal advisor," *Int. J. Res. Anal. Rev. (IJRAR)*, vol. 12, no. 2, pp. 440–444, May 2025. IJRAR Paper

[15] V. V. Natarajan, P. Singhal, D. Pandey, M. Sharma, R. Rautdesai, D. Khubalkar, and A. Gupta, "Crime forecasting using historical crime location using CNN-based images classification mechanism," 2023. DOI: 10.4018/978-1-6684-8618-4.ch013.

[16] V. V. Natarajan, P. Das, and A. Rajiv, "A robust detect and avoid system for autonomous drone navigation," *NexusTech*, vol. 1, Art. no. 2026004, 2026. DOI: 10.31893/tech.2026004.