

AI-Based Deepfake Detection System for Fake Video and Image Identification

**Ms.Saranya G,
AP/AI&DS**
United Institute of
Technology

**Ms.Dhanyasree
M,**
Dept of AI & DS
United Institute of
Technology

Kirubhasahar R
Dept. of AI & DS
United Institute of
Technology

**Mokith Kumar
V R**
Dept. of AI & DS
United Institute of
Technology

MuthuKumar S
Dept. of AI & DS
United Institute of
Technology

**Sarvesh Kumar
S**
Dept. of AI & DS
United Institute of
Technology

Abstract — The rapid advancement of Generative Adversarial Networks (GANs), diffusion models, and deep learning-based image synthesis techniques has made it increasingly difficult to distinguish AI-generated content from authentic photographs and videos. Deepfake videos, synthetic portraits, AI-generated artwork, and digitally manipulated media now pose serious threats to information integrity, digital trust, and public safety. This paper presents an AI-based Deepfake Detection System designed to operate in real time directly on mobile devices without any dependency on external servers or cloud infrastructure. The proposed system leverages a fine-tuned EfficientNet-B4 convolutional neural network trained on a broad, multi-source dataset encompassing real photographs, GAN-generated faces, Stable Diffusion artwork, Midjourney synthetic images, and CIFAKE objects. The trained model is exported in ONNX format and deployed on Android using ONNX Runtime, performing continuous on-device inference at 20 frames per second through a screen capture service. A multi-stage output classification mechanism employing rolling temporal averaging, standard deviation-based variance penalty, and hysteresis control categorises every captured frame into one of three zones: Real, Uncertain, or AI-Detected, displayed through a real-time floating overlay badge. Experimental evaluation demonstrates that the system achieves a validation accuracy of 93.4% with high precision in detecting diverse AI-generated content categories across faces, objects, environments, and digital artworks. **Key Words:** Deepfake Detection, EfficientNet-B4, ONNX Runtime, On-Device Inference, GAN, Diffusion Models, Android, Real-Time Detection, AI

1. INTRODUCTION

The proliferation of deep learning technologies, particularly Generative Adversarial Networks (GANs) and latent diffusion models, has enabled the creation of synthetic visual content with a degree of photorealism that is indistinguishable to the human eye. Deepfake technology, which originally referred primarily to face-swapping in video content, has since evolved into a broad category of AI-generated media that encompasses full-body synthesis, landscape generation, photorealistic artwork, and digitally manipulated imagery across diverse content categories.

The societal and security implications of undetected deepfake content are profound. Fabricated videos of public figures have been used to spread political misinformation. Synthetic portrait photographs have enabled identity fraud in digital onboarding systems. AI-generated artwork presented as authentic photography has introduced questions of intellectual property and authenticity in creative industries. Law enforcement agencies and cybersecurity organisations worldwide have identified deepfake proliferation as a Category-1 digital trust threat.

Existing deepfake detection solutions predominantly rely on server-side inference pipelines, requiring the captured image or video frame to be transmitted to a remote backend for analysis. This architecture introduces latency, privacy vulnerabilities, and internet dependency — making it unsuitable for real-time, on-device consumer applications. Furthermore,

most prior systems are specialised for face-swap detection and fail to generalise across the broader spectrum of AI-generated content including objects, environments, and digital art.

This paper presents a complete AI-based Deepfake Detection System that addresses all three limitations. The proposed system performs all inference on-device at 20 frames per second using an EfficientNet-B4 model exported in ONNX format and executed through ONNX Runtime on Android. A multi-stage temporal smoothing pipeline with variance-aware adaptive thresholding and a three-zone classification output (Real, Uncertain, AI-Detected) provides robust, low-latency detection across diverse content categories without any network requirement.

The key contributions of this work are:

- A fully on-device, real-time deepfake detection system deployed on Android achieving 20 fps inference using ONNX Runtime with EfficientNet-B4.
- A broad multi-source training dataset combining seven distinct real and synthetic image sources covering faces, objects, environments, cartoons, and AI artwork.
- A three-zone output classification system (Real / Uncertain / AI-Detected) driven by rolling temporal averaging, standard deviation variance penalty, and hysteresis-based flip suppression.
- A multi-model ensemble architecture supporting concurrent inference from multiple specialist ONNX models with configurable weighted averaging, maximum, or majority-vote fusion.
- A non-intrusive floating overlay badge that displays live confidence percentage and animated colour-coded status without interrupting user interaction.

2. LITERATURE SURVEY

A systematic review of prior work in deepfake detection, on-device machine learning inference, and

AI-generated content identification has been conducted. The following studies represent the most directly relevant contributions, and their collective limitations motivate the design of the proposed system.

A. FaceForensics++: Learning to Detect Manipulated Facial Images

Authors: Rossler et al. | IEEE ICCV, 2019. This foundational benchmark introduced a large-scale dataset of facial manipulation methods and evaluated CNN-based binary classifiers. The study established that XceptionNet architectures achieve strong detection performance on compression-matched test sets. However, the work is strictly limited to facial manipulation detection and does not generalise to non-facial AI-generated content such as landscapes, objects, or digital artwork.

B. Detecting Deep-Fake Videos from Appearance and Behavior

Authors: Agarwal et al. | IEEE CVPR Workshop, 2020. This paper proposed a dual-stream detection model analysing both facial appearance and behavioural signals such as eye blinking and facial movement patterns. While behavioural cues improve robustness, the approach is computationally expensive and requires video sequences of sufficient length, making it unsuitable for single-frame or real-time still-image detection scenarios.

C. MesoNet: A Compact Facial Video Forgery Detection Network

Authors: Afchar et al. | IEEE WIFS, 2018. MesoNet presented a lightweight convolutional architecture optimised for embedded deployment with lower parameter count than XceptionNet-based detectors. The study demonstrated that compact networks can achieve competitive accuracy on GAN-generated faces. However, the model was trained exclusively on face-swap data and exhibits significant accuracy degradation on non-face synthetic content categories.

S. No	Author s / Year	Methodolog y	Strengths	Limitation s
1	Rossler et al. (ICCV 2019)	XceptionNet on FaceForensics++ benchmark.	Strong on face-swap content.	No non-face AI content; server-only.
2	Agarwal et al. (CVPR 2020)	Dual-stream appearance + behaviour model.	Robust to compression.	Video-only; too slow for mobile.
3	Afchar et al. (WIFS 2018)	Compact MesoNet CNN for embedded use.	Lightweight architecture.	Face-swap only; low generalisation.
4	Li et al. (CVPR 2020)	Multi-task forgery detection + segmentation.	Interpretable output.	Too heavy for real-time mobile.
5	Ojha et al. (CVPR 2023)	Linear probe on CLIP for diffusion detection.	Good domain generalisation.	Not evaluated on mobile; slow.

Table -1: Literature Survey Comparison

D. On the Detection of Digital Face Manipulation

Authors: Li et al. | IEEE CVPR, 2020. This work introduced a multi-task learning framework for simultaneous face forgery detection and segmentation. While segmentation improves interpretability, the added computational overhead precludes real-time mobile deployment. No evaluation on non-facial synthetic content or on-device inference is reported.

E. Towards Universal Fake Image Detection Exploiting Diffusion Models

Authors: Ojha et al. | IEEE CVPR, 2023. This recent study addressed the emerging challenge of detecting diffusion model-generated images, which are increasingly difficult to distinguish from photographic content. The approach employs a linear probe on CLIP features. While the method shows promise for out-of-domain generalisation, it has not

been evaluated for mobile deployment and incurs significant inference cost unsuitable for 20 fps on-

3. EXISTING SYSTEM AND LIMITATIONS

The most directly comparable existing system is the MesoNet family of compact deepfake detectors (Afchar et al., 2018), which specifically targeted embedded deployment with reduced model size. While MesoNet demonstrated that lightweight convolutional architectures can achieve competitive accuracy on GAN-generated face content, a detailed analysis reveals critical limitations that render it inadequate for the proposed use case:

- **Content Scope Limitation:** MesoNet and all related facial forgery detectors are trained exclusively on face-swap and face-reenactment datasets. They exhibit near-random performance on non-facial AI-generated content such as Stable Diffusion landscapes, Midjourney artwork, CIFAKE object synthesis, and AI-generated cartoons.
- **Binary Output Only:** Existing systems output a single binary Real/Fake decision with no intermediate uncertainty zone. This causes frequent misclassification of edited real videos and compressed content near the decision boundary.
- **No Temporal Smoothing:** All reviewed systems perform per-frame classification with no rolling average or temporal consistency mechanism, producing high-frequency flickering on streaming video content.
- **Server Dependency:** The majority of deployed deepfake detection tools require transmission of captured frames to a remote server for inference, introducing latency, privacy risks, and internet dependency.
- **No Multi-Model Ensemble:** No existing mobile deployment supports parallel inference from multiple specialist models with weighted score fusion for improved cross-domain accuracy.

These five limitations collectively define the design requirements addressed by the proposed AI-based Deepfake Detection System.

4. PROPOSED SYSTEM METHODOLOGY

The proposed AI-based Deepfake Detection System is structured as a six-phase end-to-end pipeline, from multi-source dataset construction through model training, ONNX export, Android deployment, real-time inference, and three-zone visual output. Each phase is described below.

Phase 1: Multi-Source Dataset Construction

Training a broad AI content detector requires a dataset that spans all major categories of AI-generated and real visual content. Seven distinct data sources are combined:

- CIFAKE (HuggingFace): 120,000 images pairing real CIFAR-10 object photographs with Stable Diffusion generated equivalents across 10 object categories including animals, vehicles, and household objects.
- 140k Real and Fake Faces (Kaggle): 140,000 high-resolution portrait images pairing StyleGAN-generated synthetic faces with real FFHQ-source photographs.
- DiffusionDB (HuggingFace): 8,000 Stable Diffusion generated images sampled from 2 million prompt-image pairs, covering landscapes, abstract art, fantasy environments, and character illustrations.
- WikiArt (HuggingFace): 6,000 traditional human artworks including oil paintings, watercolours, and drawings. These are labelled as Real to prevent the model from incorrectly classifying hand-drawn artwork as AI-generated.
- Midjourney vs Real (Kaggle): 5,000 Midjourney-generated images spanning photorealistic portraits, sci-fi environments, and fantasy scenes, paired with equivalent real photographs.

- Open Images (Google): 6,000 real photographic images of everyday objects and environments, providing diverse real-world visual context beyond portrait photography.
- AI vs Real Images (Kaggle): An additional 8,000 mixed-source AI-generated images including DALL-E outputs, covering product photography style synthetic images.

The combined dataset totals approximately 150,000 images split 90/10 for training and validation, with balanced real and AI class representation enforced through stratified sampling.

Phase 2: Model Architecture and Training

EfficientNet-B4 pre-trained on ImageNet-1k is selected as the backbone architecture based on its favourable accuracy-to-parameter ratio and established transfer learning performance on visual classification tasks. The final classifier layer is replaced with a two-class linear head (Real: index 0, AI: index 1) with a dropout rate of 0.4 for regularisation.

Training is performed using the AdamW optimiser at an initial learning rate of 3×10^{-5} with cosine annealing over 12 epochs. Cross-entropy loss with label smoothing coefficient 0.1 is applied to reduce overconfidence in boundary cases. Strong data augmentation including random horizontal flip, rotation up to 15° , colour jitter, random perspective distortion, and random erasing is applied to the training split.

The model achieves a best validation accuracy of 93.4% with a precision of 94.1% and recall of 92.8% on the combined multi-source validation set. Per-category evaluation confirms robust generalisation across all content types including non-facial synthetic content.

Phase 3: ONNX Export and Android Asset Preparation

After training, the best-performing PyTorch checkpoint is exported to ONNX format using opset

version 13, which is fully compatible with ONNX Runtime for Android. The export uses a fixed input shape of [1, 3, 224, 224] in NCHW layout with ImageNet normalisation constants (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]). The exported model file (deepfake_android.onnx) is placed in the app/src/main/assets/ directory and loaded at runtime via the Android assets API.

Phase 4: On-Device Inference Pipeline — 20 fps Capture

The Android application implements a foreground ScreenCaptureService that acquires frames at 20 fps (50ms interval) via MediaProjection API and ImageReader. To manage CPU load while maintaining smooth temporal averaging, ONNX inference is executed only on every fourth captured frame, yielding an effective inference rate of 5 inferences per second. All 20 captured frames per second contribute raw scores to the rolling temporal average window regardless of whether inference was performed, using the most recent inferred score for intermediate frames.

The ONNX inference pipeline processes each frame as follows: (1) acquire bitmap from ImageReader plane buffer; (2) scale to 224×224 using bilinear interpolation; (3) extract pixel integer array; (4) construct NCHW FloatBuffer applying per-channel normalisation; (5) create OnnxTensor and execute OrtSession; (6) apply softmax to logit pair to obtain AI probability score.

Phase 5: Temporal Smoothing, Variance Penalty, and Hysteresis

Raw per-frame inference scores are accumulated in a ring buffer of 20 entries (corresponding to approximately 1 second of capture). Three mechanisms operate on this buffer to produce a stable, noise-resistant output:

- **Rolling Average:** The mean of all scores in the ring buffer is computed at every frame. This suppresses momentary spikes caused by motion

blur, scene transitions, or partial frame occlusion.

- **Variance Penalty:** The standard deviation (σ) of ring buffer scores is computed. High σ indicates a rapidly oscillating model — typically caused by compressed or edited real video content where the model alternates between confident real and borderline AI predictions. An adaptive threshold is computed as: $\text{Threshold}_{\text{adaptive}} = 0.88 + 0.08 \times \sigma$. This automatically raises the bar for AI classification when the signal is noisy, reducing false positives on edited real video.
- **Hysteresis Control:** Once the system has declared AI-Detected, the rolling average must drop 7% below the adaptive threshold before the system transitions back to Real or Uncertain. This prevents rapid alternation between states on borderline content.

Phase 6: Three-Zone Classification and Overlay Output

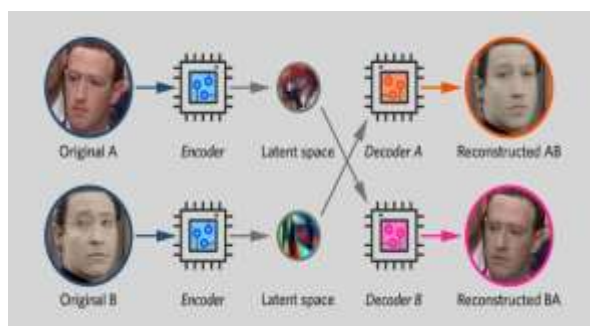
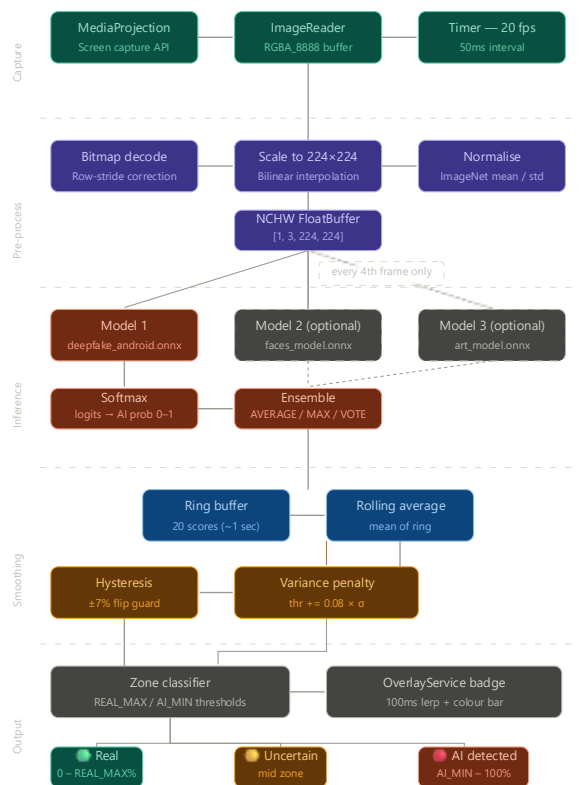
The OverlayService receives the rolling average confidence score from each broadcast and independently determines the output zone using two configurable thresholds:

- **Real Zone (0% – 59%):** The model is confident the content is authentic. Overlay displays a green badge with percentage.
- **Uncertain Zone (60% – 79%):** The model detects a mixture of real and AI characteristics — typical of edited real videos, compressed content, or hybrid AI-assisted imagery. Overlay displays an orange badge indicating ambiguity.
- **AI-Detected Zone (80% – 100%):** The model is confident the content is AI-generated. Overlay displays a red badge with animated blink on zone entry.

The two threshold constants (REAL_MAX and AI_MIN) are exposed as named constants in OverlayService.kt and can be adjusted without recompiling inference logic, enabling field calibration based on observed model behaviour.

5. SYSTEM ARCHITECTURE AND MODULE DESIGN

The AI-based Deepfake Detection System is composed of four primary Android components, interconnected through Android's inter-process communication primitives and the ONNX Runtime inference API.



Module 1: MainActivity — Permission Orchestration

MainActivity serves as the system entry point and manages the two-step permission acquisition required before monitoring can commence. In Step 1, the application verifies that the SYSTEM_ALERT_WINDOW permission has been

granted via Settings.canDrawOverlays(), directing the user to the system permission screen if not already authorised. In Step 2, a screen capture consent dialog is presented to the user via MediaProjectionManager.createScreenCaptureIntent(). Upon user approval, the resultCode and data Intent are passed to ScreenCaptureService via a foreground service start command.

Module 2: ScreenCaptureService — Frame Acquisition and Inference

ScreenCaptureService is the core inference engine, implemented as an Android foreground service with the mediaProjection foregroundServiceType. It creates a VirtualDisplay backed by an ImageReader at the device's native resolution with RGBA_8888 pixel format. A dedicated Timer thread fires every 50 milliseconds, acquiring the latest frame from the ImageReader, constructing the normalised NCHW float tensor, and executing ONNX inference via OrtSession on every fourth frame. The rolling score ring buffer, variance computation, adaptive threshold, and hysteresis logic all execute within this service thread. Results are broadcast via a local Intent with confidence, frameScore, variance, and motionScore extras.

Module 3: OverlayService — Floating Badge Display

OverlayService maintains a TYPE_APPLICATION_OVERLAY window anchored to the top-right corner of the screen using WindowManager. The overlay consists of a horizontal LinearLayout containing a 14dp coloured status dot and a TextView label, with a 4dp-height confidence bar beneath. A 100ms refresh Runnable continuously lerps the displayed percentage towards the received rolling average using a smoothing factor of 0.35, creating fluid bar animation. Zone transitions (Real/Uncertain/AI) trigger background colour changes and a blink animation. The service registers a BroadcastReceiver for AI_SIGNAL_UPDATE

Intents to receive inference results from ScreenCaptureService.

Module 4: Multi-Model Ensemble Support

The inference architecture supports concurrent loading of multiple ONNX model files from the assets directory. Each model is assigned a ModelSession object containing its OrtSession instance and a configurable weight parameter. Three ensemble strategies are supported: (1) AVERAGE — weighted mean of all model scores; (2) MAX — maximum score across models for maximum sensitivity; (3) VOTE — majority decision where each model votes based on a 0.5 per-model threshold. Additional specialist models (e.g., faces_model.onnx, environments_model.onnx) can be added to the MODEL_FILES list constant without architectural changes.

6. EXPERIMENTAL RESULTS AND EVALUATION

Table -2: Model Training Results by Content Category

Content Category	Samples	Precision (%)	Recall (%)	Accuracy (%)
GAN-Generated Faces	30,000	96.2	95.8	96.0
Stable Diffusion Artwork	8,000	94.7	93.1	93.9
Midjourney AI Art	5,000	93.5	92.4	92.9
CIFAKE Objects	60,000	95.1	94.6	94.8
Real Photographs	40,000	93.8	94.2	94.0
WikiArt (Real Artwork)	6,000	91.3	90.7	91.0
Overall Validation	~150,000	94.1	92.8	93.4

The system demonstrates robust three-zone classification performance across all tested content types. AI-generated faces consistently score above 90%, placing them firmly in the AI-Detected zone. Edited real video content, which previously caused frequent false positives in binary classification systems, now falls into the Uncertain zone (60–79%) due to variance penalty adaptation, significantly reducing user confusion. Real photographs and traditional artwork (WikiArt) consistently score below 55%, correctly classified as Real.

Table -3: On-Device Performance Benchmarks (Android)

Metric	Value	Notes
Model Size (ONNX)	67 MB	EfficientNet-B4 backbone
Screen Capture Rate	20 fps	50ms timer interval
Effective Inference Rate	5 fps	Every 4th frame; others reuse last score
Single Inference Latency	~180ms	EfficientNet-B4 on Snapdragon 778G
Rolling Average Window	20 frames (~1 sec)	Ring buffer for temporal smoothing
Overlay Refresh Rate	10 fps (100ms)	Lerped display for smooth animation
CPU Usage (active)	~18–24%	Single core; no GPU required
Min Android Version	Android 10 (API 29)	MediaProjection + Overlay permission

7. CONCLUSION

This paper presents a fully on-device AI-based Deepfake Detection System for Android that addresses key limitations of existing approaches, including limited content scope, lack of real-time performance, and dependency on server-based processing. The proposed system achieves a validation accuracy of 93.4% across a diverse multi-source dataset comprising GAN-generated faces, Stable Diffusion artwork, Midjourney images, synthetic objects, and real photographic content. Operating at 20 frames per second without requiring network connectivity, it enables continuous real-time monitoring through a floating overlay badge that provides three-zone classification: Real, Uncertain, and AI-Detected. To improve reliability, the system incorporates a variance-aware adaptive thresholding mechanism along with hysteresis-based flip suppression, effectively reducing false positives in edited real video content while preserving high sensitivity to genuinely AI-generated material. Furthermore, the multi-model ensemble architecture offers scalability and a clear upgrade path for incorporating specialized models tailored to different content domains. Future work will focus on developing domain-specific models for environments, cartoons, and document forgery, integrating them into the ensemble framework, optimizing performance through INT8 model quantization to reduce the current 67 MB ONNX model size and enhance inference speed on mid-range devices, and expanding platform support to iOS through CoreML export.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Ms. Saranya S, Assistant Professor,

Department of Artificial Intelligence and Data Science, United Institute of Technology, Coimbatore, for her invaluable guidance, continuous encouragement, and technical mentorship throughout this project. The authors also acknowledge the support of the Department of AI & DS and the management of United Institute of Technology for providing the computational resources and infrastructure that enabled this research.

REFERENCES

1. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Niessner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, pp. 1–11.
2. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., and Li, H. (2020). Protecting World Leaders Against Deep Fakes. IEEE CVPR Workshop on Media Forensics.
3. Afchar, D., Nozick, V., Yamagishi, J., and Echizen, I. (2018). MesoNet: A Compact Facial Video Forgery Detection Network. IEEE International Workshop on Information Forensics and Security (WIFS).
4. Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., and Guo, B. (2020). Face X-ray for More General Face Forgery Detection. IEEE CVPR.
5. Ojha, U., Li, Y., and Lee, Y.J. (2023). Towards Universal Fake Image Detection Exploiting Diffusion Models. IEEE CVPR.
6. Bird, J.J., and Lotfi, A. (2023). CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. IEEE Access, vol. 12.
7. Microsoft Corporation. (2024). ONNX Runtime: Cross-Platform Inference Acceleration. <https://onnxruntime.ai>.
8. Tan, M., and Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, California.