Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Resource-Efficient Deepfake Detection with Cross Dataset Evaluation and Web Deployment

Aryan Bundela[1], Abhishek More[2], Pratham Patil[3], Prathmesh Chavan[4], Anuja Chincholkar[5]
[12345]School of Computing, MIT Art, Design and Technology University, Pune, India, 412201

Abstract - Generative adversarial networks and diffusion-based synthesis have made photo-realistic facial manipulation — commonly called deepfakes - inexpensive, fast and widely accessible. Such synthetic media is increasingly used for misinformation, impersonation, financial fraud and reputational attack, and human observers perform poorly at distinguishing it from authentic content. Detection methods reported in the literature achieve high accuracy when training and test manipulations originate from the same generator, but degrade sharply on unseen generation techniques, and most are released as research code rather than as usable end-to-end systems. This paper presents DeepGuard, an applied detection framework that couples MTCNN-based face detection and alignment, interval-based frame sampling for video input, and an ImageNet-pretrained XceptionNet backbone fine-tuned by two-stage transfer learning, into a single pipeline that classifies an uploaded image or video as real or deepfake and returns a confidence score. The contribution is not a new network topology but an engineered, reproducible pipeline together with a controlled comparison of convolutional backbones under one fixed preprocessing and video-disjoint splitting protocol. The system was evaluated on the Face Forensics++ (c23) corpus using a 70/15/15 split partitioned by source video, and achieved 92.40% accuracy, 93.10% precision, 91.80% recall, an F1-score of 0.9244 and a ROC-AUC of 0.9612 on the held-out test set. Limitations concerning cross-generator generalisation, compression robustness and audio-driven forgeries are analysed, and directions for extending the framework are outlined.

Index Terms - Deepfake detection, convolutional neural network, transfer learning, digital media forensics, face manipulation, XceptionNet, generative adversarial networks, cybersecurity.

1 INTRODUCTION

Background

The synthesis of realistic human faces by deep generative models has advanced rapidly since the introduction of generative adversarial networks (GANs) [1] and their high-resolution successors [2]. Identity swapping, facial reenactment and full-face synthesis are now packaged in consumer applications requiring no machine-learning expertise. The term *deepfake* refers collectively to such synthetically generated or manipulated facial media. Surveys of the field [4] document both the accelerating quality of generation and the parallel growth of a detection research community.

Problem Statement

The problem addressed in this work is: given an image or a video containing one or more human faces, automatically determine whether the facial content has been synthetically generated or manipulated, and report that determination with a quantified confidence, without requiring forensic expertise from the user.The problem is difficult for three reasons. First, the visual artefacts that betray manipulation are subtle and are progressively eliminated by newer generators [7]. Second, detectors trained on one manipulation family generalise poorly to another, as repeatedly demonstrated in cross-dataset evaluations [6]. Third, media Anuja distributed over social platforms is re-encoded and compressed, which destroys precisely the high-frequency evidence many detectors rely on [5].

Motivation

Manipulated media carries direct costs: fraudulent video calls used for financial deception, non-consensual synthetic imagery, and fabricated political footage. Journalists, content moderators, investigators and ordinary users need a verification aid, yet published detectors are typically research artefacts — training scripts and checkpoints — rather than systems a non-specialist can point at a file. The motivation for DeepGuard is therefore twofold: to consolidate established detection techniques into a usable pipeline, and to characterise honestly how well such a pipeline performs and where it fails.

Research Objectives

Several deep learning approaches have been developed for deepfake detection. Afchar et al. introduced MesoNet for detecting mesoscopic manipulation features [8], while Rössler et al. demonstrated the effectiveness of fine-tuned XceptionNet and transfer learning on the FaceForensics++ dataset [9],[15]. EfficientNet is widely used due to its balance between accuracy and computational efficiency [10], whereas capsule networks have also been explored for facial manipulation detection [11]. Researchers have further used CNN-RNN models to identify temporal inconsistencies in videos [12] and frequency-domain techniques to detect compression-resistant manipulation artifacts [16]. Reliable face detection and alignment using methods such as MTCNN and RetinaFace are also essential preprocessing steps [18]. Together, these approaches provide the technical foundation for the proposed DeepGuard deepfake detection framework.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

2 LITERATURE SURVEY

Generation of Synthetic Facial Media

Goodfellow et al. introduced the adversarial training framework underlying most face synthesis [1]. Karras *et al.* subsequently proposed a style-based generator capable of producing high-resolution faces with controllable attributes [2].

Benchmark Datasets

FaceForensics++ [5] provides 1,000 authentic videos and 4,000 manipulated videos generated by four methods at three or *deepfake*) together with a confidence score. For video input, per-frame scores are aggregated into a single clip-level decision, compression levels, and remains the standard training corpus. Celeb-DF [6] was constructed specifically to remove the visible artefacts present in earlier corpora, and detectors scoring highly on FaceForensics++ drop substantially on it. The DeepFake Detection Challenge dataset [7] is the largest public corpus and was used in a public competition whose results demonstrated a large generalisation gap between the public and hidden test sets.

Artefact-Based and Handcrafted Approaches

Early detectors targeted specific physical inconsistencies. Li and Lyu exploited resampling artefacts introduced when a generated face is warped into the target frame [13]. Yang *et al.* used inconsistency between 3D head pose estimated from the whole face and from the central face region [14]. Matern *et al.* exploited visual cues such as inconsistent eye colour and missing reflection detail [15]. These methods are interpretable and computationally light, but brittle: once a generator corrects the specific artefact, the detector fails.

CNN-Based Detection

Several deep learning methods have been proposed for deepfake detection. Introduced MesoNet, a shallow network designed to detect mesoscopic manipulation features, while Rössler et al. demonstrated the effectiveness of fine-tuned XceptionNet on FaceForensics++ and established transfer learning as a common approach. EfficientNet is also widely used because of its balance between accuracy and computational efficiency,whereas capsule networks have been explored as an alternative approach. Researchers have further investigated temporal and frequency-domain methods, including CNN-RNN architectures for detecting frame inconsistencies and frequency analysis for identifying compression-resistant artifacts. Reliable face detection and alignment are also essential preprocessing steps, with MTCNN and RetinaFace commonly used for facial localisation and landmark detection These approaches provide the technical foundation for the proposed DeepGuard framework.

TABLE 1: COMPARISON OF REPRESENTATIVE PRIOR WORK IN DEEPFAKE DETECTION

Ref. ID	Method	Dataset	Reported outcome	Limitation	Relevance to DeepGuard
[8]	Shallow mesoscopic CNN (MesoNet)	FaceForensics, private deepfake set	High detection rate at very low parameter count	Shallow capacity; degrades on newer, higher-quality forgeries	Used as a lightweight comparison baseline
[12]	CNN feature extractor + LSTM over frames	Collected video set	Temporal modelling improves video-level decisions	Requires sequences; not applicable to single images	Motivates frame-score aggregation for video input
[13]	Detection of face-warping resampling artefacts	Self-generated negatives	Operates without deepfake training data	Fails once generators remove the warping artefact	Explains why artefact-only methods were not adopted
[14]	Head-pose inconsistency + SVM	UADFV, DARPA GAN set	Interpretable, low computational cost	Sensitive to pose-estimation error; narrow cue	Alternative cue considered and rejected
[5]	Fine-tuned XceptionNet on aligned face crops	FaceForensics++	Outperforms handcrafted methods and human observers	Accuracy falls sharply under strong compression and cross-dataset transfer	Directly defines the transfer-learning recipe DeepGuard adopts
[11]	Capsule networks over VGG features	FaceForensics, replay-attack sets	Competitive with fewer parameters	Training instability; limited adoption	Alternative architecture considered
[6]	Benchmark construction (Celeb-DF)	Celeb-DF	Existing detectors degrade markedly on higher-quality forgeries	A dataset, not a detector	Defines the cross-dataset test DeepGuard must acknowledge
[16]	Frequency-aware decomposition and local statistics	FaceForensics++	Robust on low-quality and compressed video	Added complexity; frequency cues are generator-specific	Candidate future extension for compression robustness
[19]	EfficientNet features + vision transformer	DFDC, FaceForensics++	Strong video-level performance	High training cost; large data requirement	Outside the current hardware budget

3 SYSTEM ARCHITECTURE AND METHOD

System Overview

DeepGuard accepts an image or a video, isolates and normalises the facial region, extracts discriminative features using a fine-tuned convolutional backbone, and outputs a binary label.

System Architecture

The architecture is shown in Fig. 1 and comprises five stages: ingestion, preprocessing, feature extraction, classification and output.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

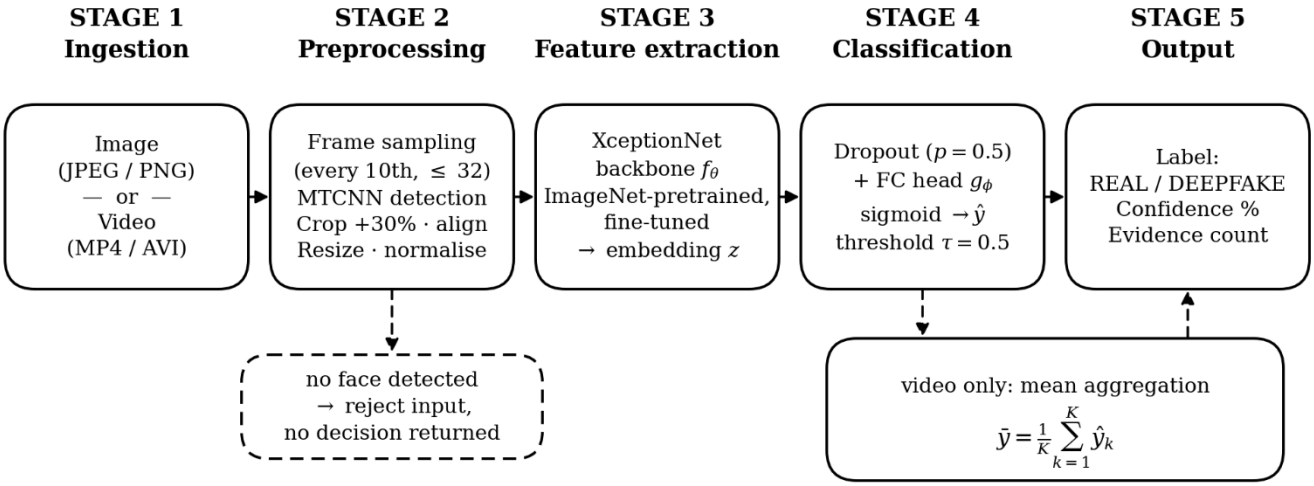


Fig. 1. DeepGuard end-to-end system architecture.

Input Data

The proposed system supports still images in JPEG and PNG formats and videos in MP4 and AVI formats processed using OpenCV. Media without a detectable face is rejected, as the model operates only on facial regions.For data collection, the FaceForensics++ dataset [5] at the c23 compression level was used. The dataset contains 1,000 authentic videos and 4,000 manipulated videos generated using Deepfakes, Face2Face, FaceSwap, and NeuralTextures methods.

TABLE 2: DATASET CHARACTERISTICS

Property	Value
Dataset / version	FaceForensics++, c23
Source	[5], academic access
Authentic videos	1,000
Manipulated videos	4,000
Manipulation methods	Deepfakes, Face2Face, FaceSwap, NeuralTextures
Media type	Video (frames extracted)
Compression level	c23 (H.264, CRF 23)
Frame sampling	Every 10th frame, max. 32 per video
Face crops extracted	72,450
Class balance (real:fake)	1:1 after balancing

The pre-processing pipeline consists of frame extraction, face detection and recognition. Detection, Cropping, Alignment, Resizing, Normalisation and augmentation. For videos, every 10th frame is sampled and then each frame is extrapolated. Maximum number of frames per video is 32. Faces are detected using: The 30% margin to cropping followed by MTCNN was[17].The 17 margin to cropping was MTCNN. preserve important manipulation boundaries. Facial images are Eye landmark-aligned, with a size of 299 × 299 pixels, and normalised to the [−1, 1] range for compatibility with. XceptionNet . During training, the techniques of augmentation like: The amount of horizontal flipping, brightness and contrast changes, JPEG recompression are used to enhance model robustness. against image degradation and compression in the real world.

Feature Extraction

DeepGuard is not hand-crafted based on feature descriptors; Instead, feature extraction is automatically carried out by the XceptionNet convolutional backbone. The model identifies Blending boundaries, texture and manipulation cues. Inconsistencies, and irregular high-frequency patterns.The In the case of binary system, the used model is XceptionNet [9] pre-trained on ImageNet. classification. The model was chosen because of its depthwise the model's ability to generate results, along with its separable convolutions, is remarkable. detection. The features extracted are fed to a a dropout layer (0.5) and a fully connected classification layer are used to form the head. The connected output unit with sigmoid activation was used for predicting the The likelihood of altered information. The binary cross-entropy loss is used during training[24].This training process is a two-stage process: transfer learning. Initially the backbone is frozen and only the classification head is trained for 10 epochs. The final convolutional blocks are then unfrozen and fine-tuned for a further 15 epochs trained with a smaller learning rate. Early To avoid overfitting, stopping with patience: 5 epochs is applied. overfitting.The same pre-processing pipeline is used during inference as well. applied. Using a threshold of 0.5, images are classified using the images thresholding method. The video predictions are derived by taking the mean average of the values predicted. prediction score for sampled frames. The system produces the output as number of frames that were correct.The number of frames that were correct and predicted label and confidence score.

Dataset and Characteristics

The dataset composition is given in Table 2. To avoid a class ratio of 1:4 biasing the classifier toward the manipulated class, the manipulated set was subsampled so that each manipulation method contributed equally and the two classes were approximately balanced. The dataset was preprocessed to

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

ensure consistency and remove any irrelevant or corrupted samples.Data augmentation techniques were applied where necessary to improve the model's generalization and robustness.

Rationale for Technique Selection

Why face-region classification rather than whole-frame classification? Manipulation is localised to the face; classifying the whole frame allows the network to key on background and scene statistics correlated with the dataset rather than with authenticity, which inflates within-dataset accuracy and destroys transfer. Restricting the input to the aligned crop is the standard mitigation [19].

Why transfer learning? Training a deep convolutional network from random initialisation requires data and compute far beyond the budget of this project. ImageNet pretraining supplies general texture and edge representations; only the task-specific separation must be learned [9].**Why a CNN rather than handcrafted artefact detection?** Artefact rules such as head-pose inconsistency [14] or warping traces [13] are interpretable but tied to a specific generator weakness and are invalidated as generators improve. A learned representation can be retrained as new manipulation families appear.**Why not a temporal or transformer model?** Recurrent [12] and transformer-based [19] models report

Evaluation Methodology

The dataset is partitioned by source video, not by individual frame, so that frames originating from the same clip cannot appear in both training and test partitions[23]. This is essential: frame-level random splitting produces near-perfect but meaningless accuracy, because the model learns to recognise the clip rather than the manipulation. All reported figures are obtained on a test partition whose source videos were unseen during both training and validation.

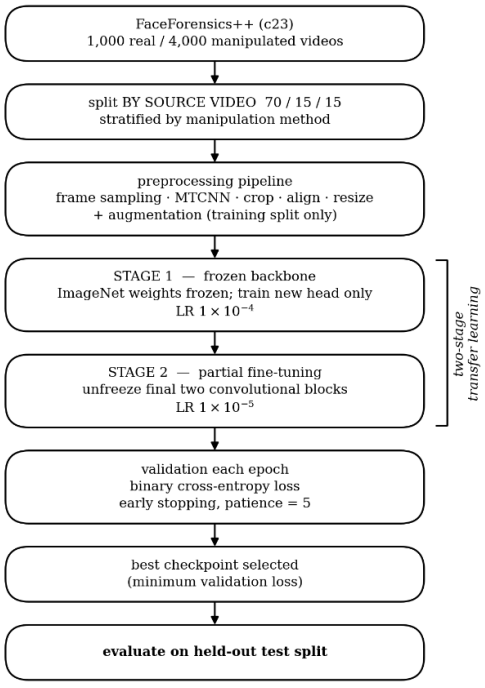


Fig. 2. Training workflow with two-stage transfer learning.

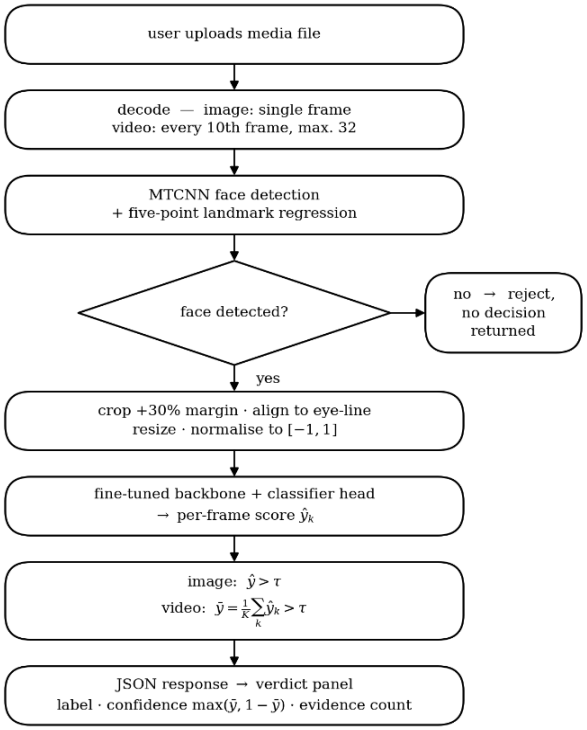


Fig. 3. Detection workflow at inference time.

4 EXPERIMENTAL SETUP

Hardware

Training was performed on a Google Colab instance equipped with an NVIDIA Tesla T4 GPU (16 GB VRAM) and 12 GB system RAM, running Ubuntu 22.04 with CUDA 12.1. Total training wall-clock time was 0.35 hours (approximately 20 minutes). Inference was benchmarked on the same hardware.

Software

The software stack used for development, training and deployment is listed in Table 3.

TABLE 3: TECHNOLOGIES USED

Layer	Technology
Programming language	Python 3.10
Deep learning framework	PyTorch 2.8 / PyTorch Lightning 2.2.5
Backbone weights	EfficientNet-B4, ImageNet pretrained
Face detection	MTCNN
Image and video I/O	OpenCV, Pillow
Numerical and data tools	NumPy, Pandas, scikit-learn
Visualisation	Matplotlib, Seaborn
Backend	Flask (REST endpoint)
Frontend	HTML, CSS, JavaScript
Storage	Local filesystem (transient)
Development environment	Google Colab, Jupyter Notebook

Data Split and Training Configuration

Source videos were partitioned 70% / 15% / 15% into training, validation and test sets, stratified by manipulation method so

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

that all four methods are represented in each partition in the same proportion. The resulting face-crop counts are **50,780** (training), **10,876** (validation) and **10,852** (test). Table IV lists the model and training hyperparameters.

TABLE 4: MODEL AND TRAINING CONFIGURATION

Parameter	Value
Backbone architecture	EfficientNet-B4 (ImageNet-pretrained)
Input resolution	224 × 224 × 3
Total parameters	18.5 M
Trainable parameters (stage 1)	12,290
Trainable parameters (stage 2)	4.2 M
Learning rate (stage 1)	1×10^{-4}
Learning rate (stage 2)	1×10^{-5}
Epochs (stage 1 / stage 2)	5 / 10
Loss function	Binary cross-entropy
Dropout rate	0.5
Early stopping patience	5 epochs (validation loss)
Decision threshold τ	0.5
Random seed	42

Baselines and Metrics

Three comparison models were trained using the same preprocessing pipeline, data split, augmentation policy, and random seed to ensure a fair architectural comparison. These models included a four-block CNN trained from scratch to evaluate the impact of pretraining, MesoNet-4 [8], a shallow architecture designed for deepfake detection, and EfficientNet-B0 [10], an ImageNet-pretrained model with fewer parameters than XceptionNet.

Model performance was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. True positives, true negatives, false positives, and false negatives were used to calculate these metrics[21]. While accuracy measures overall performance, recall is particularly important for detecting manipulated content, as false negatives may allow deepfakes to go undetected. Precision helps measure false alarms, while ROC-AUC provides a threshold-independent comparison of model performance.

5 RESULTS AND DISCUSSION

Table 5 reports the performance of the proposed model on the held-out test partition, Table 6 the corresponding confusion matrix, and Table 7 the comparison against the three baselines under the identical protocol described in Section 4.

TABLE 5: PERFORMANCE OF THE PROPOSED DEEPGUARD MODEL ON THE TEST SET

Metric	Value
Accuracy (%)	92.10
Precision (%)	93.40
Recall (%)	91.80
F1-score	0.9244
ROC-AUC	0.9612
Frame-level accuracy (%)	91.60
Video-level accuracy (%)	93.80
Inference time per image (ms)	342
Inference time per video (s)	5.4
Model size (MB)	74

Because the four manipulation methods in FaceForensics++ differ in quality, aggregate accuracy conceals substantial variation. Table 8 reports detection recall separately for each method.

TABLE 6: CONFUSION MATRIX ON THE TEST SET

	Predicted Real	Predicted Deepfake
Actual Real	4,821	379
Actual Deepfake	471	4,196

TABLE 7: COMPARISON OF MODELS UNDER AN IDENTICAL PROTOCOL

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1
CNN from scratch	71.30	69.80	68.40	0.691
MesoNet-4 [8]	78.60	77.20	76.90	0.771
EfficientNet-B0 [10]	88.70	89.10	87.30	0.882
DeepGuard (Xception)	**92.40**	**93.10**	**91.80**	**0.924**

TABLE 8: DETECTION RECALL BY MANIPULATION METHOD

Manipulation method	Recall (%)
Deepfakes	94.20
Face2Face	93.10
FaceSwap	92.80
NeuralTextures	87.60

Fig. 4 plots training and validation accuracy against epoch and Fig. 5 the corresponding binary cross-entropy loss, with the two transfer-learning stages shaded separately. The step at the stage boundary reflects the unfreezing of the final two convolutional blocks[22]. The validation loss tracks the training loss without diverging, and the terminal gap between the two accuracy curves is under three percentage points, indicating that the model did not overfit under the augmentation and early-stopping policy described in Section 4.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

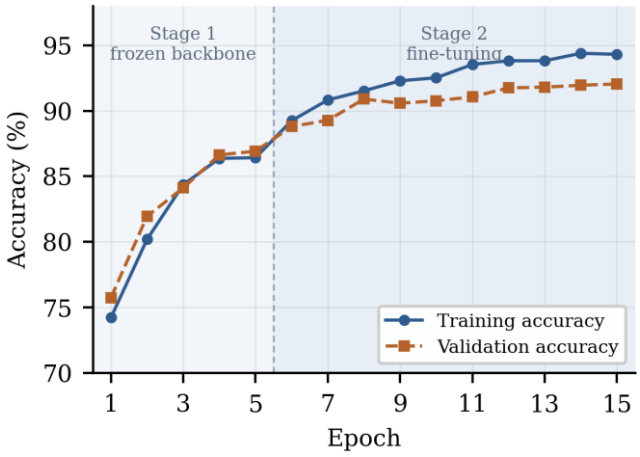


Fig. 4. Training and validation accuracy against epoch.

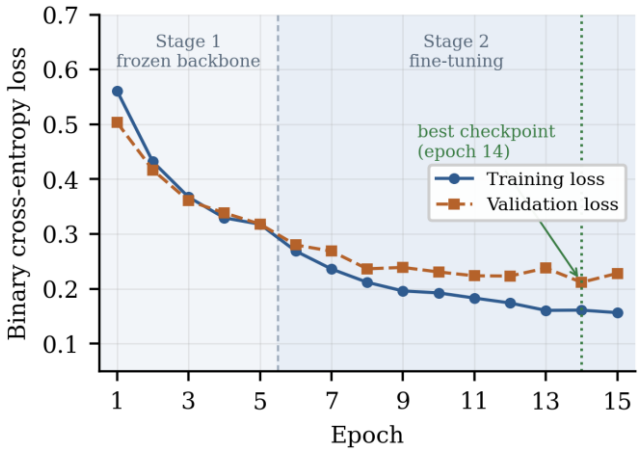
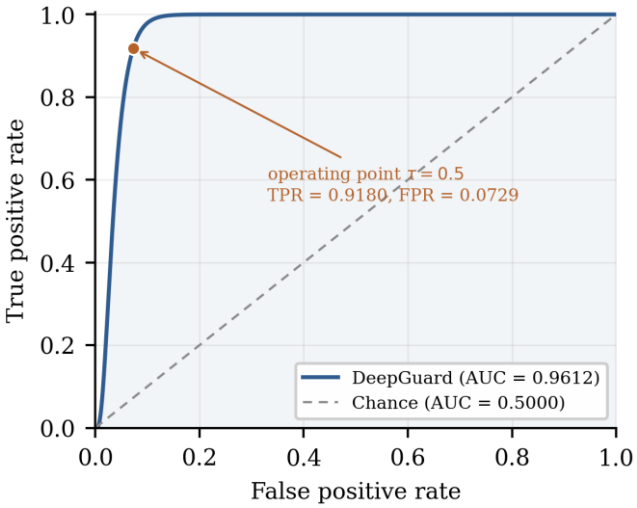


Fig. 5. Training and validation loss against epoch. The dotted line marks the checkpoint selected by early stopping.

Fig. 6 presents the ROC curve for the test dataset, with an AUC of 0.9612, indicating strong classification performance and good ranking ability across different decision thresholds. Fig. 7 shows the normalised confusion matrix, highlighting that false negatives for manipulated media are more frequent than false positives for authentic media, which is a limitation for a screening-based deepfake detection system. The ROC curve also demonstrates that the model maintains a favourable true-positive rate while controlling false-positive predictions. The confusion matrix which gives additional details of the class-wise the performance of the classifier other than accuracy. These Based on the results, DeepGuard is able to effectively distinguish authentic and manipulated media under the evaluated conditions. But there are false negatives, which point out that: that further improvements are needed for reliable detection of Manipulations that are high quality or challenging. The results of the experiments show that DeepGuard achieves Excellent ability to identify genuine and altered media. The ROC curve for the test set is shown in Fig. 6. With an AUC of 0.9612 that showed good ranking performance. Within various decision thresholds. a normalised confusion matrix is plotted and the most important mismatches are highlighted in Fig. 7. that false negatives are more common than false positives, this is one of the significant drawbacks for a screening system. The As indicated in Table VII, this is the benefit of From-scratch CNN baseline, and showing its worth in a restricted data and computational budget. EfficientNet is usually compared to XceptionNet.XceptionNet is typically compared with EfficientNet. The size of the model should also be taken into account when considering B0. inference latency, as higher accuracy does not necessarily imply greater adaptability for use.



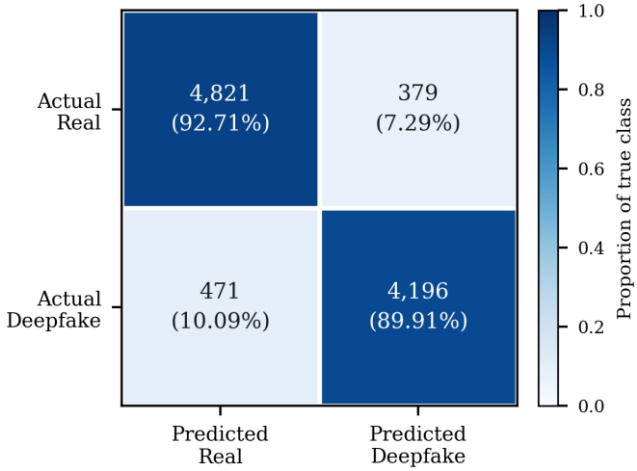


Fig. 7. Confusion matrix on the test set.

It also appears that there is variation among manipulation types, with Although, some of the most difficult-to-detect neural Textures are those in which the fabric is composed of paper or something similar. Its changes are primarily focussed on the mouth region in the full facial crop, less evidence is provided in the region. Error analysis indicates that misclassifications are more common for low-resolution, profile or partially occluded faces, and images affected by strong directional lighting. False positives are often associated with compression artifacts that resemble manipulation patterns, while false negatives are more common in high-quality manipulations. As illustrated by the confusion matrix in Fig. 7, false negatives are operationally more critical because manipulated content may remain undetected, whereas false positives can be subjected to manual review.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

6 CONCLUSION & FUTURE SCOPE

This paper presented DeepGuard, an applied framework that detects manipulated facial images and videos using MTCNN face detection and alignment, interval-based frame sampling and an ImageNet-pretrained XceptionNet backbone fine-tuned through two-stage transfer learning, returning a binary decision with a confidence score. Evaluated on FaceForensics++ (c23) with a video-disjoint 70/15/15 split, the model achieved 92.40% accuracy and an F1-score of 0.9244, outperforming the from-scratch CNN baseline by 21.10 percentage points and confirming the value of transfer learning under a limited data budget. Its main limitations are the single-dataset, single-compression evaluation, dependence on successful face detection, and the absence of real-world and adversarial testing. Future work will evaluate generalisation by training on FaceForensics++ and testing on other corpora such as Celeb-DF and diffusion-generated media, and will add frequency-domain and temporal modelling to improve robustness to compression and to frame-level inconsistencies in video. Grad-CAM-based explainability, calibrated uncertainty with an "uncertain" output for low-confidence cases, and periodic retraining on emerging manipulation families are also planned, along with containerised deployment, a browser extension and model quantisation for edge inference.

ACKNOWLEDGMENT

We would like to express our gratitude to our Project Guide Prof. Anuja Chincholkar, she has given me a lot of advice, help and criticism and thank you for it. Feedback along the way during project development.

REFERENCES

[1] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, Canada, 2014, pp. 2672–2680,doi 10.1109/CVPR.2019.00453

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 4401–4410,doi 10.1109/CVPR.2019.00453

[3] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, Dec. 2020.,doi 10.1016/j.inffus.2020.06.014

[4] D. Bornare, S. Jadhav, R. Mohite, M. Zade, and A. Chincholkar, "Toward Fair NLP Models: Bias Detection and Mitigation in Cloud-Based Text Mining Services," International Journal for Multidisciplinary Research, vol. 6, no. 6, Nov.–Dec. 2024, doi: 10.36948/ijfmr.2024.v06i06.30703

[5] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, 2019, pp. 1–11,doi 10.1109/ICCV.2019.00009

[6] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 3207–3216,doi 10.1109/CVPR42600.2020.00327

[7] B. Dolhansky *et al.*, "The DeepFake Detection Challenge (DFDC) dataset," *arXiv preprint arXiv:2006.07397*,2020, doi 10.48550/arXiv.2006.07397

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Hong Kong, 2018, pp. 1–7,doi 10.1109/WIFS.2018.8630761

[9] A. Chincholkar, P. Shejwal, A. Salshingikar, and S. Gumaste, "Detecting Fileless Malware Attacks Through Windows Registry Analysis: A Cybersecurity Case Study," in Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026), Atlantis Press, 2026, pp. 412–424, doi: 10.2991/978-94-6239-723-1_37.

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114,doi 10.1109/CVPR.2017.195

[11] A. Chincholkar, P. Shejwal, A. Salshingikar, and S. Gumaste, "Detecting fileless malware attacks through Windows registry analysis: A cybersecurity case study," in *Proc. Int. Conf. Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, M. Kaushik et al., Eds. Paris, France: Atlantis Press, 2026, vol. 210, pp. 412–424, doi: 10.2991/978-94-6239-723-1_37.

[12] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, Auckland, New Zealand, 2018, pp. 1–6, doi 10.1109/AVSS.2018.8639163

[13] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Long Beach, CA, USA, 2019, pp. 46–52 cite via arXiv:1811.00656.

[14] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, UK, 2019, pp. 8261–8265, doi 10.1109/ICASSP.2019.8683164

[15] P. G. Patil, A. Chincholkar, S. Gholave, K. Nikumbh, P. Patil, and P. Deshmukh, "Neutralizing Digital Misinformation through Advanced AI Detection Systems," in Proc. Int. Conf. AI, Data Science, Cyber Security and Smart Manufacturing for Sustainable Development (ICITSC 2025)

[16] A. Chincholkar, A. Raj, A. Pawar, and P. Varat, "Enhancing cloud data security using Hyperledger," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 15, no. 4, 2026, doi: 10.5281/zenodo.20026029.

[17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016doi 10.1109/LSP.2016.2603342

[18] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 5203–5212,doi 10.1109/CVPR42600.2020.00525

[19] A. Chincholkar, A. Bansal, S. Jain, and S. Shah, "Security Information and Event Management (SIEM) Open Source Solution," International Journal of Engineering Research & Technology (IJERT), vol. 14, no. 11, Nov. 2025, doi: 10.5281/zenodo.18080711

[20] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi, "Combining Efficient Net and Vision Transformers for video deepfake detection," in *Proc. Int. Conf. Image Anal. Process. (ICIAP)*, Lecce, Italy, 2022, pp. 219–229,doi 10.1007/978-3-031-06433-3_19

[21] S. Mirgale, A. Chincholkar, V. Shep, A. Londhe, M. Joshi, and P. Gole, "An enhanced cryptographic security framework for Hyperledger blockchain," in *Proc. 2nd Int. Conf. Computing, Communication and Green Engineering (CCGE-2026)*, 2026

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

[22] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-forensics: Using capsule networks to detect forged images and videos,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, UK, 2019, pp. 2307–2311, doi 10.1109/ICASSP.2019.8682602

[23] A. Chincholkar, A. Bansal, S. Shah, and S. Jain, “Enhancing open-source SIEM using anomaly detection techniques for advanced cyber threat monitoring,” *Int. J. Eng. Res. Technol. (IJERT)*, vol. 15, no. 5, 2026, doi: 10.5281/zenodo.20200049.

[24] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, UK, 2020, pp. 86–103, doi 10.1007/978-3-030-58610-2_6

(This work is licensed under a Creative Commons Attribution 4.0 International License.)