Video and Image Manipulation Detection Based on ResNet and EfficientNet

Stephanus Hermawan Susanto, Rafli Irfansyah Kusumawardhana*, Ariq Sudibyo*, Fiqki Haidar Amrulloh*, Aisya Ulul Asmi*, Wahyono⁴***

Department of Computer Science and Electronics, Universitas Gadjah Mada
**Authors contributed equally*
***corresponding author*

Abstract— **This study examines the performance of EfficientNet-B7 and ResNet50 for deepfake image detection using the FaceForensics++ and Celeb-DF-v2 datasets. A two-phase training strategy, starting with a frozen backbone and later fine-tuning the entire network was adopted mainly to keep training stable in the early stages. EfficientNet performs strongly on FaceForensics++ (90.1% accuracy) and 97.6% accuracy on Celeb-DF-v2. ResNet50 followed with 93.5% accuracy. However, when both models were tested on two completely unseen videos, the results were far less reassuring: EfficientNet-Celeb-DF-v2 misclassified both videos, ResNet-FF++ misclassified the real video as fake, while EfficientNet-FF++ and ResNet-Celeb-DF-v2 misclassified the fake video as real. This mismatch between benchmark success and real-world behavior suggests that current models may be learning dataset quirks more than actual manipulation. The findings point toward the need for stronger domain-adaptive training methods, cross-dataset learning, and possibly multimodal features to build detectors that hold up outside controlled test environments.**

Keywords— deepfake detection, EfficientNet-B7, ResNet50, face forensics

I. Introduction

In today's digital era, the rise of easily accessible deepfake applications has become a serious issue. Deepfake technology uses machine learning algorithms to produce fake content that looks very realistic and is hard to distinguish from the original[1][2]. The misuse potential keeps increasing because these deep learning algorithms are now easy to access and affordable. When this technology is used by irresponsible people, it leads to more cybercrimes. One example is the spread of manipulated photos or videos aimed to damage someone's reputation[2][3]. A study even shows that 96 percent of detected deepfakes are non-consensual pornographic content, often involving edited photos without the person's permission[1][5]. Because social media platforms contain huge amounts of data, fake content can spread quickly, so tools that can automatically detect fake images and videos are needed.

Generative Adversarial Networks (GANs) [2] are one of the main technologies used to create high-quality deepfakes, especially for face generation and manipulation[3]. Several tools have been introduced to detect deepfakes, but many still have limitations. For example, Deepware Scanner[4] cannot detect GAN-generated face images or images manipulated by traditional methods, and Google Assembler has restricted access. Microsoft Video Authenticator also does not perform well on GAN-generated images[6].

This research proposes a focused experimental framework for deepfake detection that benchmarks complementary deep architectures: a ResNet‑based model (e.g., ResNet50) as a robust per‑frame classifier for GAN‑face detection, and an EfficientNet‑based model (e.g., EfficientNet‑B4) optimized for video‑level detection and efficiency. The framework systematically compares these architectures on accuracy to identify which designs best for detecting fake generated video.

II. Research Methods

A. Data

The dataset used in this study was obtained from FaceForensic++ and Celeb-DF-v2. The FaceForensics++ dataset contains 1,000 real videos, and each of these videos is then forged using automated methods such as Deepfakes, Face2Face, FaceSwap, and NeuralTextures[7]. Meanwhile, the Celeb-DF-v2 dataset contains 590 real videos and 5,639 fake videos generated from those real videos[8].

B. Data preprocess

The FaceForensics++ and Celeb-DF-v2 dataset is provided in the form of video clips, so it cannot be used directly for image-based deepfake classification. Since the EfficientNet and ResNet model requires face images as input, the videos need to be converted into frame-level samples first. For that reason, a preprocessing pipeline is used to extract faces from selected frames of each video, and the resulting cropped face images become the actual data used for training and validation.

C. Data Preparation

For FaceForensic++, the preprocessing stage begins by reading the metadata file, which contains the video paths and their corresponding labels (real or fake). While Celeb-DF-v2, preprocessing iterates over real videos folder and their matched synthesis counterparts, samples evenly spaced frames, and limits fake samples per real video for balance. It runs MTCNN to detect the largest face per frame, crops and resizes the face regions, saves the images to disk, and writes per-image metadata containing filepaths, bboxes, confidences, frame indices and labels.

D. Configuration Setup

Several parameters are defined before running the face extraction. These include the output image size (256×256), the number of frames to sample from each video, and the location of the dataset. InsightFace (FaceAnalysis with the “buffalo_l” model) and MTCNN is initialized to detect faces from each video frame. GPU support is enabled to speed up detection.

E. Data Splitting

Instead of splitting by hand, the pipeline uses a simple hashing method to automatically place each video into the training or validation set. About 70% of videos go into the training set and the remaining 30% into the validation set. This avoids mixing frames from the same video across different splits.

F. Face Extraction Process

For every video in the dataset, the total number of frames is checked first. Then, 10 frame indices are selected. Each selected frame is passed through the face detector. If a face is found, the largest face in the frame is chosen, cropped from the original image, resized to 256×256, and saved. Videos with no detectable faces or broken frames are skipped automatically. Each cropped face is saved using a filename based on the video name and frame index, placed under the correct folder: real or fake, and train or validation. This turns the original video dataset into a clean set of face images that can be used directly for training a deepfake classification model.

G. Models

a. Efficientnet-B7

EfficientNet is a deep learning model based on a convolutional neural network (CNN) architecture introduced by Tan and Le in 2019[9]. The goal of this model is to achieve high accuracy without using excessive computational resources. Previous approaches usually scaled the network by only increasing the depth, widening the architecture, or enlarging the input image size separately. EfficientNet takes a different approach. It uses a method called compound scaling, where the depth, width, and resolution are scaled together in a balanced way. This makes the model stronger without becoming too heavy. The compound scaling method uses a compound coefficient ϕ that uniformly scales the width, depth, and resolution.

1. depth: $d = \alpha^\phi$
2. width: $w = \beta^\phi$
3. resolution: $r = \gamma^\phi$
4. subject to $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$
5. $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$

Where:

1. α: scaling factor for network depth (usually between 1 and 2)
2. β: scaling factor for network width (usually between 1 and 2)
3. γ: scaling factor for image resolution (usually between 1 and 2)
4. ϕ (phi): compound coefficient (a positive integer) that controls the scaling

This architecture has several versions, from B0 to B7. Each version provides a different balance between performance and resource requirements. B0 is the smallest version, making it lightweight and fast, suitable for limited hardware or real-time systems. The largest version, B7, is much stronger in accuracy but needs more computational power. The choice of version can be adjusted depending on the priority of the task.

In this study, EfficientNet B7 is used to detect deepfake images.

b. ResNet

ResNet is a CNN architecture introduced by He, Zhang, Ren, and Sun in 2016, and it comes in several versions such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152[10]. This architecture consists of convolutional layers, pooling layers, and multiple residual blocks that help extract complex visual features. It also includes skip-connections to deal with the vanishing gradient problem[11]. This allows ResNet to be trained effectively even with deeper networks and enables the model to detect fine details in an image[112].

In this study, ResNet is used for the fake facial image classification task because it has shown good performance in extracting complex features and capturing small details in images, which is expected to help achieve high classification accuracy.

H. Evaluation Metrics

The model performance in this study is evaluated using accuracy, precision, recall, and F1-score. These four metrics are chosen because they provide a clear picture of how well the model can distinguish real data from manipulated data. Precision shows how many of the positive predictions are actually correct, while recall measures how many of the actual positive samples are successfully detected. The F1-score is used to balance both metrics so the evaluation does not depend on only one side.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

Description:

TP : True Positive

FP : False Positive

TN : True Negative

FN : False Negative

III. Result And Discussion

After implementing the EfficientNet and ResNet architectures for deepfake image detection, both models were trained and evaluated on the FaceForensics++ and Celeb-DF-v2 datasets. These datasets contain a diverse mixture of authentic and manipulated facial images, enabling the models to be examined under a variety of forgery scenarios.

For the FaceForensics++ experiment, the model was trained in one stage for 20 epochs with the whole network updated from the beginning. From the loss curves, the training loss keeps going down steadily, which means the model is continuously learning. The validation loss also generally decreases, but it goes up and down at several points, showing that the model still struggles on some validation batches even though the overall trend is improving. Meanwhile, the validation accuracy keeps increasing from about 0.69 to around 0.86 by the end of training. So even though the validation loss isn't perfectly smooth, the final results suggest that the model is still learning the dataset well across the 20 epochs which can be seen in Fig. 3. ResNet50 demonstrates faster initial convergence but shows clear signs of overfitting. As illustrated in Fig. 4 Training accuracy reaches above 0.99 by epoch 5, while validation accuracy plateaus at 0.85-0.86. The training loss drops to near zero within 10 epochs, but validation loss fluctuates erratically between 0.35 and 0.70, with spikes at epochs 6-7 and 14-15.

For the Celeb-DF-v2 experiments in particular, training was carried out in two stages: Phase 1 focused on updating only the classifier head while keeping the backbone frozen, and Phase 2 then fine-tuned the entire network end-to-end. This progressive strategy provided a more stable optimization path, limiting early overfitting and allowing the models to gradually adapt to dataset-specific features. As illustrated in Fig. 1 and Fig. 2, the training curves for both ResNet50 and EfficientNetB7 clearly highlight the transition between the two training phases, showing that while Phase 1 offers only modest improvements with the backbone frozen, the onset of full end-to-end fine-tuning in Phase 2 leads to a pronounced drop in loss and a substantial, consistent rise in accuracy for both models.

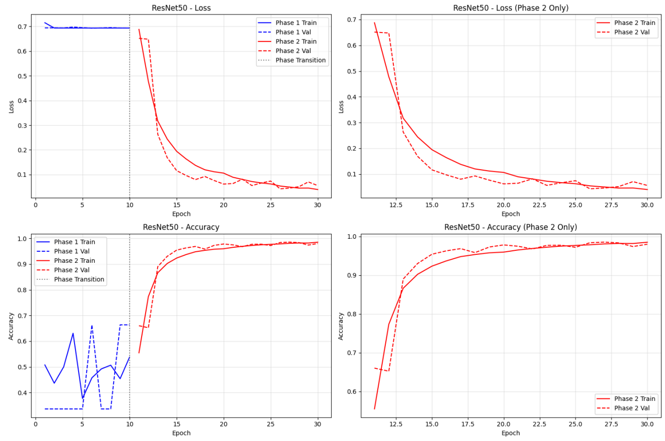

Fig 1. Training and validation curves of the ResNet50 model across two-phase training on Celeb-DF-v2.

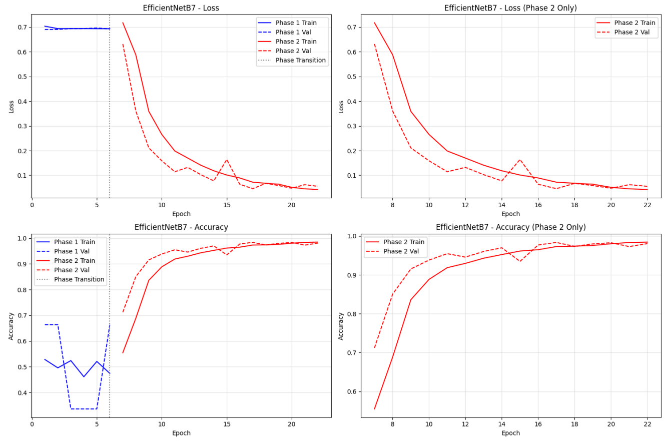

Fig 2. Training and validation curves of the EfficientNetB7 model across two-phase training on Celeb-DF-v2.

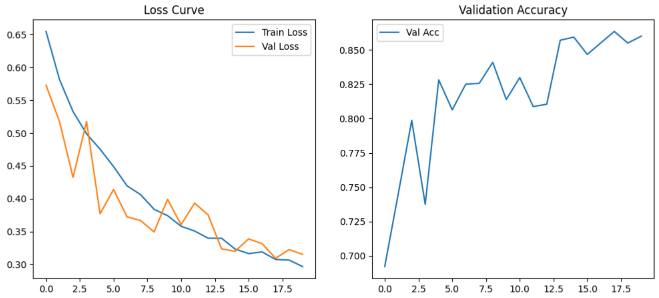

Fig 3. Training and validation curves of the EfficientNetB7 model training on FaceForensics++

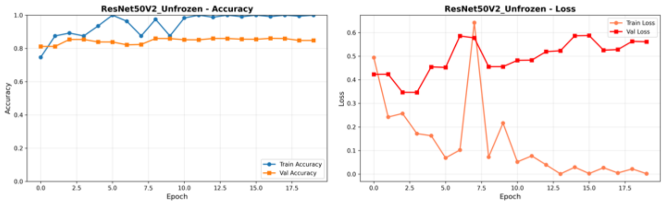

Fig 4. Training and validation curves of the ResNet50 model training on FaceForensics++

Once the training process was finished, the performance of each model was measured using standard evaluation metrics. The results are summarized in the following tables.

TABLE I
PERFORMANCE OF EFFICIENTNET AND RESNET

Model	Accuracy	Precision	Recall	F1-score
FaceForensic++				
EfficientNet	90.1%	83.5%	89.1%	86.2%
ResNet	81.98%	90.17%	54.1%	67.63%
Celeb-DF-v2				
EfficientNet	97.5%	99.6%	93.4%	96.4%
ResNet	93.5%	95.3%	85.9%	90.3%

On the FaceForensics++ dataset, EfficientNet demonstrates consistently strong performance, reaching 90.1% accuracy with well-balanced precision and recall. This indicates that the model effectively internalized the visual cues and manipulation artifacts characteristic of this benchmark. ResNet50, on the other hand, presents a more uneven profile: although its precision is high at 90.17%, its recall drops markedly, leading to a lower overall accuracy. In practical terms, this means ResNet50 behaves more conservatively flagging fewer fake samples, but doing so with greater confidence when it commits to a prediction.

On the Celeb-DF-v2 dataset, EfficientNet delivers a surprisingly strong performance, reaching 97.5% accuracy, with an exceptionally high precision of 99.6% and a recall of 93.4%. These numbers suggest that the model handles the dataset's mix of diverse video sources and realistic manipulation methods reasonably well, even if some fakes still manage to slip past. Its predictions aren't perfect, but the balance between catching manipulated frames and avoiding unnecessary false alarms is much more stable than one might expect given the dataset's complexity.

ResNet50 delivers strong and stable results, with 93% accuracy, 95% precision, and 86% recall, yielding an F1-score of about 90%. This performance reflects high precision with moderate recall, indicating few false positives but some missed positives. The consistency suggests that the two-phase fine-tuning aligns well with the residual architecture, enabling robust representations across Celeb-DF-v2. While not significantly outperforming EfficientNet, ResNet50 remains reliable and avoids major variability.

To further evaluate the models' generalization capability beyond the training and validation distributions, additional testing was conducted on completely unseen video data, sources that were never part of the FaceForensics++ or Celeb-DF-v2 datasets. For this evaluation, two test videos were selected: one authentic (real) video and one manipulated (fake) video. Frames were sampled at a rate of 10% from each test video, providing a representative but computationally efficient snapshot of the video content.The results are summarized in the following tables.

TABLE 2
PERFORMANCE OF EFFICIENTNET AND RESNET ON UNSEEN DATA

Model	Actual Label	Predicted Label	Confidence
EfficientNet-FF++	Real	Real	59.16%
	Fake	Real	85.11%
ResNet-FF++	Real	Fake	65.40%
	Fake	Fake	55.35%
EfficientNet-Celeb-DF-v2	Real	Fake	98.63%
	Fake	Real	66.44%
ResNet-Celeb-DF-v2	Real	Real	50.57%
	Fake	Real	54.22%

The results expose critical limitations in model robustness: two out of four models failed to correctly classify the real video, with confidence levels ranging from 50.57% to 98.63%, indicating systematic sensitivity to learned artifacts that leads to authentic content being flagged as manipulated

Models trained on FaceForensics++ showed contrasting behaviors. EfficientNet-FF++ correctly classified the real video (59.16%) but misclassified the fake video as real with 85.11% confidence, suggesting difficulty in detecting manipulation techniques outside its training domain. ResNet-FF++ misclassified the real video as fake (65.40%) but correctly identified the fake video with lower confidence (55.35%), reflecting uncertainty that may be preferable in deployment scenarios. These failures underscore severe domain generalization challenges in deepfake detection, likely driven by distribution mismatch, dataset bias, and over-reliance on low-level artifacts, resulting in unacceptable false positive rates that undermine real-world applicability in contexts such as journalism or legal proceedings.

Models trained on Celeb-DF-v2 performed even worse: EfficientNet misclassified the real video as fake with 98.63% confidence and predicted the fake video as real with 66.44%, while ResNet misclassified the fake video as real (54.22%) and showed minimal confidence (50.57%) for the real video. These outcomes contrast sharply with their strong in-distribution performance, highlighting that high benchmark accuracy does not guarantee robustness under domain shift.

IV. CONCLUSION

This study evaluated EfficientNet-B7 and ResNet50 for deepfake image detection using the FaceForensics++ and Celeb-DF-v2 datasets, with both models trained under a two-phase pipeline that froze the backbone first and then gradually fine-tuned the full network. The approach helped stabilize early training and allowed both architectures to adapt more smoothly to the characteristics of each dataset. On FaceForensics++, EfficientNet achieved strong and well-balanced performance (90.1% accuracy), indicating that it captured the dataset's fairly consistent manipulation patterns. ResNet50, meanwhile, showed high precision but struggled with recall, suggesting a more cautious prediction style that caused it to miss many fake samples.

On Celeb-DF-v2, EfficientNet achieved good results, recording 97.5% accuracy and high precision (99.6%), though its recall remained slightly weaker. ResNet50 followed closely, delivering 93.5% accuracy with balanced metrics, reflecting the effectiveness of the two-phase fine-tuning in conjunction with its residual architecture. However, these promising outcomes proved deceptive when the models were evaluated beyond their training domain. In tests on two completely unseen videos, three of the four trained models misclassified the real video as fake. Even the Celeb-DF-v2 models, which had performed nearly flawlessly on their benchmark, failed in the same manner, underscoring a fundamental limitation in generalization.

Together, these findings reinforce a central challenge in deepfake detection research: strong benchmark performance does not translate into trustworthy real-world behavior. Instead, the models appear to latch onto dataset-specific quirks rather than robust, generalizable cues of manipulation. Over-confidence on out-of-distribution samples further highlights the need for improved calibration, broader training diversity, and methods that explicitly address domain shift.

Several directions could further strengthen this work. Combining FaceForensics++ and Celeb-DF-v2 through multi-dataset or progressive training, along with domain-generalization techniques such as data augmentation that alters style, compression, or color distribution, could help models learn generalizable features rather than dataset-specific patterns. Incorporating temporal information from videos rather than relying exclusively on still images could also capture subtle motion irregularities that frame-based models miss. Another promising avenue is the fusion of additional modalities, such as audio cues or physiological consistency signals, which are more difficult to synthesize convincingly. Finally, future evaluations should consider recently developed high-quality deepfake methods and user-generated material from real platforms to assess how well the models handle truly unconstrained data. These steps would help move deepfake detection systems closer to reliable, wide-scale deployment.

References

[1] S. Raj, J. Mathew, and A. Mondal, "FDT: A python toolkit for fake image and video detection," SoftwareX, vol. 22, p. 101395, May 2023, doi: 10.1016/j.softx.2023.101395..

[2] Goodfellow et al., "Generative adversarial networks," Communications of the ACM, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[3] A. Kammoun, R. Slama, H. Tabia, T. Ouni, and M. Abid, "Generative Adversarial Networks for Face Generation: A survey," ACM Computing Surveys, vol. 55, no. 5, pp. 1–37, doi: 10.1145/3527850.

[4] "Deepware," Deepware.Ai | Scan & Detect Deepfake Videos. https://scanner.deepware.ai/.

[5] H. Ajder, G. Patrini, F. Cavalli, and L. Cullen, "The State of Deepfakes: Landscape, Threats, and Impact," Deeptrace, Deeptrace, pp. 1–6, Sep. 2019. doi: 10.1109/avss.2018.8639163.

[6] "Microsoft Mobile Phone Authenticator App | Microsoft Security." https://www.microsoft.com/en-in/security/mobile-authenticator-app.

[7] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1–11, Oct. 2019, doi: 10.1109/iccv.2019.00009.

[8] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: a Large-Scale challenging dataset for DeepFake forensics," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020, doi: 10.1109/cvpr42600.2020.00327.

[9] Tan, M., & Le, Q. V. (2019b). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1905.11946.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv (Cornell University), Jan. 2015, doi: 10.48550/arxiv.1512.03385.

[11] L. K. Yee, I. R. A. Hamid, C. ChaiWen, Z. Abdullah, K. Kipli, and C. F. M. Foozy, Deepfake Image Detection Using ResNet50 Model. IEEE Xplore, 2024, pp. 80–87. doi: 10.1109/cybercomp60759.2024.10913843.

[12] S. Borade et al., "ResNet50 DeepFake Detector: Unmasking Reality," Indian Journal of Science and Technology, vol. 17, no. 13, pp. 1263–1271, Mar. 2024, doi: 10.17485/ijst/v17i13.285.