SN Computer Science (2025) 6:598
https://doi.org/10.1007/s42979-025-04121-y

ORIGINAL RESEARCH

DeDER: Detecting Deepfakes with EfficientNetB0 and ResNet50 on Celeb-DF

Yashoda Chouhan[1] · Chetankumar Chudasama[1] · Deepak Kumar Verma[1]

Received: 30 January 2025 / Accepted: 7 June 2025

Abstract

The challenge of identifying the difference between genuine and altered images or videos is becoming increasingly more difficult due to advancements in deepfake technology. This research examines the application of two prominent deepfake detection models based on convolutional neural networks, EfficientNetB0 and ResNet50, using the Celeb-DF dataset. Both models utilize transfer learning, which employs pre-trained weights and requires training the model to detect minute details in the altered media. In addition to this, we propose an ensemble approach that combines predictions from multiple models to enhance detection performance through probability averaging. The results from the experiments show that EfficientNetB0 and ResNet50 have individual accuracies of 98.17% and 98.03% respectively, but the ensemble model achieved a more accurate 99.46%. This approach offers promising capabilities for immediate application in deepfake detection, particularly in the domains of forensic analysis of digital media, content safety verification systems, and cybersecurity. This research highlights the agility and effectiveness of modern model architectures in combating the rapidly evolving threat of synthetic media.

Keywords Deepfake detection · Resnet50 · EfficientNetB0 · Celeb-DF · Deep learning

Introduction

Deepfake technology is a powerful tool in the digital world. It can create AI-generated or manipulated content through images, videos, and audio. GNNs and DNNs are utilized to create deepfakes to a great extent, such as altering images and videos that appear real to the human eye. It includes changing facial expressions, altering speech, and videos [1, 2]. Although these technologies benefit multimedia production, entertainment, and other applications, they are also used in many harmful ways. Some are creating fake videos of public figures in situations they have never been in before, saying things they have never said. This can hinder political processes [3, 4]. One urgent concern leads to the deepfake creation of pornography, which is often created without consent, making society worry about the privacy and security of their rights [5, 6]. This, in turn, reduces people's trust in digital media [7]. Given the rapid improvement in deepfake generation techniques and growing concerns about malicious usage, the need for strong mechanisms in Deepfake detection cannot be more critical. The tools for this are complementary to guaranteeing the integrity of digital media and go hand in hand with protecting privacy and safety in the modern social world, which faces the increasing peril of manipulated content. Therefore, detection methods that can keep up with technological developments are needed. However, despite significant progress and active development in constructing models for deepfake generation, detecting such manipulations remains a considerable challenge. The difficulty primarily arises because manipulations included in the deepfakes are performed very subtly. The utilization of advanced models of AI allows deepfakes to become so realistic that, for instance, in videos, facial expression, and lighting conditions, together with other environmental factors, continuously change [8]. In such circumstances, the ability to detect will become poorer when distinguishing

Chetankumar Chudasama and Deepak Kumar Verma have contributed equally to this work.

✉ Chetankumar Chudasama
er.chetanchudasama@gmail.com

Yashoda Chouhan
yashoda2908@gmail.com

Deepak Kumar Verma
deepak.verma1980@gmail.com

[1] Computer Engineering, Marwadi University, Morbi Road, Rajkot, Gujarat 360002, India

Published online: 28 June 2025

between real and manipulated video or image content becomes practically impossible for the naked eye. Another complication arises due to the need to detect real-time deepfakes in social media, video conferences, and live streams. These applications require detection systems to analyze content in real-time, flagging manipulated media during dissemination rather than after it has occurred [9]. The need for real-time detection has become much more critical today, as deepfake technologies have become available, enabling malicious actors to create fake content on the fly. This research utilizes CNNs to extract features from images and videos for detecting minute traces created by deepfake technologies [10, 11]. Residual Networks, such as ResNet50, have shown awe-inspiring results in tasks that require the robust extraction of image features and have therefore been adapted with excellent efficiency for Deepfake detection tasks [12, 13]. Efficiency-oriented lightweight architectures like EfficientNetB0, which balance accuracy with computational cost, have also been considered for real-time deepfake detection [14, 15]. Despite these advancements, some gaps remain in the current research. Most current recognition systems work well in stable settings or with fixed data. Still, they struggle to maintain accuracy in changing environments, such as live-streaming platforms, where content is constantly updated. The performance of the models also reduces when new kinds of deepfakes are introduced. Datasets like Celeb-DF also have setbacks, as they provide a wide variation in lighting conditions, facial expressions, and backgrounds [16, 17]. Thus, there is a need to fine-tune the models, in addition to research on hyperparameter optimization, to enhance deepfake detection, especially in dynamic scenarios. We explored the potential of ensemble models and proceeded with the averaging method. Combining the predictions of both models enabled us to tackle complex deepfake detection effortlessly, resulting in a more robust model for real-world scenarios. This study aims to enhance the performance of EfficientNetB0 and ResNet50 models in detecting deep fakes, thereby addressing current challenges in this field. ResNet50 is known for its enhanced feature extraction capabilities, whereas EfficientNetB0 has an architecture that utilizes fewer computational resources, making it a promising choice for deepfake detection tasks. This study focuses on fine-tuning and applying hyperparameter techniques, such as dropout, batch size, and learning rate, to enhance model performance [18, 19]. We need to develop real-time detection algorithms to detect deepfakes before they can cause harm. This research, through finetuning models with selected hyperparameters, aims to make digital media more trustworthy and reduce the risks associated with malicious content manipulation. The results of this study have significant implications for digital forensics and cybersecurity, particularly in video verification, providing new tools to counter the growing threat of deepfakes across various fields. Using the ensemble model added resilience in detecting deepfakes and ensured better performance and results when evaluated. As deepfake technology continues to improve, the conclusions of this work will be crucial for securing personal privacy, national security, and public trust. This research is organized as follows: Section “Literature Review” presents a literature review of previous work, Section “Methodology” discusses the methodology, and Section “Experimental Setup” presents the experimental setup. Section “Results” presents the results, while the discussions and conclusions are provided in Sections “Discussions” and “Conclusion”, respectively.

Literature Review

The advent of deepfake technology has created increasing concern over the veracity of digital content. Deepfakes, primarily used for artistic and entertainment purposes, have also posed a serious threat to security, privacy, and trust. As such, detecting deepfakes in images and videos is an important area of research. This study examines past work on deepfake identification. It focuses on improvements in model designs, the Celeb-DF dataset, methods for finetuning, and the issues researchers face. Deepfake discovery typically relies on identifying marks and errors that are added during the editing process. Such artifacts are not necessarily perceptible to the human eye but may be determined by machine learning methods. The initial works concentrated mainly on Convolutional Neural Networks (CNNs) for anomaly detection in facial expressions, blink patterns, and other minute cues. Chesney and Citron [1] discussed how GANs generate realistic deepfake material and emphasized the urgency to create a reliable deepfake detection method. He et al. [6] worked with ResNet50, a deep neural network with 50 convolutional layers. This model can learn to effectively and correctly identify deepfakes, particularly in terms of face features and textures. Tan and Le proposed the EfficientNetB0 model [7], which represented a significant leap toward models that effectively serve the purpose. It increases model depth, width, and precision in a manner that is compatible with most CNN models, which typically utilize fewer parameters. This makes it suitable for real-time detection. A variety of datasets is essential to enhance the performance of deepfake detection models. One of the significant contributions to the dataset was made by Li et al. [16], who released the Celeb-DF dataset. It has over 5000 deepfake videos, a substantial resource for practical training. This dataset encompasses a variety of backgrounds, lighting conditions, and facial orientations in the videos. He discussed how Celeb-DF is more challenging compared to earlier datasets, such as FaceForensics++ created by Rossler et al. [3], as it focuses on uneven lighting and mismatch in facial areas, which more straightforward datasets can overlook. Nguyen et al. [4] demonstrated that enhancing pre-trained models is an essential task for creating an effective deepfake detection method, and that finetuning the last few layers can help reduce computational cost. To further improve their performance, it was trained on the Celeb-DF dataset. The image data augmentation techniques, such as rotation and flipping, were investigated by Zhao et al. [10] and have been proven to increase model performance. According to Wang et al. [9]., the progress of generative models necessitates more sophisticated methods of detection. Moreover, skewed datasets pose another problem: certain deepfakes are disproportionately represented in public datasets, which can negatively impact the performance of models.

He mentioned that generative models, such as GANs, which create deepfakes, are becoming increasingly advanced, necessitating the need to update existing deepfake detection methods. One of the issues faced is data imbalance. We have an abundance of certain types of deepfakes as compared to others. It results in excelling in a specific kind of deepfake detection and outperforming the others. Wang et al. [20] discussed the importance of a balanced dataset, which would help detect all manipulation methods. Recent models perform poorly in utilizing different synthesized deepfake methods. Moreover, dataset imbalance and a lack of diversity in the training data further complicate model training, leading to biases towards specific types of manipulation.

Methodology

This section outlines the methodology employed to design, train, and evaluate deepfake detection models using ResNet-50 and EfficientNet-B0. We explain the models used, the dataset chosen, the preprocessing steps, the training setup, and evaluation metrics, ensuring a clear and comprehensive understanding of the process followed in this research.

Models Used

This study employed two advanced CNN models: ResNet-50 and EfficientNet-B0. These systems have become popular in image classification because they accurately extract complex features from pictures. A pre-training process on large datasets, such as ImageNet, enables us to fine-tune both models for deepfake detection. Transfer learning enables our models to identify subtle flaws in altered videos. The combination of flexibility and speed makes ResNet50 and EfficientNetB0 excellent tools for this task.

Resnet50

ResNet50 is a deep learning model that utilizes convolutional layers. It was introduced in 2015 by Kaiming He et al. [6] and is known for its practical training of deep models. ResNet-50 is built around the concept of residual blocks, where the output of a block is the sum of the learned transformation $F(x)$ and the input x. This can be expressed as:

$$y = F(x) + x$$

This residual connection enables the network to learn residual mappings, thereby alleviating the vanishing gradient problem in deep networks. ResNet-50 specifically uses a bottleneck architecture in its residual blocks, where three convolutional layers—two 1×1 and one 3×3 convolution—are stacked to reduce and then restore the feature dimensions. This is represented as:

$$y = W_3 \cdot \mathrm{ReLU}\left(W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot x + b_1\right) + b_2\right) + b_3$$

This approach ensures fewer parameters and better computational efficiency. The network also incorporates identity mapping, ensuring that the input x is directly passed to the output alongside the learned residual, maintaining adequate gradient flow during backpropagation. After passing through multiple residual blocks, ResNet-50 applies a global average pooling operation, averaging the feature map to produce a vector. The formula for global average pooling is:

$$\mathrm{GlobalAvgPool}(x) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ij}$$

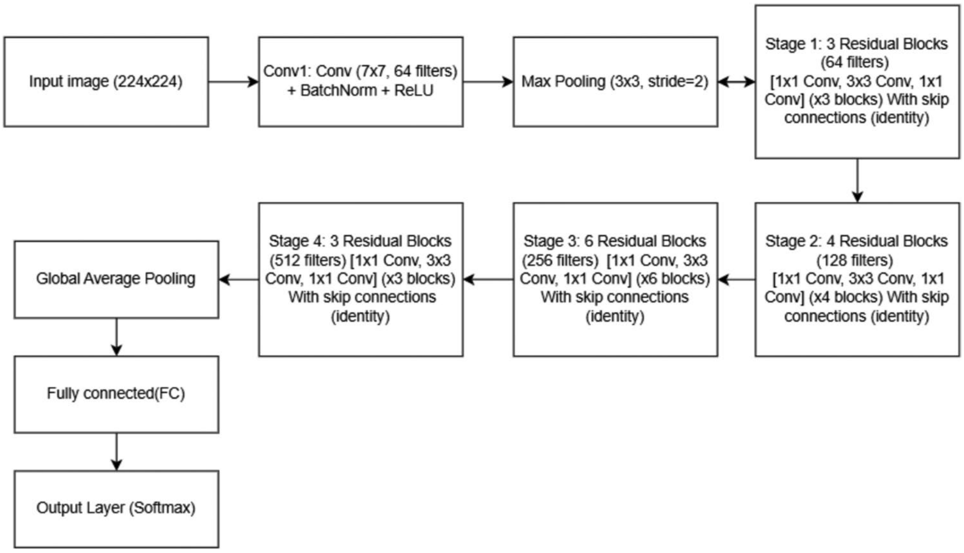
where x_{ij} are the feature map elements with dimensions $H \times W$. A fully connected layer is applied to produce the final output after pooling, represented as:

$$\hat{y} = W_{fc} \cdot \mathrm{GlobalAvgPool}(x) + b_{fc}$$

This makes ResNet-50 highly efficient at training deep networks while maintaining high performance in image classification.

This method solved training issues such as the disappearing slopes and helps the model learn more complex patterns. ResNet50 has 50 layers, which makes sure it is capable of working correctly as shown in Fig. 1. The model's ability to differentiate images demonstrates its capacity to handle challenging datasets. We chose ResNet50 for our deepfake detection task because it can capture complex, layered picture features. Initially, we utilized the pre-trained ResNet-50 model, which was trained on the ImageNet dataset. This allowed the model to learn many general features before we tweaked it to spot fake deepfake videos.

Fig. 1 Basic Resnet50 architecture


Input image (224x224)
Conv1: Conv (7x7, 64 filters) + BatchNorm + ReLU
Max Pooling (3x3, stride=2)
Stage 1: 3 Residual Blocks (64 filters) [1x1 Conv, 3x3 Conv, 1x1 Conv] (x3 blocks) With skip connections (identity)
Stage 2: 4 Residual Blocks (128 filters) [1x1 Conv, 3x3 Conv, 1x1 Conv] (x4 blocks) With skip connections (identity)
Stage 3: 6 Residual Blocks (256 filters) [1x1 Conv, 3x3 Conv, 1x1 Conv] (x6 blocks) With skip connections (identity)
Stage 4: 3 Residual Blocks (512 filters) [1x1 Conv, 3x3 Conv, 1x1 Conv] (x3 blocks) With skip connections (identity)
Global Average Pooling
Fully connected(FC)
Output Layer (Softmax)

EfficientNetB0

Mingxing Tan and Quoc V. Le [7] introduced EfficientNet-B0. EfficientNet-B0 employs a novel compound scaling method to balance network depth, width, and resolution, enabling the model to achieve higher performance while maintaining efficiency. The scaling process follows this formula:

$$\text{New Size} = \alpha \cdot \text{Current Size}$$

Where α is a scaling factor applied to depth, width, or resolution. Unlike older techniques, which scale these factors, Compound Scaling discovers the best mix for top performance. The model's architecture is based on the MobileNetV2 inverted residual block, which uses depthwise separable convolutions. The block structure can be represented as:

$$y = \text{DepthwiseConv}(x) \cdot W_1 + b_1$$

where x is the input feature map, W_1 is the weight matrix for depthwise convolution, and b_1 is the bias. The depthwise convolution is followed by a pointwise convolution to map the outputs to the next layer:

$$z = \text{PointwiseConv}(y) \cdot W_2 + b_2$$

The compound scaling method is applied in EfficientNet-B0 by uniformly scaling the network dimensions (depth, width, and resolution). The scaling strategy for each factor is shown in a mathematical form below:

$$\text{Depth Scale} = \alpha^{\text{depth\_factor}}$$

$$\text{Width Scale} = \alpha^{\text{width\_factor}}$$

$$\text{Resolution Scale} = \alpha^{\text{resolution\_factor}}$$

This approach enhances its capacity for performance and efficiency. EfficientNet-B0 incorporates a block called squeeze-and-excitation, which enhances its representational capacity. It works by first squeezing the input feature map to a channel descriptor:

$$z = \text{GlobalAvgPool}(x)$$

Then, it applies a fully connected two-layer network to capture channel-wise dependencies:

$$\hat{z} = \sigma\left(W_1 \cdot z + b_1\right)$$

At last, the attention map $\hat{z}$ is used to rescale the feature map:

$$x' = \hat{z} \cdot x$$

EfficientNet-B0 is known to perform better than some of the older models, such as ResNet-50 and Inception-v4, on the ImageNet dataset. It achieves this by using fewer parameters and requiring less computing power. This improvement is achieved through the use of depthwise separable convolutions and the Swish activation function. These features make the model simpler without hurting its learning ability.

The EfficientNetB0 architecture is shown in Fig. 2. EfficientNet-B0 performs well in applications that require quick responses and have limited resources, such as mobile phones or edge computing setups.

Ensemble Method

We developed an ensemble method that utilizes ResNet50 and EfficientNetB0. First, both networks were used as input on the image: $I = 224 \times 224 \times 3$. The output of each is a feature extraction for both networks' convolutional layers. It is then passed through a Global Average Pooling layer. This layer

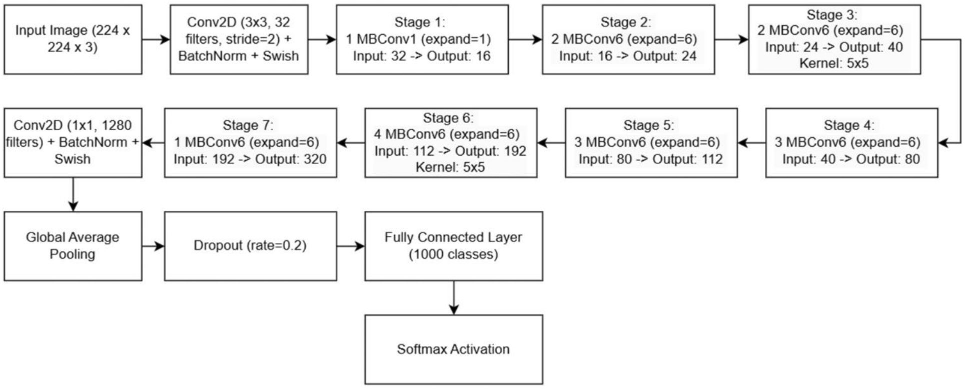

Input Image (224 x 224 x 3)
Conv2D (3x3, 32 filters, stride=2) + BatchNorm + Swish
Stage 1: 1 MBConv1 (expand=1) Input: 32 -> Output: 16
Stage 2: 2 MBConv6 (expand=6) Input: 16 -> Output: 24
Stage 3: 2 MBConv6 (expand=6) Input: 24 -> Output: 40 Kernel: 5x5
Stage 4: 3 MBConv6 (expand=6) Input: 40 -> Output: 80
Stage 5: 3 MBConv6 (expand=6) Input: 80 -> Output: 112
Stage 6: 4 MBConv6 (expand=6) Input: 112 -> Output: 192 Kernel: 5x5
Stage 7: 1 MBConv6 (expand=6) Input: 192 -> Output: 320
Conv2D (1x1, 1280 filters) + BatchNorm + Swish
Global Average Pooling
Dropout (rate=0.2)
Fully Connected Layer (1000 classes)
Softmax Activation

Fig. 2 Basic EffcientNetB0 architecture.

reduces spatial dimensions, and for ResNet50, it gives a 1D vector V_{resnet} while for EfficientNetB0, it gives $V_{efficientnet}$, represented as:

$$V_{resnet} = \mathrm{GAP}\left(F_{resnet}\right)$$

$$V_{efficientnet} = \mathrm{GAP}\left(F_{efficientnet}\right)$$

Where F_{resnet} and $F_{efficientnet}$ are the feature maps produced by both models.

The 1D vectors V_{resnet} and $V_{efficientnet}$ are then passed through a fully connected Dense layer (128), followed by a Dropout layer(0.3) to prevent overfitting. This results in the activations:

$$A_{resnet} = \mathrm{Dense}\left(V_{resnet}\right)$$

$$A_{efficientnet} = \mathrm{Dense}\left(V_{efficientnet}\right)$$

Dropout regularization prevents the model from over-relying on specific features during training. Next, each dense layer's output is passed through a Sigmoid activation function, which maps the activations to a probability between 0 and 1:

$$P_{resnet} = \sigma\left(A_{resnet}\right) = \frac{1}{1 + e^{-A_{resnet}}}$$

$$P_{efficientnet} = \sigma\left(A_{efficientnet}\right) = \frac{1}{1 + e^{-A_{efficientnet}}}$$

Where σ is the Sigmoid function. The final step includes the averaging of the probabilities from both models to create an ensemble prediction:

$$P_{ensemble} = \frac{P_{resnet} + P_{efficientnet}}{2}$$

This averaged probability is then passed through a final Sigmoid activation to produce the final output probability:

$$P_{final} = \sigma\left(P_{ensemble}\right) = \frac{1}{1 + e^{-P_{ensemble}}}$$

Lastly, the image is classified as fake if $P_{final} \geq 0.5$ and as real if $P_{final} < 0.5$. Mathematically, the final classification is:

$$\text{Prediction} = \begin{cases} \text{Fake, if } P_{final} \geq 0.5 \\ \text{Real, if } P_{final} < 0.5 \end{cases}$$

The basic architecture of the ensemble model is shown in Fig. 3. This ensemble approach leverages the strengths of ResNet50 and EfficientNetB0, thereby enhancing the model's overall performance by combining their predictions.

Celeb-DF Dataset

In this research, we leveraged the Celeb-DF dataset. It is a collection of 590 real videos and 5639 manipulated videos, comprising over 2 million frames combined. It is beneficial for deepfake detection tasks, as it incorporates various features, including background, lighting, and face texture. This makes it a challenging dataset. It can help train deepfake detection models that perform well in real-world scenarios.

It contains high-quality videos that capture small details in every frame, as shown in Fig. 4. This matters because fake videos often exhibit subtle visual differences, such as unusual facial movements or odd lighting. Celeb-DF video content made this research perfect. It offered range and complexity, allowing for the training of models that could work well in various scenarios and situations.

Preprocessing

Data preparation plays a vital role in detecting deepfakes. It helps make the data uniform and suitable for training. In this research, we extracted a total of 50,000 frames and split them into training, test, and validation folders with a 7:2:1 ratio, as shown in Table 1.

The steps to prepare the data included:

Resizing: Every video frame was set to the exact size of 224 × 224 pixels. This size is standard for many AI learning models, such as ResNet50 and EfficientNetB0, ensuring that all pictures fed to the model have the exact measurements.

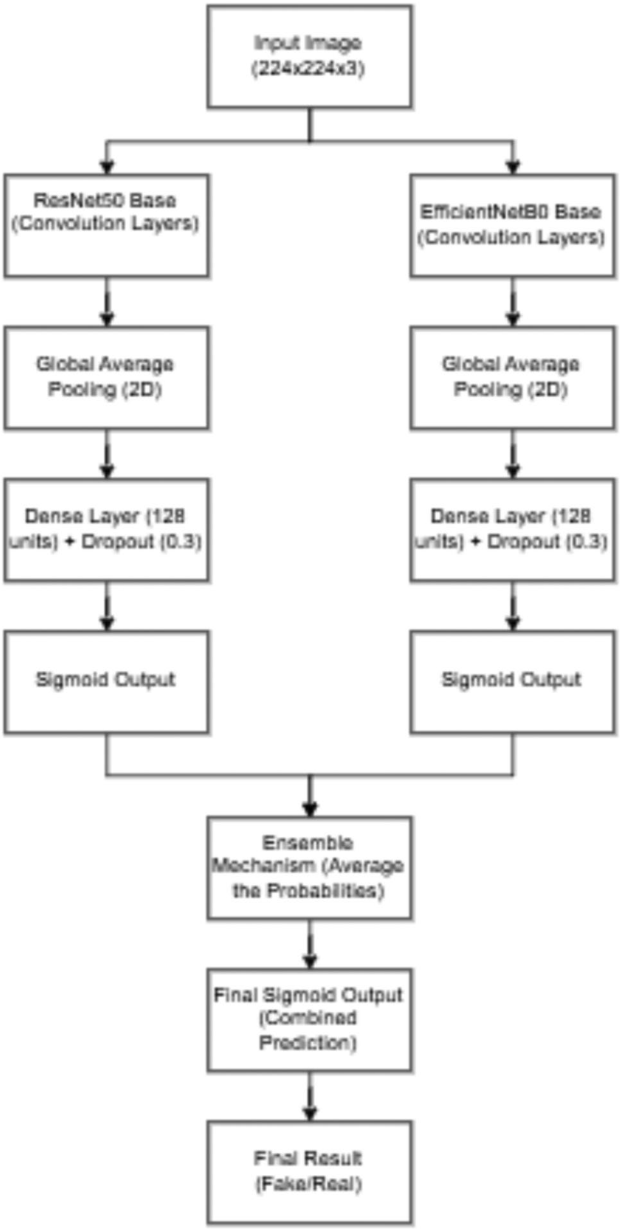

Input Image (224x224x3)
ResNet50 Base (Convolution Layers)
EfficientNetB0 Base (Convolution Layers)
Global Average Pooling (2D)
Global Average Pooling (2D)
Dense Layer (128 units) + Dropout (0.3)
Dense Layer (128 units) + Dropout (0.3)
Sigmoid Output
Sigmoid Output
Ensemble Mechanism (Average the Probabilities)
Final Sigmoid Output (Combined Prediction)
Final Result (Fake/Real)

Fig. 3 Basic Ensemble model architecture.

$$I' = \text{Resize}\left(I, H', W'\right)$$

Where I is the original image, and H' and W' are the target height and width.

Normalization: The pixel values in each image were set to fall between 0 and 1. This helps the model learn faster during training by evening out the input data. The formula is:

$$I_{\text{norm}}(x, y) = \frac{I(x, y)}{255}$$

Where $I(x, y)$ is the original pixel value, and $I_{\text{norm}}(x, y)$ is the normalized pixel value.

Augmentation: We employed data augmentation techniques to enhance model generalization and mitigate overfitting. We rotated the images randomly, flipped them horizontally, and adjusted their brightness slightly. These changes help mimic real-life situations and add more variety to our training data. The combined augmentation can be represented as:

$$I_{\text{aug}} = \mathcal{A}(I) = f_{\text{bright}}\left(f_{\text{flip}}\left(f_{\text{rot}}(I)\right)\right)$$

Where:

I is the original image,

$f_{\text{rot}}(I)$ represents a random rotation applied to I,

$f_{\text{flip}}(I)$ represents a random flip applied to $f_{\text{rot}}(I)$,

$f_{\text{bright}}(I)$ represents a random brightness adjustment applied to $f_{\text{flip}}(f_{\text{rot}}(I))$,

I_{aug} is the augmented image.

θ is a randomly chosen rotation angle.

and f is a random brightness factor.

These preparation steps let our models learn from a broader range of data and handle real-world hurdles like changes in light, how faces are turned, and different backgrounds.

Training Setup

The models underwent two primary phases of training: pre-training and finetuning. We initialized ResNet50 and EfficientNet-B0 with pre-trained weights from ImageNet to equip them with strong general feature knowledge. This significantly accelerated the training process from scratch, as the models now learned deepfake-specific patterns and were free from noise. Following the initial pre-training, we used the Celeb-DF dataset to finetune the models. Since the videos are classified as real or fake, binary cross-entropy loss was used to train the models. Early stopping ensured that the models would cease training as soon as the validation accuracy stopped increasing, preventing overfitting. Adam's optimizer was used to finetune with a learning rate of 1e–5.

Experimental Setup

Here, we describe the training, validation, and testing protocol in our experiments. We also include hyperparameter tuning, control factors used during model training, and additional information regarding the environment used for training and testing under different conditions. In both models, the data was divided into three divisions: 70% for training, 20% for validation, and 10% for testing. The training set was used to fit the models, the validation set was used to finetune

Fig. 4 Snippet of Celeb-DF dataset

Table 1 Dataset Split shown in frames

	Train	Test	Val
Real	17,500	5000	2500
Fake	17,500	5000	2500

Table 3 Performance of Resnet50 in metrics(in %)

Accuracy	Precision	Recall	F1 Score	AUC ROC Score
98.03	98.44	96.6	98.02	99.85

Table 2 Summary of experimental setup

Component	Details
Hardware	Google Colab L4 GPU
Framework	TensorFlow (latest), Python 3.x
Batch Size	32
Learning Rate	1e–5
Optimizer	Adam
Epochs	30
Early Stopping	Enabled (based on validation loss)
Input Size	224 × 224 pixels
Data Split	Train: 70%, Validation: 20%, Test: 10%

the hyperparameters and assess the model's performance during training, and the test set was held out to evaluate the final model's performance. A fair comparison could be made with fixed data splits, allowing all models to train and test on the same data distribution.

The essential aspects of the training and testing setup are outlined in Table 2. This configuration was uniformly applied for both ResNet50 and EfficientNetB0 for consistent, fair comparison and reproducibility.

Results

The studies were designed to ensure a fair comparison, so both models were trained and evaluated under similar conditions. The output metrics provide us with several performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, which enable us to analyze the performance of each model in accurately identifying real videos as real and fake videos as fake.

ResNet50 Performance

ResNet50 attained an accuracy of 98.03% with 98.44% precision, 96.6% recall, F1-score of 98.02%, and ROC-AUC of 99.85% as presented in Table 3, marking it as proficient in differentiating real videos from fake ones.

The confusion matrix, along with the training and validation graphs for accuracy and loss achieved by the Resnet50 model, is shown in Figs. 5 and 6, respectively.

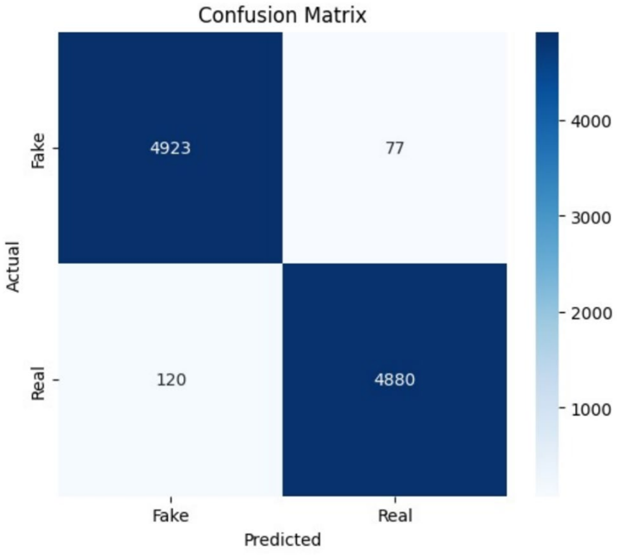


Fig. 5 Confusion Matrix of Resnet50

EfficientNetB0 Performance

The EfficientNetB0 model showed similarly impressive results, with an overall 98.17% accuracy, as shown in Table 4, which is very close to ResNet50's performance.

With EfficientNetB0, precision improved minimally to 98.52%, while recall remained at 97.8%. Moreover, the cumulative performance metric—F1-score of 98.16% and ROC-AUC of 99.82, were similar to those of ResNet50.

Like ResNet50, the confusion matrix for EfficientNetB0 indicated that the model performed well in distinguishing between real and fake content. However, it did have a slightly higher number of false negatives than ResNet50, as demonstrated in Fig. 7, the training and validation graphs for accuracy and loss achieved are shown in Fig. 8. Nevertheless, EfficientNetB0 remains a highly effective and reliable model for detecting deepfakes, exhibiting robust overall performance.

Table 4 Performance of EffficientNetB0 in metrics(in %)

Accuracy	Precision	Recall	F1 Score	AUC ROC Score
98.17	98.52	97.8	98.16	99.82

Runtime Performance Results

The runtime results for EfficientNetB0 and ResNet50 are shown in Table 5.

Ensemble Method Performance

As outlined in Table 6, the ensemble model surpassed the two individual models, achieving 99.46% in accuracy, 99.47% in precision, 99.44% in recall, 99.45% F1-score, and ROC-AUC of 99.98%.

The confusion matrix, validation loss, and accuracy graph are displayed in Figs. 9 and 10, respectively.

Lastly, the ensemble method proved successful, yielding a reliable model for detecting deepfakes. The combination of the models yielded excellent performance, demonstrating a promising solution for detecting deepfakes in real-world scenarios.

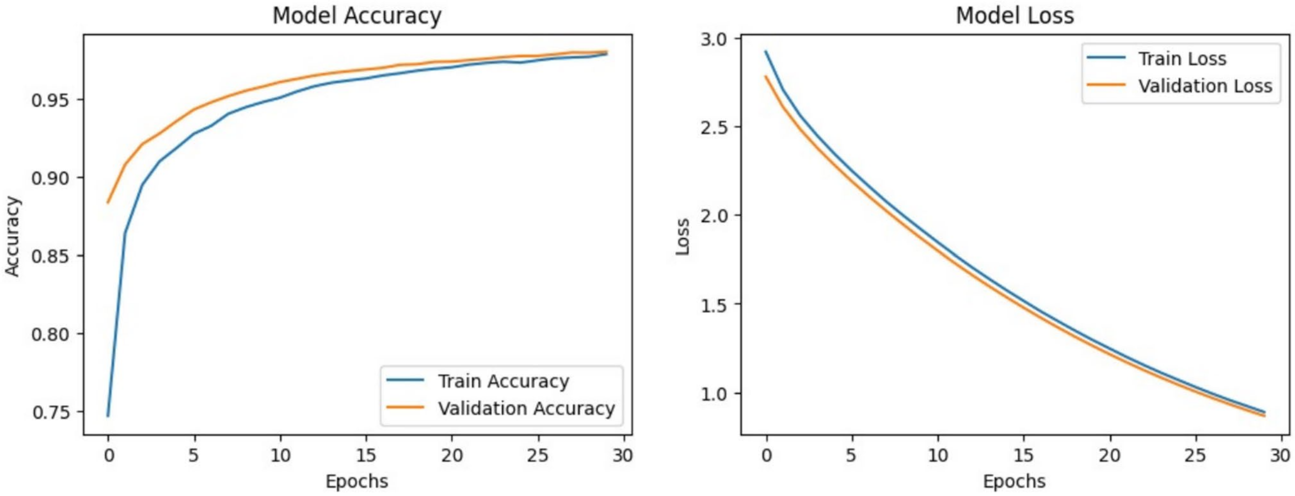


Fig. 6 Training and validation graphs for accuracy and loss of Resnet50

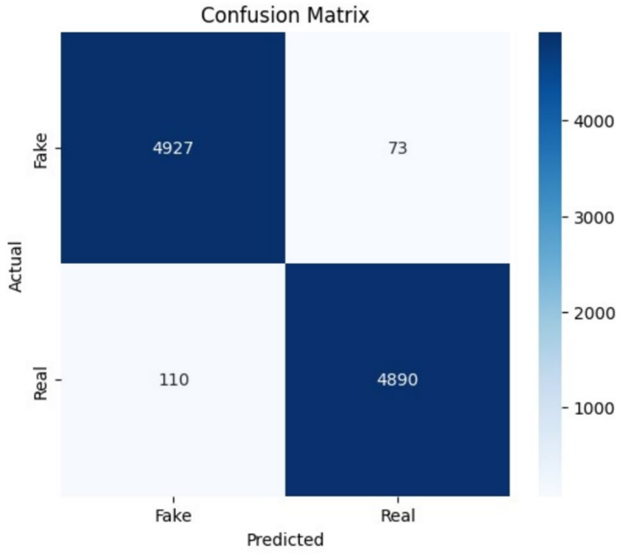


Fig. 7 Confusion Matrix of EffciientNetB0

Comparison with Other Research Findings

Various research findings were studied and compared with the results received in this research, as shown in Table 7.

Discussions

The performance of ResNet50 and EfficientNetB0 in a deepfake detection task, as measured by this experiment, will be valuable in constructing all future models. Both models demonstrated their strong abilities to approach the task from the viewpoint of distinguishing real and fake videos, but each model had its pros and cons regarding deepfake detection. To advance the research, we proposed an ensemble method focused on averaging. This method combines the strengths of both models to improve results in deepfake detection. ResNet50 demonstrated high accuracy and precision, indicating that it is effective in detecting deepfakes when sufficient distortion or artifacts occur during the modification process. However, while it excelled in precision and accurately identified true positives (correctly labeling fake videos as fake), its recall was somewhat lower. Sometimes, Resnet50 had trouble recognizing deepfakes, especially when the changes in the deepfakes were more minor and difficult to detect.

On the other hand, EfficientNetB0 was not as accurate, which means it often identified deepfakes, even when the differences were subtle. Higher recall means fewer deepfakes will be missed in the real world, where the fakes can be very realistic. This, however, came at the cost of some precision; in other words, EfficientNetB0 was more likely

Table 5 Runtime performance results

Model	Inference Time/ Frame (ms)	FPS	Model Size (MB)
EfficientNetB0	28.4	35.2	20.4
ResNet50	51.3	19.5	98.0

Table 6 Performance of Ensemble model in metrics(in %)

Accuracy	Precision	Recall	F1 Score	AUC ROC Score
99.46	99.47	99.44	99.45	99.98

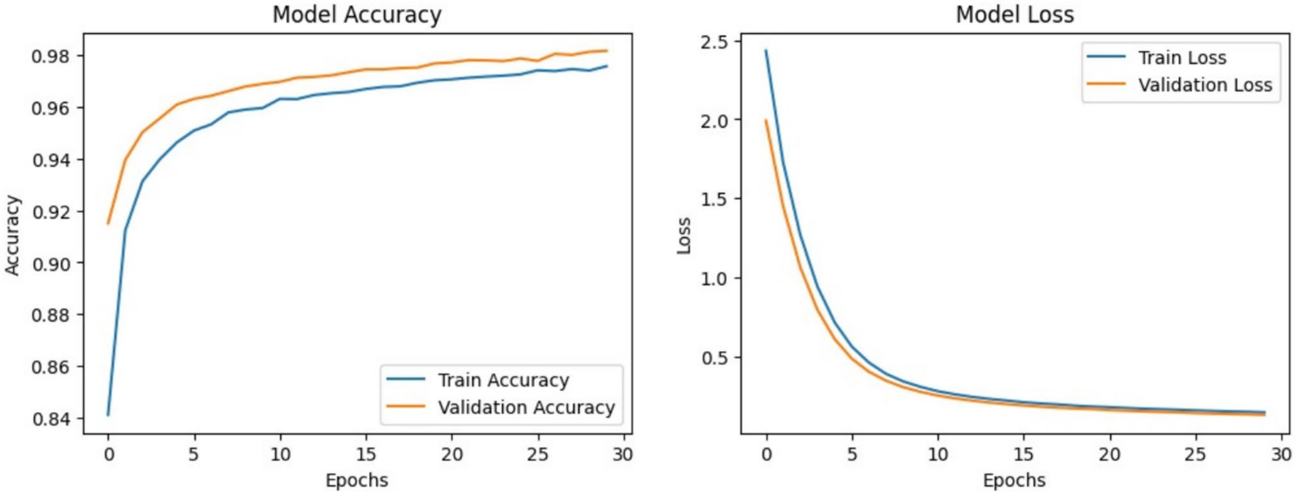


Fig. 8 Training and validation graphs for accuracy and loss of EffcientNetB0

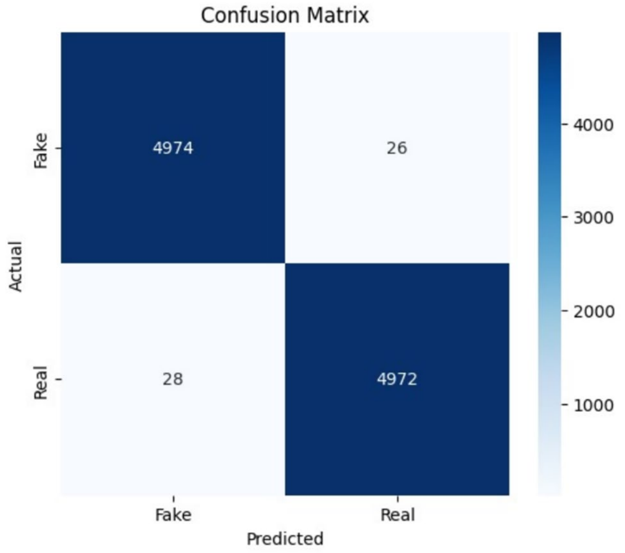


Fig. 9 Confusion matrix of ensemble method.

to misclassify real videos as fake, resulting in a higher number of false positives. Despite this, its higher recall and F1 score, an important metric that balances precision and recall, indicate that EfficientNetB0 is more balanced when correctly identifying both real and fake content, especially in cases where deepfake detection is critical. The ensemble model showed exceptional improvement in performance as compared to Resnet50 and EfficientNetB0, achieving a high score in the evaluated metrics. Both models produced solid ROC AUC scores, which measure the accuracy of their predictions of the video (in terms of whether it is likely to be fake or actual). In particular, EfficientNetB0 is a more effective model for detecting a wider range of deepfake videos, albeit at the expense of some accuracy. Overall, ResNet50 achieved a greater level of accuracy and precision, performing marginally better.

However, EfficientNetB0 outperformed it in terms of recall, making it the better model if deepfakes need to be detected first and foremost. The ensemble model's high recall and low false-negative rate make it highly useful in real-world scenarios where detecting deepfakes is crucial. In future research, we could focus on optimizing the ensemble approach by introducing different models and datasets to boost performance. Our benchmarks corroborate the assertion that ResNet50 fails to meet real-time performance requirements while EfficientNetB0 surpasses the mark at over 35 FPS on a standard GPU.

Conclusion

We utilized two models, ResNet50 and EfficientNetB0, and trained them on the Celeb-DF dataset. We achieved an outstanding accuracy of 98.03% and 98.17%, respectively. To achieve this, we finetuned pre-trained models on a dataset comprising diverse deepfake videos from the real world, resulting in models that could convincingly distinguish between real and fake content. This work demonstrates the promise of deep learning for practical applications,

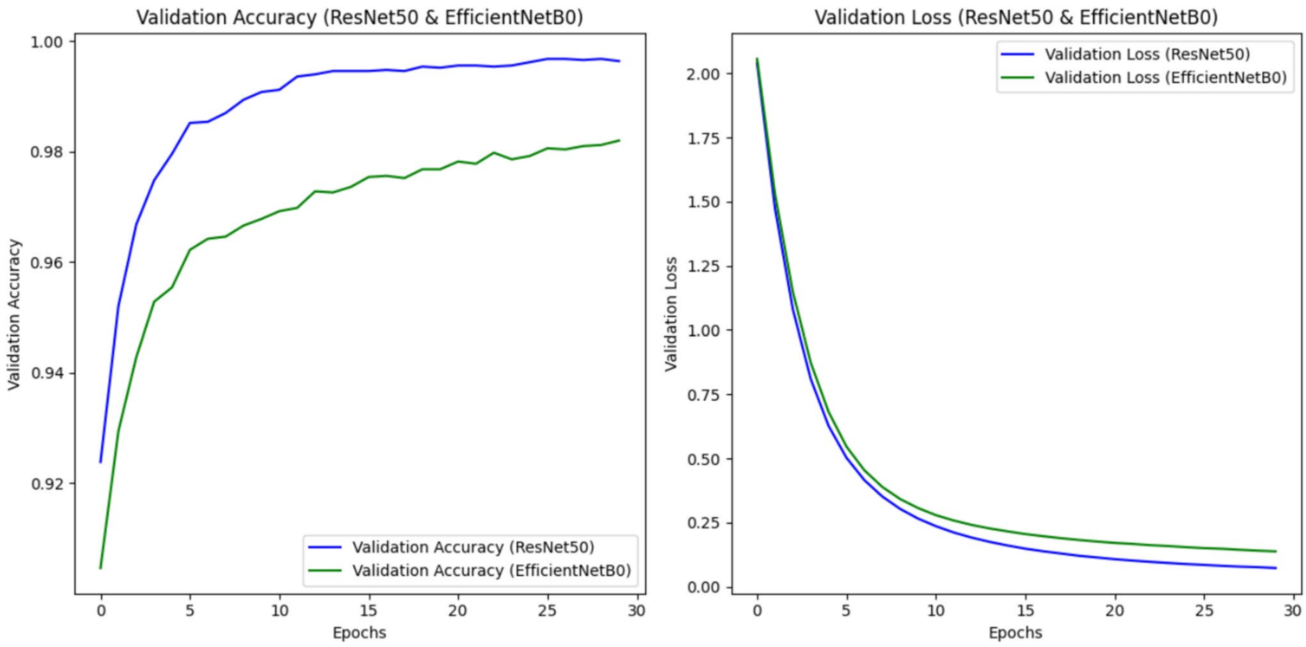


Fig. 10 Training and validation graphs for accuracy and loss of the Ensemble method

Table 7 Results compared with other findings

	Reference	Accuracy(%)
	Rossler et al. [3], 2019	96.32
	Yan et al. [21], 2023	97.12
	Radford et al. [22], 2021	96.98
	Zhang et al. [23], 2024	79.5
	Soares et al. [24], 2022	88.4
	Aghasanli et al. [25], 2023	89.2
	Bouter et al. [26], 2023	92.5
	Wang and Chow [27], 2023	70.1
	Zheng et al. [28], 2021	86.9
	Raza et al. [29], 2023	92.9
	Lopez Pellcier et al. [30], 2021	95.1
Ours	Resnet50	98.03
	EfficientNetB0	98.17
	Ensemble model	99.46

particularly in media verification, security, and privacy protection. After reviewing the solo performance of the two models, we explored the ensemble approach by averaging their predictions. The ensemble model achieved an accuracy of 99.46% and performed well across all metrics. This method allowed us to leverage the strengths of both models to increase performance. The future holds many interesting research directions. Incorporating speech recognition would enable the detection of inconsistencies at both visual and auditory levels of fake footage. Another area for development is real-time deepfake detection, which is crucial for use cases such as live streaming and social media platforms. To prevent the spread of harmful deepfake content, we may enable immediate detection and reporting.

Furthermore, the need for models with adversarial robustness is becoming increasingly crucial as deepfake generation methods continue to evolve. Adversarial training can help prepare models for new manipulation techniques, making them more adaptable and relevant in the long term. Optimization of models for computational efficiency remains crucial for real-time or resource-constrained applications. The fact that it provides both good performance and low computational cost will enable deepfake detection systems to become practical in large-scale, real-time applications. Ultimately, the advancement of deepfake detection technology will lead to a more reliable and secure digital landscape, enabling users to engage with the online world with greater confidence.

Author Contributions Author 1 conceived and designed the study, performed the experiments, and analyzed the data. Authors 2 and 3 supervised the research, guided the study, and critically reviewed the manuscript. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work.

Data Availability The manuscript utilizes no primary data. All analyses are based on existing literature and publicly available information.

Declarations

Conflict of Interest The authors have no competing interests to declare relevant to this article's content.

Clinical Trial No clinical trials were conducted as part of this study.

References

1. Chesney DK, Citron D. Deepfakes: a looming challenge for privacy, democracy, and national security. Calif Law Rev. 2019;107(5):1753–806.
2. Dolhansky B, Binns KA, Farahani R. The deepfake detection challenge. In: Proceedings of the IEEE/CVF International Conference on computer vision (ICCV), 2020; pp. 1–9.
3. Rossler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M. Faceforensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on computer vision, 2019; pp. 1–11
4. Nguyen TT, Nguyen QVH, Nguyen DT, Nguyen DT, Huynh-The T, Nahavandi S, Nguyen TT, Pham Q-V, Nguyen CM. Deep learning for deepfakes creation and detection: a survey. Comput Vis Image Underst. 2022;223: 103525.
5. Li Y, Yang X, Sun P, Qi H, Lyu S. Celeb-df: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2020; pp. 3207–3216.
6. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 2016; pp. 770–778.
7. Tan M, Le Q. Efficientnet: rethinking model scaling for convolutional neural networks. In: International Conference on machine learning, 2019; pp. 6105–6114 . PMLR.
8. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 2017; pp. 1492–1500.
9. Yu P, Xia Z, Fei J, Lu Y. A survey on deepfake video detection. Iet Biomet. 2021;10(6):607–24.
10. Wang J, Wu Z, Ouyang W, Han X, Chen J, Jiang Y-G, Li S-N. M2tr: multi-modal multi-scale transformers for deepfake detection. In: Proceedings of the 2022 International Conference on multimedia retrieval, 2022; pp. 615–623.
11. Dolhansky B, Farahani R, Binns KA. Real-time deepfake detection and countermeasures. IEEE Trans Multimed. 2020;22(5):1290–300.
12. Korshunov M, Marcel S. Deepfake video detection: a survey. Multimedia Tools Appl. 2020;79(9):6613–33.
13. Yang A, He Y, Lu S. A survey on deepfake detection: from classification to generative models. IEEE Trans Inf Forensics Secur. 2020;15:1571–86.
14. Zhang K, Li Z, Zhang Z. Detecting deepfakes with convolutional neural networks. J Mach Learn Res. 2020;21(72):1–22.
15. Xie L, Tian H, Zhao Y. Deepfake video detection using multimodal networks. IEEE Trans Circuits Syst Video Technol. 2020;30(4):1116–27.

16. Li P, Hu Y, Xu J. Fake or real? deepfake detection using advanced deep learning architectures. J Vis Commun Image Represent. 2020;68: 102799.
17. Zhu X, Yang H, Li L. Real-time fake video detection using deep learning. J Inform Secur Appl. 2020;55: 102567.
18. Kaur H, Acharjya DP. Detecting deepfake videos using convolutional neural networks. Comput Mater Continua. 2021;66(3):2381–94.
19. Kim TH, Lee YK, Choi JS. A comprehensive review on deepfake detection approaches: challenges and future directions. Comput Secur. 2020;92: 101765.
20. Wang Y, Zhang Z, Yu H. Deepfake video detection with adversarial training. IEEE Trans Image Process. 2020;29:3045–57.
21. Yan Z, Zhang Y, Fan Y, Wu B. Ucf: uncovering common features for generalizable deepfake detection. 2023. arXiv preprint arXiv:2304.13949.
22. Radford A, Kim JW, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: International Conference on machine learning, PMLR, 2021; pp. 8748–8763.
23. Zhang F, Tian S, Yu L, Yang Q. Multi-channels prototype contrastive learning with condition adversarial attacks for few-shot event detection. Neural Process Lett. 2024;56(31):30–1.
24. Soares E, Angelov P, Suri N. Similarity-based deep neural network to detect imperceptible adversarial attacks. In: Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI). 2022.
25. Aghasanli A, Kangin D, Angelov P. Interpretable-through-prototypes deepfake detection for diffusion models. In: Proceedings of IEEE/CVF International Conference on computer vision (ICCV). 2023.
26. Bouter ML, Pardo JL, Geradts Z. Protoexplorer: interpretable forensic analysis of deepfake videos using prototype exploration and refinement. 2023. arXiv preprint arXiv:2309.11155.
27. Wang T, Chow K-C. Noise based deepfake detection via multihead relative-interaction. In: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI). 2023.
28. Zheng Y, Bao J, Chen D, Zeng M. Exploring temporal coherence for more general video face forgery detection. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). 2021.
29. Raza MA, Malik K. Multimodaltrace: deepfake detection using audiovisual representation learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023.
30. Pellcier AL, Li Y, Angelov P. Pudd: towards robust multi-modal prototype-based deepfake detection. In: CVPR Workshop, 2021; p. 3809.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.