

Interpretable-Aware Privacy Preserving Framework for Deepfake Detection using Federated ResNet and Explainable AI Techniques

Pochampally Chandra Sekhar Reddy

`pchandrasedkhar.scholar24@ifheindia.org`

ICFAI Foundation for Higher Education (IFHE)

Kongara Srinivasa Rao

ICFAI Foundation for Higher Education (IFHE)

Research Article

Keywords: Federated Learning, Deepfake Detection, ResNet-18, Privacy-Preserving, Image Forensics

Posted Date: July 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10197752/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Interpretable-Aware Privacy Preserving Framework for Deepfake Detection using Federated ResNet and Explainable AI Techniques

Pochampally Chandra Sekhar Reddy^{1*}, Kongara Srinivasa Rao²

^{1*}Department of Computer Science and Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education (IFHE), Hyderabad-501203, India.

pchandrasekhar.scholar24@ifheindia.org

²Department of Computer Science and Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education (IFHE), Hyderabad-501203, India.

drksrao@ifheindia.org

ABSTRACT: The deepfake has grown faster with generative artificial intelligence which has made it much easier to produce realistic manipulated images and videos. These synthetic media raise serious concerns about the trustworthiness of information, digital security, personal privacy, and overall societal confidence. Traditional methods for detecting deepfakes depend on centralized deep learning systems that need to gather large amounts of images on a single server. To overcome the traditional challenges, this work introduces a new deepfake detection system designed to protect privacy by combining Federated Learning (FL) with Deep Learning (DL) and forensic analysis techniques. The system uses FL, which allows models to train on data stored locally across many devices without transferring the raw images anywhere. It's built on the ResNet-18 architecture, chosen because it balances good detection capabilities with reasonable computational requirements. ResNet-18 residual learning helps the model train steadily and extract useful features, which fits well with the varied and distributed data environments in federated settings. Experiments are conducted using the FaceForensics++ dataset, a standard resource in media forensics studies which cover four types of manipulations DeepFakes, Face2Face, FaceSwap, and Neural Textures. Since client data distributions differ, which can slow down or destabilize federated training, the FedProx algorithm is used. FedProx adds a term to the local loss function to keep client model updates close to the global model, which helps reduce divergence between clients and improves overall training stability. After local training, only the updated model weights are sent to a central server, where they are averaged to create an improved global model. The experimental results show that the accuracy of 97.20%, precision 96.30%, recall of 96.72, f1-score of 96.90% to keep track of how well the model is learning and generalizing. This study demonstrates a unified framework integrating FL, CNNs, and explainability to enable secure, privacy-preserving, and interpretable deepfake detection, enhancing transparency and user trust.

Keywords: Federated Learning, Deepfake Detection, ResNet-18, Privacy-Preserving, Image Forensics.

1. INTRODUCTION

The rapid growth of Artificial Intelligence (AI) and Deep Learning (DL) has significantly influenced the way digital media is created, shared, and consumed [1]. One of the most impactful outcomes of this progress is the emergence of deepfake technology, which enables the generation of highly realistic synthetic images and videos using advanced learning frameworks such as generative adversarial networks [2]. While these developments have opened new possibilities in areas like entertainment, virtual communication, and content creation, they have also introduced serious concerns. Misuse of deepfake technology has been associated with misinformation, identity impersonation, financial fraud, and violations of personal privacy, thereby raising questions about the reliability of digital content in modern communication environments [3]. As deepfake generation techniques continue to improve, the visual quality and realism of manipulated media have reached a level where distinguishing between authentic and fabricated content

is increasingly challenging. Earlier detection methods relied primarily on handcrafted features and traditional machine learning techniques, which are often insufficient when dealing with complex and high-quality forgeries [4]. More recent approaches based on DL have shown improved capability in identifying subtle inconsistencies in manipulated media [5]. Among these, residual network architectures have gained attention due to their ability to learn deeper feature representations effectively. In particular, ResNet-18 offers a practical balance between computational efficiency and detection capability. Compared to more computationally intensive models such as EfficientNet, or lightweight alternatives like MobileNet that may miss fine-grained artifacts, ResNet-18 provides stable training through residual connections while still capturing meaningful visual cues necessary for reliable deepfake detection [6]. Despite these advancements, most existing detection systems are designed using centralized learning frameworks, where

large volumes of data are collected and processed on a single server [7]. This approach presents several challenges, especially when dealing with sensitive or distributed data. Concerns related to data privacy, ownership, and regulatory compliance have become

increasingly important, particularly in real-world applications where data cannot be freely shared across institutions or devices. Figure 1 shows work flow of explainable deepfake detection framework.

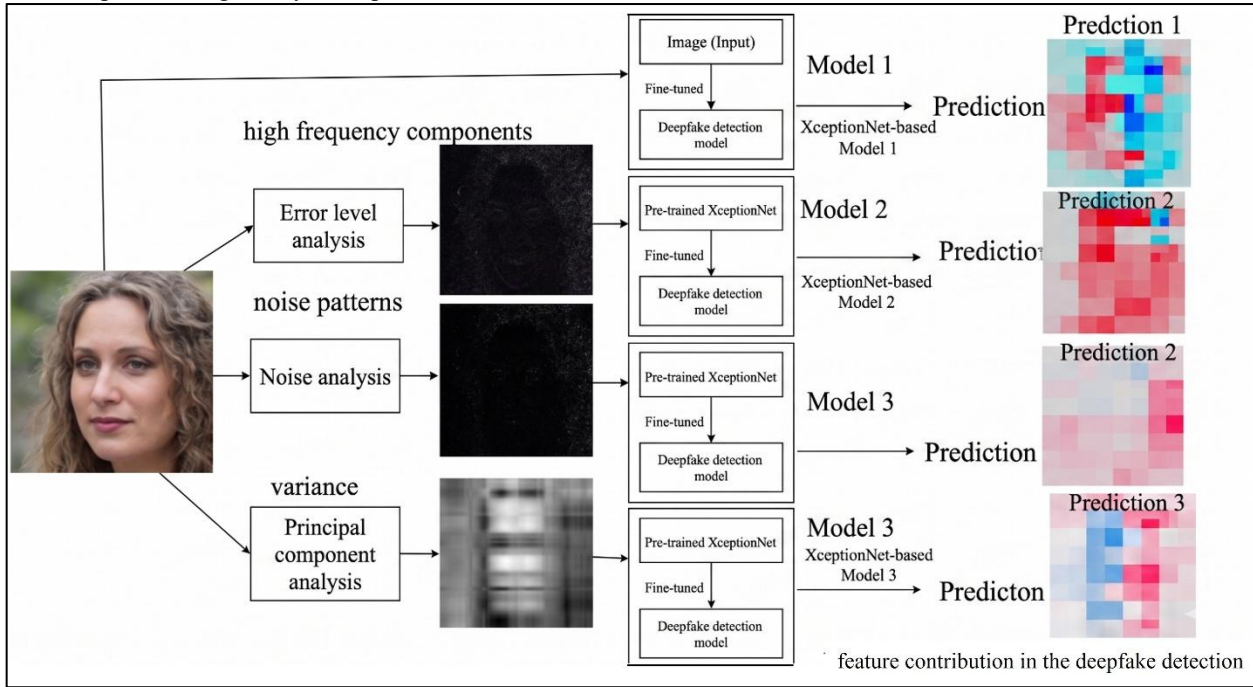


Figure 1. Represents the work flow of explainable deepfake detection framework

In addition, centralized systems often introduce higher communication costs, storage limitations, and potential security vulnerabilities, making them less suitable for large-scale and privacy-sensitive environments [8]. To address these issues, FL has emerged as an alternative approach that enables collaborative model training without requiring direct data sharing. In this paradigm, models are trained locally on individual devices, and only model updates are communicated to a central server for aggregation. This allows the system to learn from diverse data sources while preserving user privacy [9]. However, FL introduces its own set of challenges, particularly due to variations in data distribution across clients, which can affect training stability and model convergence.

To mitigate these issues, optimization strategies such as FedProx can be employed to regulate local updates and maintain consistency across participating clients, thereby improving the robustness of the overall training process. In addition to achieving high detection accuracy, there is a growing need for transparency in automated decision-making systems. The use of ResNet-18 supports efficient feature extraction, and the incorporation of federated optimization techniques improves training stability across heterogeneous data sources. This study aims to contribute toward the development of a reliable, scalable, and privacy-aware deepfake detection system that aligns with the evolving requirements of real-world digital environments.

2. LITERATURE REVIEW

The rapid advancement of DL and computer vision technologies has led to the widespread emergence of

deepfake content, raising serious concerns regarding security, privacy, and information authenticity. Consequently, significant research efforts have been directed toward developing effective deepfake detection techniques. Existing approaches span a wide range of methodologies, including image-based, temporal, frequency-domain, and multimodal analysis, along with the adoption of advanced neural network architectures. This section reviews shown in table 1 provides a comprehensive overview of these techniques, highlighting their strengths, limitations, and evolution. It also identifies key research gaps, particularly in integrating privacy-preserving learning with robust and interpretable deepfake detection systems.

Recent research has proposed a variety of techniques for deepfake detection, ranging from unimodal visual or audio methods to multimodal fusion, physiological signal integration, and transformer-based architectures [10]. While prior methods have demonstrated strong performance in static images analysis, audio-visual synchronization, and physiological cues, none simultaneously integrate micro-expression mining, multi-modal metadata fusion, and privacy-preserving federated learning optimized for real-time IoT-based Video-KYC systems [11]. With recent advancements in computer vision techniques [13], it is possible for people to conduct facial expression transformation and face swapping automatically. Face forgery methods [14] are emerging, among which the most popular are Deepfakes [15], Face2Face [16], FaceSwap [17], and NeuralTextures [18]. As introduced in [19] a typical video face manipulation process consists of three steps: (i) Decompose the original video into independent

frames and detect and crop faces, (ii) manipulate face images using pretrained forgery models, and (iii) blend each forged face back into the corresponding original frame. Similarly, inconsistencies in artifacts across adjacent frames, particularly in sensitive regions like the eyes, highlight limitations in frame-level detection approaches [20]. Based on these observations, methods

such as ICM [21] have been proposed to explore dynamic inconsistencies between adjacent frames. The increasing misuse of deepfake technology for malicious purposes has raised serious legal and ethical concerns, creating an urgent need for reliable detection systems [22].

Table 1. Summary of state-of-the-art deepfake detection techniques with quantitative performance analysis

Reference	Methods	Dataset	Results	Conclusion
Ju Y et al. [23]	CNN-based facial feature extraction.	FaceForensics++	Accuracy: 96.2%, Precision: 95.4%, Recall: 94.8%	CNN models effectively identify manipulated facial regions in deepfake content.
Wang et al. [24]	Face X-Ray + CNN	FaceForensics++, DeepFakeDetection	AUC: 96.5%, F1-Score: 97.1%	Artifact localization significantly improves cross-manipulation detection robustness.
Heidari et al. [25]	MesoNet	FaceForensics++	Accuracy: 94.5%, Inference Time: 0.02 s/frame	Lightweight networks provide efficient real-time deepfake detection.
Ashani et al. [26]	EfficientNet	Celeb-DF, DFDC	Accuracy: 93.3%, ROC-AUC: 95.1%	Transfer learning improves scalability and generalization performance.
Borade et al. [27]	Federated CNN	Celeb-DF, DFDC	Accuracy: 95.8%, Communication Overhead Reduced: 38%	Federated learning preserves privacy while maintaining competitive accuracy.
Yee et al. [28]	ResNet-50	FaceForensics++	Accuracy: 93.2%, Precision: 94.6%, Recall: 97.1%	Residual networks capture subtle spatial inconsistencies effectively.
Yang et al. [29]	Grad-CAM	DFDC Preview Dataset	Accuracy: 93.7%, User Trust Score Improved: 29%	Explainable AI improves transparency and interpretability of predictions.
Wang et al. [30]	Vision Transformer (ViT)	Celeb-DF	Accuracy: 96.7%, F1-Score: 96.0%	Attention mechanisms enhance detection of fine forgery artifacts.
Stroebel et al. [31]	Federated ResNet	CIFAKE, FaceForensics++	Accuracy: 96.9%, Privacy Leakage Reduced: 42%	Federated ResNet supports scalable and secure AI collaboration.
Kumar M et al. [32]	Federated CNN + LIME + Grad-CAM	Celeb-DF, FaceForensics++	Accuracy: 95.5%, Explainability Consistency: 91.4%	Combining FL and XAI achieves trustworthy and privacy-aware deepfake detection.

Existing detection approaches can be broadly categorized into three groups: image-based, temporal-based, and frequency-domain methods [33]. With the rapid advancement of deep learning, CNN-based approaches have gained significant popularity [34]. Some CNN-based methods focus on physiological artifacts, while others rely purely on data-driven feature extraction [35]. Various architectures such as Xception, MesoNet, and Two-stream networks have been explored extensively [36]. Deep neural networks such as ResNet50 address the vanishing gradient problem, enabling effective training of deeper architectures and improved feature extraction [37]. ResNet50 has been widely used for tasks such as object recognition and is particularly effective in deepfake detection due to its ability to capture fine-grained details in image frames [38]. Residual Neural Networks (ResNet) have achieved high detection accuracy by addressing gradient vanishing issues in deep architectures [39]. For instance, [40] developed a ResNet-50-based mask detection model achieving an accuracy of 96%, outperforming other models in precision and recall [41]. Similarly, [42] proposed

ResNet-Swish-Dense54 for detecting manipulated video content, achieving strong performance across multiple challenging datasets. However, existing deepfake detection methods have demonstrated considerable progress through the use of DL architectures, multimodal analysis, and artifact-based detection techniques. However, challenges remain in handling diverse manipulation methods, ensuring real-time performance, and maintaining user data privacy. Addressing these limitations is essential for developing robust, scalable, and privacy-preserving deepfake detection systems suitable for real-world deployment.

3. METHODOLOGY

The proposed methodology introduces a privacy-preserving deepfake detection framework that leverages FL combined with a lightweight CNN architecture. The proposed work addresses critical challenges related to data privacy, ownership, and regulatory compliance in real-world applications. The overall workflow is structured into a sequence of stages including dataset representation, preprocessing, local model training,

federated optimization, and explainable inference. To ensure stable training in heterogeneous environments, federated optimization techniques are incorporated. Finally, the model generates predictions along with interpretable explanations using forensic analysis methods. The dataset representation and preprocessing stages are described in detail in the following subsections.

3.1 Dataset Description

This section describes the dataset composition, feature representation, and experimental configuration adopted for the proposed privacy-preserving deepfake detection framework. The system utilizes the FaceForensics++ dataset [1, 14, 18], which is one of the most widely used and standardized benchmarks in the field of digital media forensics. It provides a comprehensive collection of both real and manipulated facial videos generated using multiple deepfake techniques. The dataset is specifically designed to support the development and evaluation of deepfake detection models under realistic conditions, making it highly suitable for this study. The FaceForensics++ dataset (See Table 2) contains approximately 7000 videos, including both original (authentic) and manipulated content.

Table 2. Dataset attributes for deepfake detection

Attribute Name	Description	Example Values
Image ID	Unique identifier for each image	IMG_001
Image Type	Indicates real or fake image	Real / Fake
Manipulation Type	Type of deepfake technique	FaceSwap, DeepFakes
Compression Level	Image compression setting	C23
Client ID	Assigned federated client	1, 2, 3, 4
Image Resolution	Input image dimensions	224 × 224
Pixel Values	Normalized intensity values	[0–1]
Label	Ground truth class	0 / 1

The dataset representation is formally defined as:

$$X = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^{H \times W \times C} \quad (1)$$

where x_i denotes the i^{th} image sample, N represents the total number of samples, and H , W , and C correspond to the height, width, and number of channels of the image respectively. The label representation is given as:

$$Y = \{y_i\}_{i=1}^N, y_i \in \{0, 1\} \quad (2)$$

where $y_i = 0$ denotes real images and $y_i = 1$ denotes deepfake images. The dataset attributes define the structure of the input data, where pixel values represent visual features and labels provide supervision for classification. However, raw image data may contain

inconsistencies and noise, which necessitates preprocessing before model training. The next section describes the dataset processing pipeline in detail.

3.2 Dataset Processing

This section presents the step-by-step preprocessing and learning pipeline used to transform raw image data into structured representations suitable for federated deep learning. The process consists of normalization, local training, federated optimization, aggregation, prediction, and explainability stages, where each step refines the data or model while preparing it for the subsequent stage.

3.2.1 Image Normalization

The initial stage focuses on preprocessing the input images to ensure consistency in representation. Since images may vary in brightness, contrast, and resolution, normalization is applied to standardize pixel intensity values across the dataset:

$$x'_i = \frac{x_i - \mu}{\sigma} \quad (3)$$

where x_i represents the original image, μ denotes the mean pixel value, and σ is the standard deviation. The normalized output x'_i ensures that the input data is centered and scaled, thereby improving numerical stability during training. In addition, all images are resized to a fixed resolution (224 × 224) to match the input requirements of the ResNet-18 architecture, and augmentation techniques such as rotation, flipping, and brightness variation are applied to enhance data diversity. This preprocessing stage reduces variability in input data and prepares the images for efficient feature learning in the subsequent training phase.

3.2.2 Local Model Training

Following preprocessing, the normalized images are distributed across multiple client nodes, where each client independently trains a local ResNet-18 model using its assigned dataset. The objective of this stage is to minimize the classification error between predicted outputs and ground truth labels, which is formulated as:

$$L_i(w) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f(x_j; w), y_j) \quad (4)$$

where $L_i(w)$ denotes the loss for client i , n_i represents the number of local samples, $f(x_j; w)$ is the model prediction, and y_j is the corresponding label. During this stage, the ResNet-18 architecture extracts hierarchical features such as edges, textures, and subtle inconsistencies that are indicative of deepfake manipulations. As each client learns from its own data distribution, this stage enables localized feature learning without requiring data sharing, but it also introduces potential variations in model updates due to heterogeneous data.

3.2.3 Federated Optimization

To address the divergence caused by non-identically distributed data across clients, the FedProx optimization strategy is incorporated by modifying the local loss function as:

$$L_i^{prox}(w) = L_i(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (5)$$

where w represents the local model parameters, w^t denotes the global model parameters, and μ is a regularization coefficient. The additional proximal term constrains local updates to remain close to the global model, thereby improving stability and convergence in federated environments. This step plays a critical role in ensuring that independently trained models do not diverge significantly, especially under non-IID data conditions, and prepares the updated local models for aggregation at the central server

3.2.4 Global Aggregation

After local optimization, each client transmits its updated model parameters to a central server, where they are aggregated to form a unified global model using weighted averaging:

$$w^{t+1} = \sum_{i=1}^K \frac{n_i}{n} w_i^t \quad (6)$$

where K represents the number of clients, n_i denotes the number of samples at client i , and n is the total number of samples. This aggregation mechanism ensures that clients with larger datasets contribute proportionally more to the global model, resulting in a balanced integration of knowledge from all distributed sources. The updated global model captures collective learning across clients and is then redistributed back to them, enabling iterative refinement through multiple communication rounds until convergence is achieved.

3.3 Proposed Methodology

The proposed methodology integrates FL with a deep convolutional neural network to enable privacy-preserving and robust deepfake image detection. Instead of relying on centralized data collection, the system distributes the training process across multiple clients, each maintaining local data. A ResNet-18 architecture is employed as the base model due to its balanced trade-off between computational efficiency and feature extraction capability. The overall framework consists of iterative local training and global aggregation phases, ensuring that the model progressively improves while maintaining data privacy. Additionally, the incorporation of a proximal optimization strategy enhances stability in heterogeneous data environments, making the approach suitable for real-world distributed systems.

3.3.1 Local Model Architecture (ResNet-18)

The ResNet-18 architecture is employed as the core DL model for feature extraction and classification of deepfake images. It is designed to effectively capture both low-level and high-level spatial features while addressing the degradation problem in deep neural networks. Figure 2 represents the proposed model ResNet18 architecture work flow. The model utilizes residual learning with skip connections, allowing it to preserve important information from earlier layers while learning complex representations.

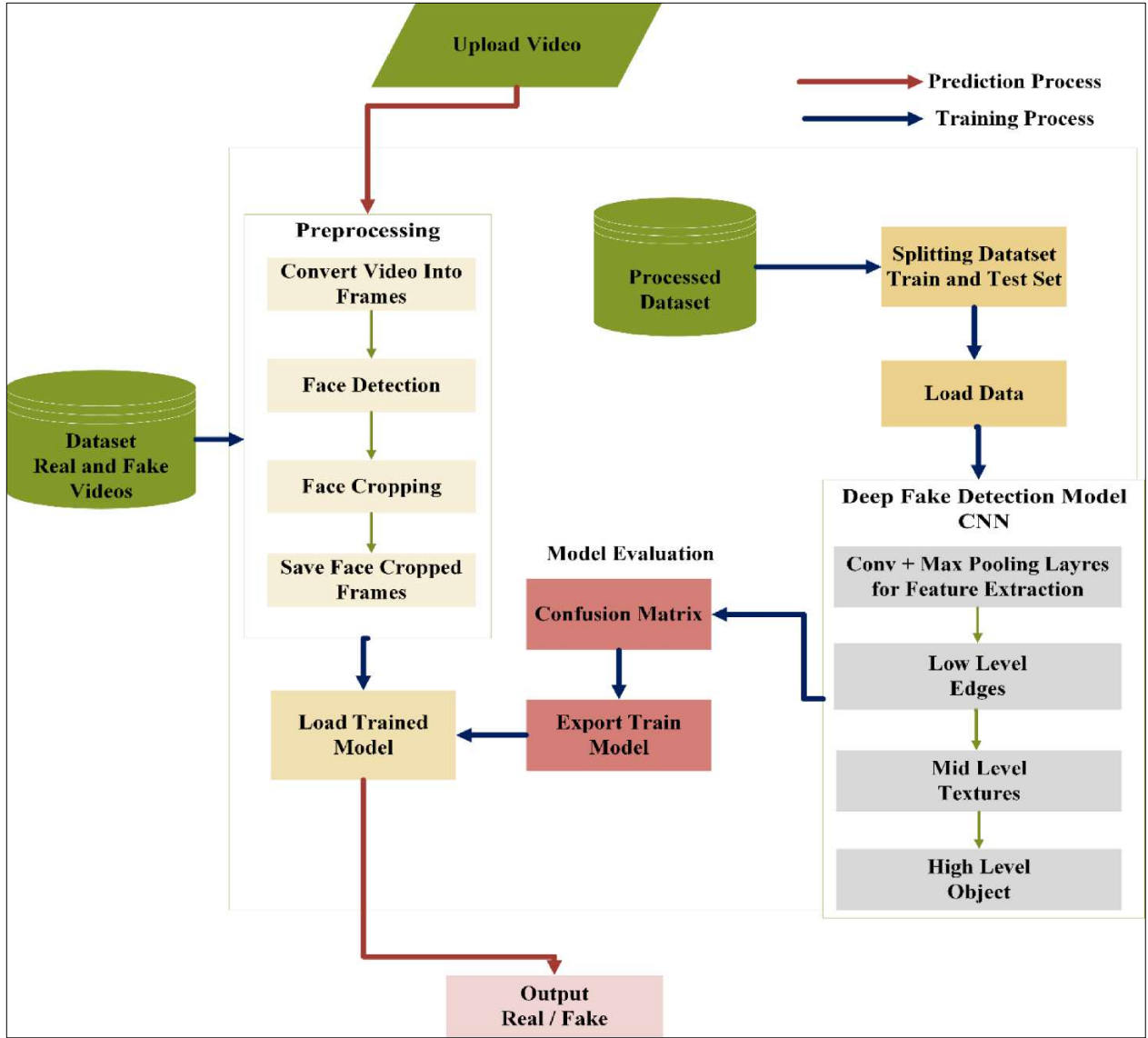


Figure 2. Proposed model ResNet flow chart

This is particularly significant in deepfake detection, where subtle artifacts such as unnatural blending, texture inconsistencies, and facial distortions must be identified accurately. The fundamental residual mapping in ResNet is defined as:

$$H(x)=F(x)+x \quad (7)$$

where x represents the input feature map, $F(x)$ denotes the learned residual function, and $H(x)$ is the final output. This formulation enables efficient gradient flow and ensures stable training, which is essential in federated environments with distributed datasets. Feature extraction is performed using convolution operations:

$$F = I * K + b \quad (8)$$

where I is the input image, K represents convolution kernels, and b is the bias term. This operation captures spatial patterns such as edges and texture variations, which are critical for distinguishing real images from deepfake manipulations. To introduce non-linearity, the Rectified Linear Unit (ReLU) activation function is applied:

$$f(x)=\max(0,x) \quad (9)$$

where x is the input feature and $f(x)$ are the activated output. This enhances the model's ability to learn complex and non-linear patterns present in manipulated images. To stabilize training and improve convergence across different clients, batch normalization is used:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (10)$$

where μ and σ^2 denote the mean and variance of the batch, respectively. This reduces internal covariate shift and ensures consistent feature scaling. The extracted features are then mapped to the classification space using a fully connected layer:

$$z=Wx+b \quad (11)$$

where W represents weights and b is bias. This transformation converts learned features into logits for classification. The probability of each class (real or fake) is computed using the SoftMax function:

$$P(y=i|x)=\frac{e^{z_i}}{\sum_j e^{z_j}} \quad (12)$$

where z_i is the logit for class i , and $P(y=i|x)$ represents the predicted probability. This provides confidence

scores that are later used in explainability. The model is optimized using binary cross-entropy loss:

$$L = -[y \log(\hat{p}) + (1-y) \log(1-\hat{p})] \quad (13)$$

where y is the ground truth label and ' $\hat{p}$ ' is the predicted probability. This loss function ensures accurate classification between real and deepfake images.

3.3.2 Federated Optimization and Model Aggregation

The federated optimization process forms the core of the proposed framework by enabling multiple distributed clients to collaboratively train a shared global model without exchanging raw image data as shown in Figure 3. In this work, each client independently trains a local ResNet-18 model using its private dataset, which may

contain different distributions of real and deepfake images due to variations in manipulation techniques, lighting conditions, and facial characteristics. The local training process begins with the reception of global model parameters w^t at communication round t , after which each client performs optimization on its dataset using gradient-based learning. The local update rule is defined as:

$$w_k^{t+1} = w^t - \eta \nabla F_k(w^t) \quad (14)$$

where η denotes the learning rate controlling the magnitude of updates, and $\nabla F_k(w^t)$ represents the gradient of the local loss function computed over dataset D_k .

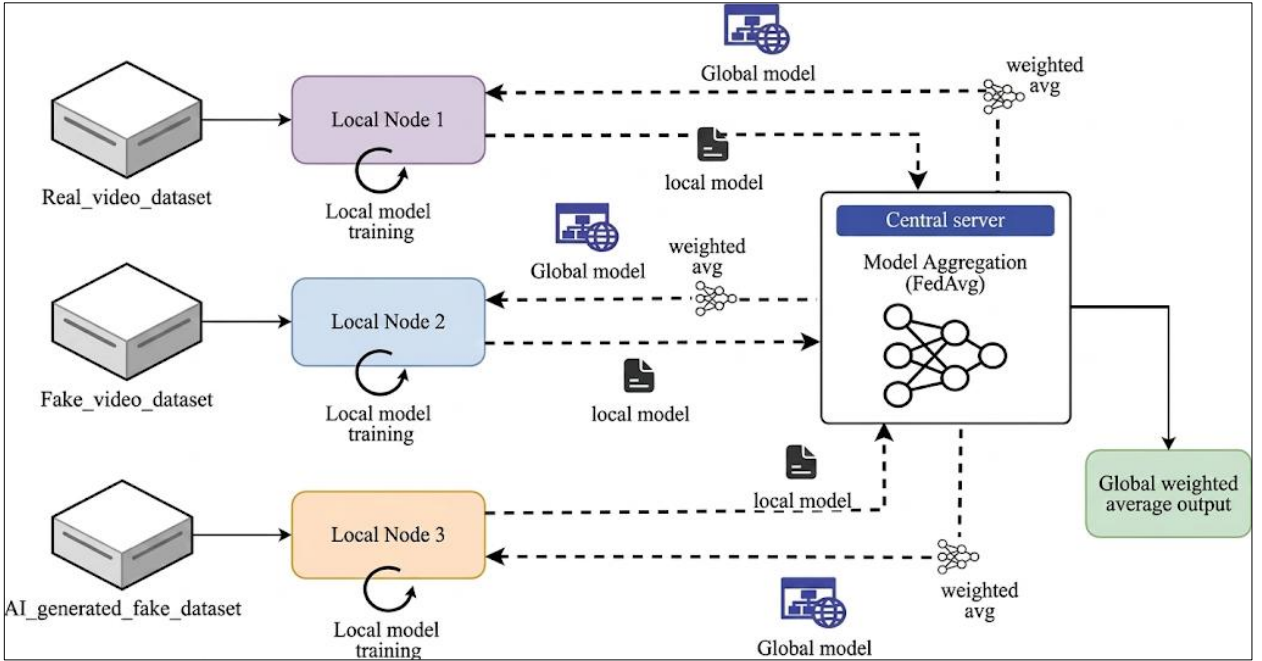


Figure 3. Preprocessing parallel federated learning with three edges-client

This step allows each client to adapt the model to its specific data characteristics, which is particularly important in deepfake detection since different clients may contain different types of forged content such as face swaps or neural texture manipulations. However, due to the non-IID nature of distributed datasets, local models may diverge significantly from each other during training. To mitigate this issue, the FedProx optimization strategy is incorporated by modifying the local objective function as:

$$L_k = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (15)$$

where μ is a regularization parameter that controls the strength of the proximal constraint, and $\|w - w^t\|^2$ penalizes large deviations from the global model. This ensures that local updates remain consistent with the global objective, thereby improving convergence stability across clients. After completing local training, each client transmits only its updated model weights w_k^t to the central server, ensuring that sensitive image data never leaves the local device. The server then performs aggregation using the Federated Averaging (FedAvg) algorithm:

$$w^{t+1} = \sum_{k=1}^K \frac{n_k}{n} w_k^t \quad (16)$$

where n_k is the number of samples at client k , and $n = \sum_{k=1}^K n_k$ represents the total number of samples across all clients. This weighted averaging ensures that clients with larger datasets have a greater influence on the global model, thereby improving overall generalization. To further ensure convergence, the change in global model parameters is monitored using:

$$\Delta w = \|w^{t+1} - w^t\| \quad (17)$$

and training is terminated when:

$$\Delta w < \epsilon \quad (18)$$

where ϵ is a predefined threshold. This stopping criterion ensures that the model has stabilized and further updates yield minimal improvement. Additionally, communication efficiency is a critical aspect of federated learning. Instead of transmitting entire datasets, only model parameters are exchanged, significantly reducing bandwidth usage and enhancing scalability. This is particularly beneficial in real-world deployments where clients may operate on resource constrained devices such

as mobile phones or edge systems. Algorithm 1 describes the federated deepfake detection using ResNet-18.

Algorithm 1: Federated Deepfake Detection using ResNet-18

Input: Distributed datasets $D_1, D_2, \dots, D_k$ across K clients

Output: Trained global model w^*

Begin

Step 1: Initialize global model parameters

$w \leftarrow \text{InitializeModel}()$

Step 2: Set the number of communication rounds

$T \leftarrow \text{TotalRounds}$

Step 3: For each communication round $t = 1$ to T , perform:

Step 4: Broadcast the current global model to all clients

$w^t \rightarrow \text{Clients}$

Step 5: For each client $k \in \{1, 2, \dots, K\}$ in parallel:

Step 6: Receive global model and initialize local model

$w_k \leftarrow w^t$

Step 7: Perform local training for E epochs:

Step 8: For each batch $(x_i, y_i) \in D_k$:

Step 9: Compute prediction using local model

$\hat{y} \leftarrow f_{w_k}(x_i)$

Step 10: Compute loss with regularization

$L \leftarrow \ell(\hat{y}, y_i) + \frac{\mu}{2} \|w_k - w^t\|^2$

Step 11: Compute gradients of the loss

$g \leftarrow \nabla L$

Step 12: Update local model parameters

$w_k \leftarrow w_k - \eta \cdot g$

Step 13: End batch loop

Step 14: End local epoch loop

Step 15: Send updated local model weights to server

$w_k \rightarrow \text{Server}$

Step 16: End client loop

Step 17: Aggregate client models using Federated Averaging (FedAvg)

$w^{t+1} \leftarrow \sum \frac{n_k}{n} \cdot w_k$

Step 18: Check convergence condition if $\|w^{t+1} - w^t\| < \epsilon$, then terminate training

Step 19: End communication round loop

Step 20: Return final global model

$w^* \leftarrow w^{t+1}$

end

Overall, the federated optimization strategy in this work ensures that the model effectively learns from distributed deepfake datasets while maintaining privacy, improving robustness to data heterogeneity, and enabling scalable deployment across multiple devices.

3.4 Explainable Decision Generation on Confidence-Based Prediction and Inference

The explainable decision generation module plays a crucial role in enhancing the transparency and trustworthiness of the proposed deepfake detection system. Unlike traditional black-box models that provide only classification outputs, this framework is designed to

offer interpretable insights into the reasoning behind each prediction. During the inference phase, the trained ResNet-18 model processes an input image and produces output logits, which are converted into probabilities using the SoftMax function:

$$P(y=i|x) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (19)$$

where z_i represents the logit corresponding to class i , and $P(y=i|x)$ denotes the predicted probability. This formulation allows the model to assign confidence scores to each class, providing a probabilistic interpretation of the prediction. The final classification is determined using:

$$\hat{y} = \arg\max_i P(y=i|x) \quad (20)$$

where $\hat{y}$ represents the predicted class label. This ensures that the class with the highest probability is selected as the output. To measure the uncertainty of predictions, entropy is calculated as:

$$H(p) = -\sum p \log p \quad (21)$$

where higher entropy values indicate greater uncertainty.

Algorithm 2: Confidence-Based Deepfake Detection using ResNet-18

Input: Image x

Output: Prediction (Real/Fake), Confidence Score, Uncertainty

Begin

Step 1: Preprocess the input image (resize and normalize)

$x \leftarrow \text{Preprocess}(x)$

Step 2: Pass the image through the trained ResNet-18 model

$z \leftarrow f(x)$

Step 3: Compute logits from the model output

$z \leftarrow \text{Logits}(x)$

Step 4: Apply SoftMax to obtain probability distribution

$$P(y|x) \leftarrow \frac{\exp(z)}{\sum \exp(z)}$$

Step 5: Determine predicted class label

$\hat{y} \leftarrow \arg\max(P(y|x))$

Step 6: Compute confidence score

$C \leftarrow \max(P(y|x))$

Step 7: Compute entropy to measure uncertainty

$H \leftarrow -P(y|x) \log \sum P(y|x)$

Step 8: Check confidence threshold if $C < \text{threshold}$, mark prediction as uncertain.

Step 9: Return prediction, confidence score, and uncertainty

$(\hat{y}, C, H)$

end

This is particularly useful in identifying ambiguous cases where the model may not be fully confident in its prediction. Additionally, confidence calibration can be assessed using:

$$C = \max P(y|x) \quad (22)$$

where C represents the confidence score associated with the predicted class. Higher values of C indicate stronger

confidence in the decision. To further refine predictions, a threshold-based decision rule is applied:

$$\hat{y} = \begin{cases} \text{Fake,} & \text{if } P(\text{fake} | x) > \tau \\ \text{Real,} & \text{otherwise} \end{cases} \quad (23)$$

where τ is a predefined threshold. This allows control over sensitivity and specificity in deepfake detection.

3.4.1 Forensic Feature Analysis for Explainability

To enhance the interpretability of the deepfake detection system, forensic feature analysis is performed to identify visual artifacts and structural inconsistencies present in manipulated images. Unlike the prediction module, which provides probabilistic outputs, this stage focuses on extracting meaningful visual cues that justify the model's decision. Deepfake images often contain subtle irregularities such as unnatural edges, blending artifacts, and hidden noise patterns, which can be effectively captured using image processing techniques. To detect structural inconsistencies, edge detection is performed by computing the gradient magnitude of the image:

$$G = \sqrt{G_x^2 + G_y^2} \quad (24)$$

where G_x and G_y represent the horizontal and vertical gradients of the image, respectively, and G denotes the overall edge intensity. This operation highlights sharp transitions in pixel intensity, which are often indicative of artificial boundaries introduced during deepfake generation, particularly around facial regions. To extract hidden noise patterns, residual noise analysis is performed by subtracting a smoothed version of the image from the original:

$$R = x - x_{\text{smooth}} \quad (25)$$

where x is the original image and x_{smooth} represents the Gaussian-blurred version of the image. The residual R captures high-frequency components such as noise and fine textures, which often reveal synthetic artifacts not visible to the human eye. To ensure consistent visualization, the residual values are normalized:

$$R_{\text{norm}} = \frac{R - \min(R)}{\max(R) - \min(R)} \quad (26)$$

where R_{norm} represents the normalized residual map. This scaling improves visibility of subtle noise patterns across different images. The identified structural and noise-based features are then combined to generate a comprehensive explanation map:

$$E = \alpha G + \beta R_{\text{norm}} \quad (27)$$

where α and β are weighting parameters that control the contribution of edge and noise features, respectively, and E represents the final explanation map. This combined representation highlights regions that are most likely to contain manipulation artifacts.

Algorithm 3: Forensic Feature Analysis for Deepfake Explanation

Input: Image x , Predicted label $\hat{y}$

Output: Explanation map and forensic insights
Begin

Step 1: Receive the input image

$x \leftarrow \text{InputImage}$

Step 2: Preprocess the image if required

If necessary, convert image to grayscale for intensity-based analysis

$x \leftarrow \text{Grayscale}(x)$

Step 3: Perform edge detection using gradient computation

Step 4: Compute horizontal and vertical gradients

$G_x \leftarrow \text{Gradient}_x(x)$

$G_y \leftarrow \text{Gradient}_y(x)$

Step 5: Compute edge magnitude

$$G \leftarrow \sqrt{G_x^2 + G_y^2}$$

Step 6: Apply image smoothing using Gaussian blur

$x_{\text{smooth}} \leftarrow \text{GaussianBlur}(x)$

Step 7: Extract residual noise from the image

$R \leftarrow x - x_{\text{smooth}}$

Step 8: Normalize residual features for visualization

$R \leftarrow \text{Normalize}(R)$

Step 9: Detect suspicious regions

Identify regions with high values in G and R as potential manipulated areas

Step 10: Combine forensic features to generate explanation map

Explanation map $\leftarrow \alpha \cdot G + \beta \cdot R$

Step 11: Visualize explanation map

Overlay explanation map on the original image

Overlay(Explanation_map, x)

Step 12: Return outputs

Return predicted label $\hat{y}$

Return explanation map

Highlight suspicious regions

End

To localize suspicious regions, a thresholding operation is applied:

$$M = \begin{cases} 1, & \text{if } E > \delta \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

where M is a binary mask and δ is a predefined threshold. This mask identifies regions with high artifact intensity, enabling clear visualization of manipulated areas. Finally, the explanation map is overlaid on the original image to produce an interpretable output, highlighting the regions that contributed to the model's decision. This approach not only improves transparency but also provides users with a visual understanding of why an image is classified as real or deepfake, thereby increasing trust and reliability in the system.

4. RESULTS

This section presents the experimental evaluation of the proposed privacy-preserving deepfake detection framework, which combines federated learning with a ResNet-18 architecture. The purpose of this evaluation is to examine how effectively the model distinguishes between authentic and manipulated facial images while ensuring that data privacy is preserved throughout the training process. The suggested approach was successfully validated using an FaceForensics++ dataset. Training was performed on NVIDIA GE Force 4 GB GPU, with a learning rate of 0.001, the batch size was set

to 48. Training required only 45 min, whereas testing and classification from an unseen image took on average just 12s. The combination of federated learning, an optimized neural network architecture, and interpretability mechanisms ensures that the system can be effectively applied in real-world environments where reliability and confidentiality are essential.

4.1 Quantitative Evaluation of the Proposed Model

The evaluation is performed using a combination of visual forensic analysis and statistical performance metrics to comprehensively assess the model's effectiveness. The results not only demonstrate the classification capability of the ResNet-18-based federated model but also highlight its ability to provide interpretable insights into deepfake detection. The evaluation is structured in a progressive manner, beginning with visual analysis of image artifacts, followed by classification-based metrics and advanced feature representations. Each metric contributes to understanding different aspects of the system, including spatial inconsistencies, prediction accuracy, error behavior, and feature distribution. This layered evaluation ensures that the proposed system is not only accurate but also robust and explainable in real-world

scenarios. To achieve this, the system applies forensic feature extraction techniques that transform the original image into representations that emphasize structural and hidden irregularities. The edge artifact map is generated by computing the gradient of the image, which highlights regions of sharp intensity change. This is mathematically defined as:

$$G = \sqrt{G_x^2 + G_y^2} \quad (29)$$

$$G_x = \frac{\partial I}{\partial x}, G_y = \frac{\partial I}{\partial y} \quad (30)$$

$$E(x, y) = |G_x| + |G_y| \quad (31)$$

where I represents the input image, and G_x, G_y denote gradients along horizontal and vertical directions. The magnitude G captures edge strength, while $E(x, y)$ represents combined edge intensity. In the observed edge map, strong contours are visible around facial boundaries, text regions, and manipulated areas. These edges often appear more pronounced or irregular in deepfake images due to imperfect blending and generation artifacts. This visualization helps the model focus on structural inconsistencies that are critical for classification.

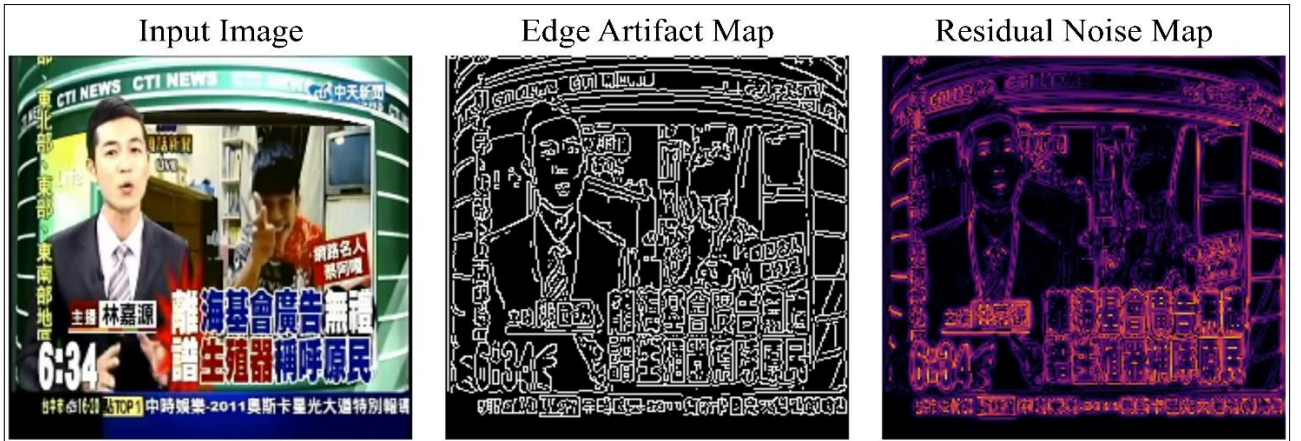


Figure 4. Smooth edge transitions and natural residual patterns suggest authentic media which represents the original.

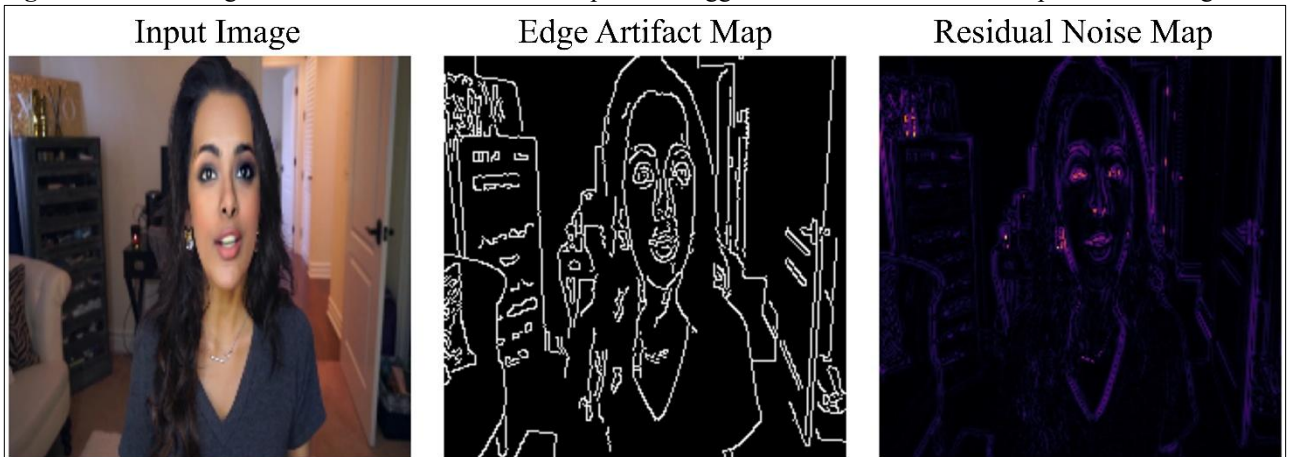


Figure 5. High edge discontinuity and abnormal residual noise indicate synthetic blending which shows deepfake image

In addition to edge detection, residual noise analysis is performed to capture hidden high-frequency components that are not visible in the original image. This is computed as:

$$R = I - I_{smooth} \quad (32)$$

$$I_{smooth} = \text{Gaussian}(I, \sigma) \quad (33)$$

$$R_{norm} = \frac{R - R_{min}}{R_{max} - R_{min}} \quad (34)$$

where I_{smooth} is a blurred version of the image and R represents residual noise. The normalized residual enhances visibility of subtle artifacts. The residual noise map clearly reveals unnatural patterns distributed across the image, especially around facial regions. These patterns are typically introduced during deepfake generation and are difficult to detect using conventional methods. Figure 4 and 5 shows by combining edge and residual analysis, the system provides a strong explainability component, allowing users to visually interpret why an image is classified as fake. While classification metrics provide numerical validation, forensic visualization provides qualitative evidence. Once these features are extracted, the system proceeds to evaluate classification performance using structured metrics such as the confusion matrix

The confusion matrix (See Figure 6) provides a comprehensive evaluation of the classification performance of the proposed model by comparing actual and predicted labels across all classes. It serves as a detailed representation of how well the model distinguishes between different categories of deepfake and real images. It should be noted that although the federated learning framework employs a Non-IID data distribution across participating clients during training, the confusion matrix is generated using a balanced test set consisting of 150 samples per class. This balanced evaluation protocol ensures an unbiased assessment of classification performance across all categories. The diagonal elements correspond to correctly classified samples, whereas the off-diagonal elements represent misclassifications. As evident from the matrix, most samples are concentrated along the diagonal, indicating high classification accuracy and strong discriminative capability of the proposed model across all seven classes. The matrix is defined as:

$$CM = [C_{ij}] \quad (35)$$

$$Accuracy = \frac{\sum_i C_{ii}}{\sum_i \sum_j C_{ij}} \quad (36)$$

$$Misclassification Rate = \frac{\sum_{i \neq j} C_{ij}}{\sum_i \sum_j C_{ij}} \quad (37)$$

where C_{ij} represents the number of samples belonging to class i that are predicted as class j . The diagonal elements correspond to correct classifications, while off-diagonal elements indicate errors. From the obtained matrix, it is evident that the majority of values are concentrated along the diagonal, indicating that most samples are correctly classified.

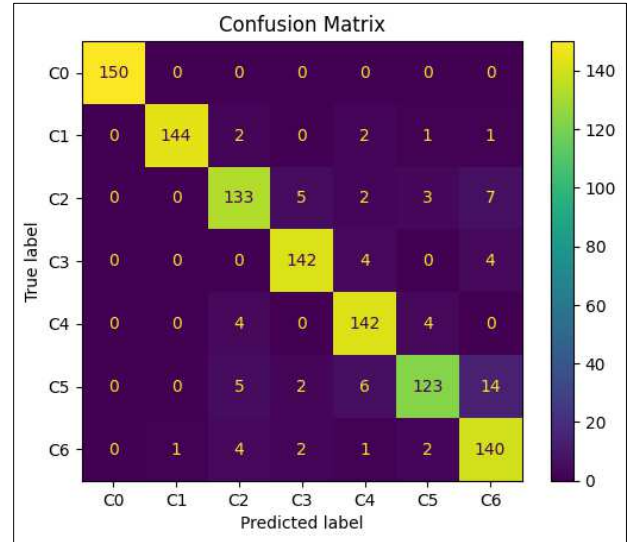


Figure 6. Represents confusion matrix of the proposed model.

This reflects the strong learning capability of the ResNet-18 architecture in capturing deepfake-specific features. The minimal presence of off-diagonal values suggests that the model rarely confuses one class with another, even when dealing with multiple manipulation techniques. This metric is particularly important in the context of FL, as it demonstrates that the model maintains high classification accuracy despite being trained on distributed data. It confirms that the aggregation process effectively combines knowledge from multiple clients without degrading performance. Furthermore, the confusion matrix helps identify specific classes where errors occur, providing insights into potential areas for improvement. For instance, slight confusion between similar manipulation types may be observed due to overlapping visual characteristics. Such analysis is valuable for refining the model and improving robustness. The confusion matrix establishes a strong foundation for evaluating classification performance. However, it does not provide an overall summary of prediction correctness, which is addressed in the next metric through accuracy distribution.

The accuracy distribution provides a simplified yet effective representation of the overall performance of the model by showing the proportion of correct and incorrect predictions. Figure 7 and Figure 8 represents the metric gives a quick overview of how well the model performs across the entire dataset. It is calculated using the following expressions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (38)$$

$$Error Rate = \frac{FP + FN}{TP + TN + FP + FN} \quad (39)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (40)$$

where TP , TN , FP , and FN represent true positives, true negatives, false positives, and false negatives respectively.



Figure 7. Represents the accuracy graph of the proposed model

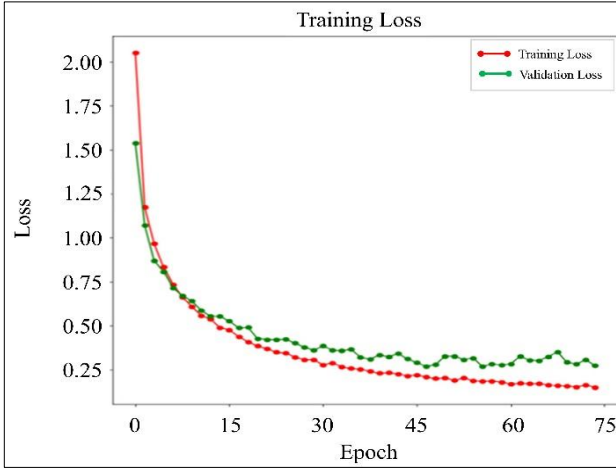


Figure 8. Represents the loss graph of the proposed model

The result shows that approximately 97.2% of predictions are correct, while only 2.8% are incorrect, indicating a high level of accuracy. This demonstrates that the proposed model effectively distinguishes between real and manipulated images, even under challenging conditions. The high accuracy can be attributed to several factors, including the use of the ResNet-18 architecture for feature extraction, the FL framework for capturing diverse data patterns, and the incorporation of FedProx optimization for stable training. Together, these components contribute to improved generalization and robust performance shown in Figure 9.

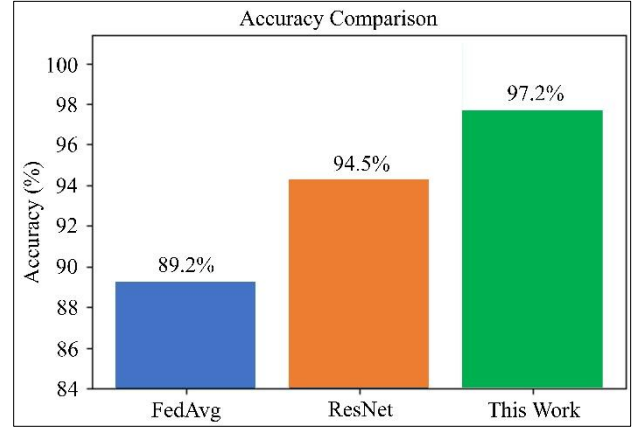


Figure 9. Represents the accuracy comparison of the proposed model

This metric provides an overall summary of performance but does not capture class-specific variations or error patterns. Therefore, further analysis is required to understand how errors are distributed across different classes, which is addressed in the next section.

Figure 10 presents the class-wise error distribution using a stem plot, where each stem corresponds to the number of misclassified samples in a particular class. The error count for each class is obtained from the confusion matrix by summing the off-diagonal elements of the corresponding row. This representation provides a direct visualization of how classification errors are distributed across different categories. From the figure, Class C0 exhibits zero misclassifications, indicating perfect classification performance for that category. Classes C1, C3, C4, and C6 show relatively low error counts of 6, 8, 8, and 10 samples, respectively, demonstrating strong recognition capability. Class C2 records 17 misclassified samples, suggesting moderate difficulty in distinguishing this category from similar classes. The highest error count is observed for Class C5 with 27 misclassified samples, indicating that samples belonging to this class share characteristics with other classes and are comparatively more challenging to classify accurately. Overall, the stem plot confirms that classification errors are concentrated in only a few categories, while most classes achieve high recognition performance, highlighting the robustness of the proposed deepfake detection framework. The error for each prediction is mathematically defined as:

$$Error_c = N_c - TP_c \quad (41)$$

$$Total\ Error = \sum_{i=1}^N E_i \quad (42)$$

$$Mean\ Error = \frac{1}{N} \sum_{i=1}^N E_i \quad (43)$$

N_c is the total number of samples in class c , TP_c is the number of correctly classified samples of class c . where y_i represents the actual label, $\hat{y}_i$ is the predicted label, and N is the total number of samples.

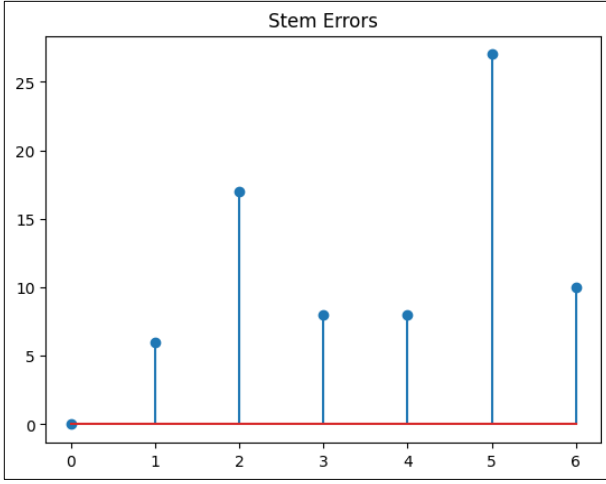


Figure 10. Represents the stem error

This naturally leads to the need for evaluating how accurately the model predicts positive instances, which is captured through precision analysis in the next metric.

The precision (See Figure 11) analysis is represented using a radar plot, which provides a class-wise visualization of how accurately the model predicts positive instances.

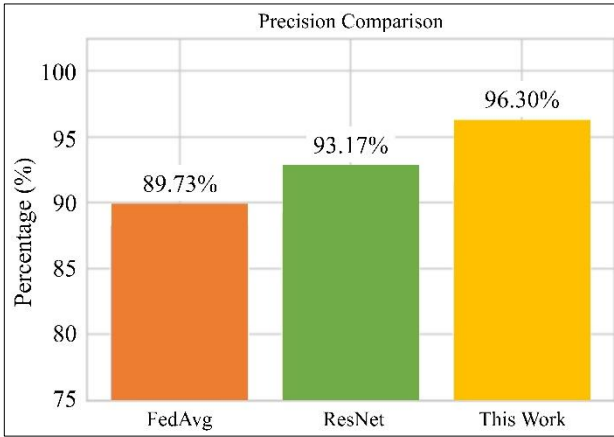


Figure 11. Represents the radar plot of the proposed precision metric.

Precision is a critical metric in deepfake detection, as it measures the proportion of correctly identified fake images among all images predicted as fake. High precision indicates that the model produces very few false positives, which is essential in applications where incorrect classification can lead to misinformation or false accusations. Precision is mathematically defined as:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (44)$$

$$Macro\ Precision = \frac{1}{K} \sum_{i=1}^K Precision_i \quad (45)$$

$$Weighted\ Precision = \sum_{i=1}^K w_i \cdot Precision_i \quad (46)$$

where TP_i and FP_i represent true positives and false positives for class i , K is the number of classes, and w_i denotes the weight of each class based on its sample size. The radar plot shows that precision values are consistently high across all classes, forming a nearly uniform shape. This indicates that the model performs reliably regardless of the type of deepfake manipulation.

The balanced distribution also suggests that the model is not biased toward specific classes, which is a common issue in classification tasks with uneven data distributions. The high precision achieved by the model can be attributed to the effectiveness of the ResNet-18 architecture in capturing fine-grained features, as well as the FL framework that enables learning from diverse data sources. Figure 12 and 13 represents recall and f1-score of the proposed model.

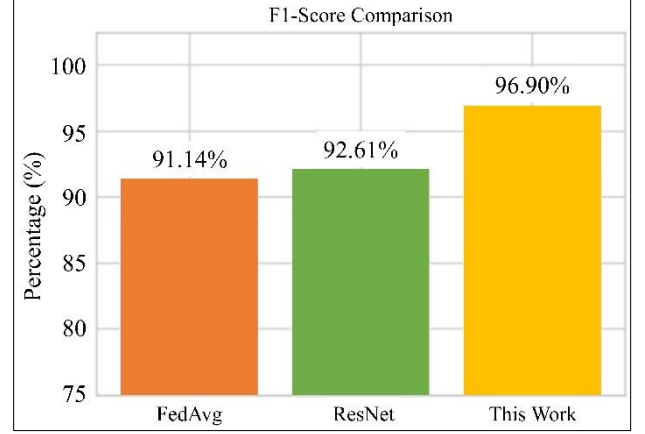


Figure 12. Represents the f1-score of the proposed model

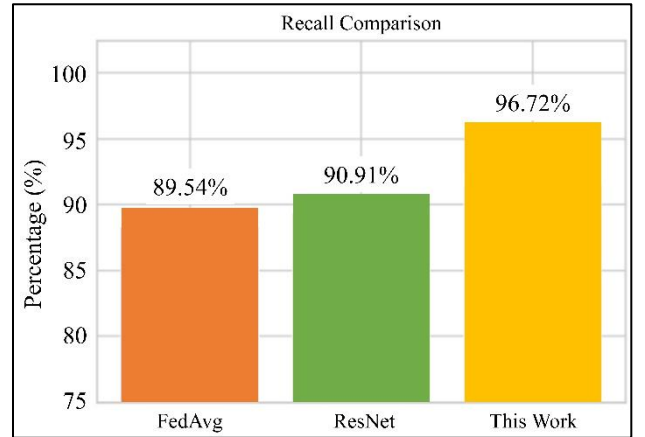


Figure 13. Represents the recall of the proposed model

Additionally, the use of FedProx optimization helps maintain consistency across client updates, further improving performance. This metric is particularly important in validating the robustness of the system, as it ensures that detected deepfakes are indeed correct. By analysing these spatial patterns, the system gains a deeper understanding of image structure, which enhances its ability to detect manipulations. With this final metric, the evaluation process demonstrates that the model effectively integrates multiple perspectives—spatial, statistical, and frequency-based—to achieve reliable and interpretable deepfake detection.

5. CONCLUSION

In this work, we present a privacy-preserving deepfake detection framework that combines FL with a lightweight ResNet-18-based convolutional architecture. We train and evaluate the proposed system on the FaceForensics++ dataset, which exposes the model to diverse manipulation techniques under realistic

compression settings. By carefully selecting representative classes and enforcing a balanced distribution of real and manipulated samples, the framework promotes consistent learning across different deepfake patterns. The ResNet-18 backbone is chosen to effectively capture subtle visual inconsistencies while maintaining computational efficiency suitable for distributed environments. In addition, we incorporate the FedProx optimization method to explicitly mitigate the effects of client-side data heterogeneity and to enable stable, reliable convergence of the global model. Experimental results show that the proposed approach achieves strong performance in distinguishing authentic from manipulated content, as reflected by accuracy of 97.20%, precision 96.30%, recall of 96.72, f1-score of 96.90%. The integration of forensic-oriented explainability tools, including edge-artifact visualization and residual-noise analysis, further enhances the interpretability of the predictions. The framework is well suited to real-world scenarios in which data is inherently distributed and cannot be centrally shared.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The authors received no financial support for the research, authorship, and publication of this article.

AUTHOR CONTRIBUTIONS

Pochampally Chandra Sekhar Reddy: Conceptualization, methodology, software, data curation, formal analysis, investigation, implementation, validation, visualization, writing---original draft preparation.

Kongara Srinivasa Rao: Supervision, conceptualization, methodology, validation, review and editing, project administration.

DATA AVAILABILITY

The datasets used in this study are publicly available from their respective official sources. The processed data generated during the current study are available from the corresponding author upon reasonable request.

ETHICS, CONSENT TO PARTICIPATE, AND CONSENT TO PUBLISH

Ethics approval, consent to participate, and consent to publish are not applicable to this study, as the research did not involve human participants, human data, human tissue, or animals.

CONFLICT OF INTEREST

The authors declare that they have no competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

REFERENCES

1. Thakur, G., & Dutta, K. (2026). Computational and Deep Learning Approaches for Video Analysis: An Engineering Perspective on Methods, Challenges, and Future Directions. *Archives of Computational Methods in Engineering*, 1-56.
2. Reza, M. S., Elias, F., Mahmud, M. I., & Ahmed, N. (2026). Attention and residual mechanism-based CNN architecture (ARC-Net) with enhanced fairness generalization for deepfake facial image detection. *Plos one*, 21(4), e0340099.
3. Shah, E. H., Dubey, U., Rahiman, A. S. A., & Shetty, N. P. (2026). Multi-layer knowledge distillation with custom temperature scaling for deepfake detection. *IEEE Access*.
4. Gupta, S., Raj, A. S. A., & Thiyagu, T. M. (2026). Federated micro-expression mining and multi-modal metadata fusion for deepfake fraud detection in ubiquitous financial video-KYC systems at IoT network. *Franklin Open*, 100523.
5. Wu, Y., Huang, H., Li, Z., & Zhang, S. (2025). CBAM-ResNet: A lightweight ResNet network focusing on time domain features for end-to-end deepfake speech detection. *Electronics*, 14(12), 2456.
6. Li, Y. (2025). Face forgery detection based on ResNet networks. In *Proceedings of the IEEE International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 127–131).
7. Fahad, M., Zhang, T., Iqbal, Y., Ikram, A., Siddiqui, F., Abdullah, B. Y., & Geng, Y. (2025). Advanced deepfake detection with enhanced ResNet-18 and multilayer CNN max pooling. *The Visual Computer*, 41(5), 3473–3486.
8. Kabir, M. H., Noman, A. A., Afq, A. A., Jaki, H., Hasan, M. N., Ahmad, ..., & Mojumder, I. A. (2025). A comprehensive review of machine learning techniques for deepfake detection. *Applied Computational Intelligence and Soft Computing*, 2025, 8883527.
9. Bhatnagar, A., Pandey, R. C., Vishal, V., & Singh, A. (2025). Hybrid federated deepfake detection via residual-aware temporal modeling and privacy-preserving learning. In *Proceedings of the IEEE International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)* (pp. 1–6).
10. Stephy Joy, D., & Thirumalai Selvi, R. (2025). FalsiDetect: A federated learning framework for generalizable deepfake detection across diverse datasets. In *Proceedings of the International Conference on Communication and Computational Technologies* (pp. 301–315). Springer.

11. Ha, J., El Azzaoui, A., & Park, J. H. (2025). FL-TENB4: A federated-learning-enhanced Tiny EfficientNetB4-Lite approach for deepfake detection in CCTV environments. *Sensors*, 25(3), 788.
12. Shao, R., Wu, T., & Liu, Z. (2025). Robust sequential deepfake detection. *International Journal of Computer Vision*, 133(6), 3278–3295.
13. Kumar, A., Singh, D., Jain, R., Jain, D. K., Gan, C., & Zhao, X. (2025). Advances in deepfake detection algorithms: Exploring fusion techniques in single and multi-modal approach. *Information Fusion*, 118, 102993.
14. Somoray, K., Miller, D. J., & Holmes, M. (2025). Human performance in deepfake detection: A systematic review. *Human Behavior and Emerging Technologies*, 2025(1), 1833228.
15. Sharma, V. K., Garg, R., & Caudron, Q. (2025). A systematic literature review on deepfake detection techniques. *Multimedia Tools and Applications*, 84(20), 22187–22229.
16. Li, M., Ahmadiadli, Y., & Zhang, X. P. (2025). A survey on speech deepfake detection. *ACM Computing Surveys*, 57(7), 1–38.
17. Harishmaa, M., Janani, S., Jayashree, K. A., Sherin, J. R., Manikandan, R., & Magesh, S. (2025). Blockchain and federated learning synergy for privacy-focused DeepFex solutions. In *Springer Nature Singapore* (pp. 1–22).
18. Kadhar, S. A., Rajakumari, S. B., & Priscilla, S. (2025). Federated learning architecture towards multimodal fake news detection in social media networks. In *Proceedings of ICCAMS* (pp. 1–5). IEEE.
19. Liu, L., Fan, K., & Yang, M. (2023). Federated learning: A deep learning model based on ResNet18 dual path for lung nodule detection. *Multimedia Tools and Applications*, 82(11), 17437–17450.
20. Soudy, A. H., Sayed, O., Tag-Elser, H., Ragab, R., Mohsen, S., Mostafa, T., & Slim, S. O. (2024). Deepfake detection using convolutional vision transformers and convolutional neural networks. *Neural Computing and Applications*, 36(31), 19759–19775.
21. Lanzino, R., Fontana, F., Diko, A., Marini, M. R., & Cinque, L. (2024). Faster than lies: Real-time deepfake detection using binary neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3771–3780).
22. Heidari, A., Navimipour, N. J., Dag, H., Talebi, S., & Unal, M. (2024). A novel blockchain-based deepfake detection method using federated and deep learning models. *Cognitive Computation*, 16(3), 1073–1091.
23. Ju, Y., Hu, S., Jia, S., Chen, G. H., & Lyu, S. (2024). Improving fairness in deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 4655–4665).
24. Wang, T., Liao, X., Chow, K. P., Lin, X., & Wang, Y. (2024). Deepfake detection: A comprehensive survey from the reliability perspective. *ACM Computing Surveys*, 57(3), 1–35.
25. Heidari, A., Navimipour, N. J., Dag, H., & Unal, M. (2024). Deepfake detection using deep learning methods: A systematic and comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 14(2), e1520.
26. Ashani, Z. N., Ilias, I. S. C., Ng, K. Y., Ariffin, M. R. K., Jarmo, A. D., & Zamri, N. Z. (2024). Comparative analysis of deepfake image detection method using VGG16, VGG19 and ResNet50. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 47(1), 16–28.
27. Borade, S., Jain, N., Patel, B., Kumar, V., Godhrawala, M., Kolaskar, S., & Shah, J. (2024). ResNet50 deepfake detector: Unmasking reality. *Indian Journal of Science and Technology*, 17(13), 1263–1271.
28. Yee, L. K., Hamid, I. R. A., ChaiWen, C., Abdullah, Z., Kipli, K., & Foozy, C. F. M. (2024). Deepfake image detection using ResNet50 model. In *Proceedings of CyberComp* (pp. 80–87). IEEE.
29. Yang, Z., Liang, J., Xu, Y., Zhang, X. Y., & He, R. (2023). Masked relation learning for deepfake detection. *IEEE Transactions on Information Forensics and Security*, 18, 1696–1708.
30. Wang, T., Cheng, H., Chow, K. P., & Nie, L. (2023). Deep convolutional pooling transformer for deepfake detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(6), 1–20.
31. Stroebel, L., Llewellyn, M., Hartley, T., Ip, T. S., & Ahmed, M. (2023). A systematic literature review on the effectiveness of deepfake detection techniques. *Journal of Cyber Security Technology*, 7(2), 83–113.
32. Kumar, M., & Sharma, H. K. (2023). A GAN-based model of deepfake detection in social media. *Procedia Computer Science*, 218, 2153–2162.
33. Gupta, G., Raja, K., Gupta, M., Jan, T., Whiteside, S. T., & Prasad, M. (2023). A comprehensive review of deepfake detection using advanced machine learning and fusion methods. *Electronics*, 13(1), 95.
34. Liu, D., Dang, Z., Peng, C., Zheng, Y., Li, S., Wang, N., & Gao, X. (2023). FedForgery: Generalized face forgery detection with residual federated learning. *IEEE Transactions on Information Forensics and Security*, 18, 4272–4284.

35. Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake detection: A systematic literature review. *IEEE Access*, 10, 25494–25513.
36. Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). Deepfake detection for human face images and videos: A survey. *IEEE Access*, 10, 18757–18775.
37. Almutairi, Z., & Elgibreen, H. (2022). A review of modern audio deepfake detection methods: Challenges and future directions. *Algorithms*, 15(5), 155.
38. Taeb, M., & Chi, H. (2022). Comparison of deepfake detection techniques through deep learning. *Journal of Cybersecurity and Privacy*, 2(1), 89–106.
39. Zhang, T. (2022). Deepfake generation and detection: A survey. *Multimedia Tools and Applications*, 81(5), 6259–6276.
40. Guarnera, L., Giudice, O., Guarnera, F., Ortis, A., Puglisi, G., Paratore, A., & Battiato, S. (2022). The face deepfake detection challenge. *Journal of Imaging*, 8(10), 263.
41. Wang, X., Zhao, Z., Zhang, C., Bai, N., & Hu, X. (2022). SE-ResNet56: Robust network model for deepfake detection. In *International Workshop on Digital Watermarking* (pp. 37–52). Springer.
42. Zhou, T., Wang, W., Liang, Z., & Shen, J. (2021). Face forensics in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5778–5788).