ISSN: 2093-5374 / E-ISSN: 2093-5382

Deep Inception V4 Convolution Neural Network Towards Predictive Representation Learning Deepfake Contents in the Online Social Networks

S. Narmatha[1*], and Dr.S. Mythili[2]

[1*]Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. narmatha289@gmail.com, https://orcid.org/0009-0005-4766-9451

[2] Professor and Head, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. smythili78@gmail.com https://orcid.org/0000-0003-3196-6257

Received: October 27, 2025; Revised: December 22, 2025; Accepted: February 05, 2026; Published: March 31, 2026

Abstract

The rapid evolution of Generative Artificial Intelligence (AI) has significantly increased the prevalence of high-fidelity deepfakes, posing a severe threat to digital security and social media integrity. Existing detection frameworks often struggle with generalization and the identification of multi-scale spatio-temporal artifacts. This research addresses these challenges by proposing a robust detection system utilizing the Deep Inception V4 architecture integrated with Predictive Representation Learning. By leveraging multi-scale convolution kernels and specialized reduction blocks, the proposed model extracts both microscopic pixel-level inconsistencies and macroscopic structural anomalies commonly found in manipulated videos. The methodology was rigorously evaluated using the large-scale Deepfake Detection Challenge (DFDC) dataset. Experimental results demonstrate that the Inception V4 model achieves a Training Accuracy of 94.87% and a Validation Accuracy of 92.74%, representing a substantial improvement over baseline CNN (82.00%) and VGG-19 (80.00%) architectures. Statistical validation using a two-sample t-test yielded a p-value < 0.001, confirming the significance of these performance gains. Furthermore, the model achieved a Receiver Operating Characteristic (ROC-AUC) of 0.96, indicating high precision and recall in identifying 2,308 true positive cases within the test subset. These findings conclude that multi-scale feature extraction, combined with self-supervised pre-training, provides a superior defense against generative threats. Future work will investigate the integration of lightweight hybrid transformers to enhance real-time detection on mobile platforms while maintaining high classification sensitivity.

Keywords: Deepfake Detection, Deep Learning, Inception V4 CNN, Predictive Representation Learning, Social Media Security, Generative AI.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 1 (March - 2026), pp. 817-837. DOI: *10.58346/JOWUA.2026.I1.045*

*Corresponding author: Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education Coimbatore, Tamil Nadu, India.

1 Introduction

Recent advancements in content generation technologies using artificial intelligence and deep learning have led to large-scale social threats due to the propagation of Deepfake content (Almars, 2021) on online social media platforms. Consequently, a robust and reliable Deepfake detection method becomes a primary necessity for social media platforms. Further, it has gained significant attention among the research community to defend this fake content propagation. Deepfake (McCloskey & Albright, 2019) is a technology that uses artificial intelligence and deep learning to generate fake videos on social media platforms for various benefits, such as modifying images or synthesizing the image and speech components of other people. Familiar free open-source tools like Faceswap (Wang et al., 2020), Deepfaker (Hsu et al., 2018), and Deep FaceLab (Güera & Delp, 2018) alter the source person's face with the target person's face to generate a new video with the same action as the source with the target person's face. Results of the Deepfake AI faceswap are illustrated in figure 1.

(a) Source (b) Target (c) Result

Figure 1: Deepfake AI faceswap application results

In order to prevent the propagation of fake content on social media, many digital forensic machine learning models and deep learning models have been developed and employed as conventional post-processing approaches to detect fake content towards digital security (Sun et al., 2021). The main shortcoming of those approaches was their susceptibility to severe performance deterioration across different generalization types and non-resilience to low-level texture feature variations. Moreover, many variants of the convolution neural network model have been employed to process generalization and low texture feature variation on processing the image in a grid or sequence structure, but it is inflexible to capture irregularities and complex facial landmarks. Hence, it increases the necessity of advanced Deepfake detection approaches.

In this paper, a new predictive learning architecture using a deep inception V4 Convolution Neural Network is proposed to detect and prevent the propagation of Deepfake content in social media platforms, as it provides a high generalization guarantee to the detection approach, and it has a good understanding of local pixels and their interconnections. Deep Inception V4 Convolution Neural Network is pre-trained with self-supervised contrastive learning to produce an inception relevancy map, as it provides a better representation of the manipulated regions, capturing spatio-temporal context features of the contents, and post-processing perturbations. The inception model can extract abstract features and describe the latent patterns of the contents on both spatial and temporal content to provide improved recognition accuracy. A particular self-supervised pre-training approach uses a preprocessing step composed of transformation, normalization, and an augmentation method to enhance content quality. Preprocessed content is employed to a deep inception v4 Convolution Neural Model, which contains four inception modules composed of the convolution layer with different filter sizes, factorization methods, and batch normalization, a filter concatenation layer, a max pooling layer, an

average pooling layer, a reduction block, and a softmax classifier with loss function and label smoothing to eliminate the overfitting issues. Moreover, the inception model is capable of capturing the complex dependencies between distinct regions of the content on a wider network. A further classifier model is incorporated in the softmax function with fine-tuning of the learning model with RMSProp optimized to classify the context features and distinguish the fake contents from the original contents with high robustness and accuracy. Training and validation of the model is carried out using the video from the DFDC benchmark dataset.

The rest of the article has been structured as follows: Section 2 represents recent research using deep learning models to detect the Deepfake content propagation in the online social media platforms. The proposed Deep Inception v4 Convolution Neural Network, including Inception v4 modules for processing the deep fake content, is designed and outlined in Section 3. Experimental analysis and performance analysis, including the dataset, implementation setting, and performance metric for proposed approaches, are presented and discussed in comparison to the conventional approaches in Section 4. Finally, section 5 provides concluding remarks on the contribution of this research.

2 Related Works

In this part, numerous conventional approaches using machine learning and deep learning architectures to detect deepfake content in videos have been detailed as follows.

2.1 Graph Convolution Network for Deepfake Detection

In this literature, the Graph Convolution Network is used for Deepfake Detection. In this, the input data is projected as graph data with nodes representing the different regions of the image. It is considered the eyes, ears, nose, and mouth in the image, and the edge represents the association among these regions of the image. Graph Convolution learns the local relationships among the nodes in the graph structure. Those graph convolutions generate the feature map, composing the complex features of the different facial landmarks of the image (Aggarwal et al., 2021; Dheeraj et al., 2021; Li et al., 2021).

A fully connected layer generates the Saliency map, which represents the importance of the individual regions to the class label on fine-grained analysis. Fine-grained analysis is considered a decision-making process that allows for better visualization of the complex feature in the significant region of the image, as it detects Deepfakes accurately. The model spans between various distributions of the contents of the resultant and source images.

2.2 Transformer Model with Aggregation and Cross Attention Blocks for Deepfake Detection

In this literature, a transformer model is employed, incorporating the aggregation and cross-attention blocks for Deepfake detection in content propagating on social media. The transformer model aggregates global and local features. Feature Aggregation blocks contain the aggregate features that represent the fake (tampered contents) and real contents. Further cross attention block process aggregated features with attention blocks to reduce redundant features with respect to frequency cues and similarity maps (Al-Dhabi & Zhang, 2021). Finally, multi-headed clustering is presented to cluster the semantic features of the attention blocks. A cluster of real and fake content is attained with its respective frequency cue in the attention blocks. Clustering is carried out using the K nearest Neighbours approach, termed as the semantic objective function, with different spectral class centres to process the semantic features with pixel-wise relations within spectral features.

2.3 Inference from Literature

The analysis of recent literature reveals that while Graph Convolutional Networks (GCNs) successfully model relational facial landmarks and Transformers capture global context, both suffer from high computational costs and a tendency to overfit to specific dataset compression artifacts. Conventional CNN models like VGG-19 and MesoNet are efficient but often fail to generalize across different Deepfake generation engines (e.g., Faceswap vs. DeepFaceLab).

This research addresses these gaps by proposing a Deep Inception V4 architecture. Unlike standard CNNs, the multi-scale kernels in Inception modules allow for the simultaneous capture of both microscopic pixel-level anomalies and macroscopic structural inconsistencies. Furthermore, by incorporating Predictive Representation Learning, this study shifts the focus from simple binary classification to learning the latent spatio-temporal patterns of manipulation, resulting in a more robust and generalizable detection system for real-world social media environments.

3 Proposed Methodology

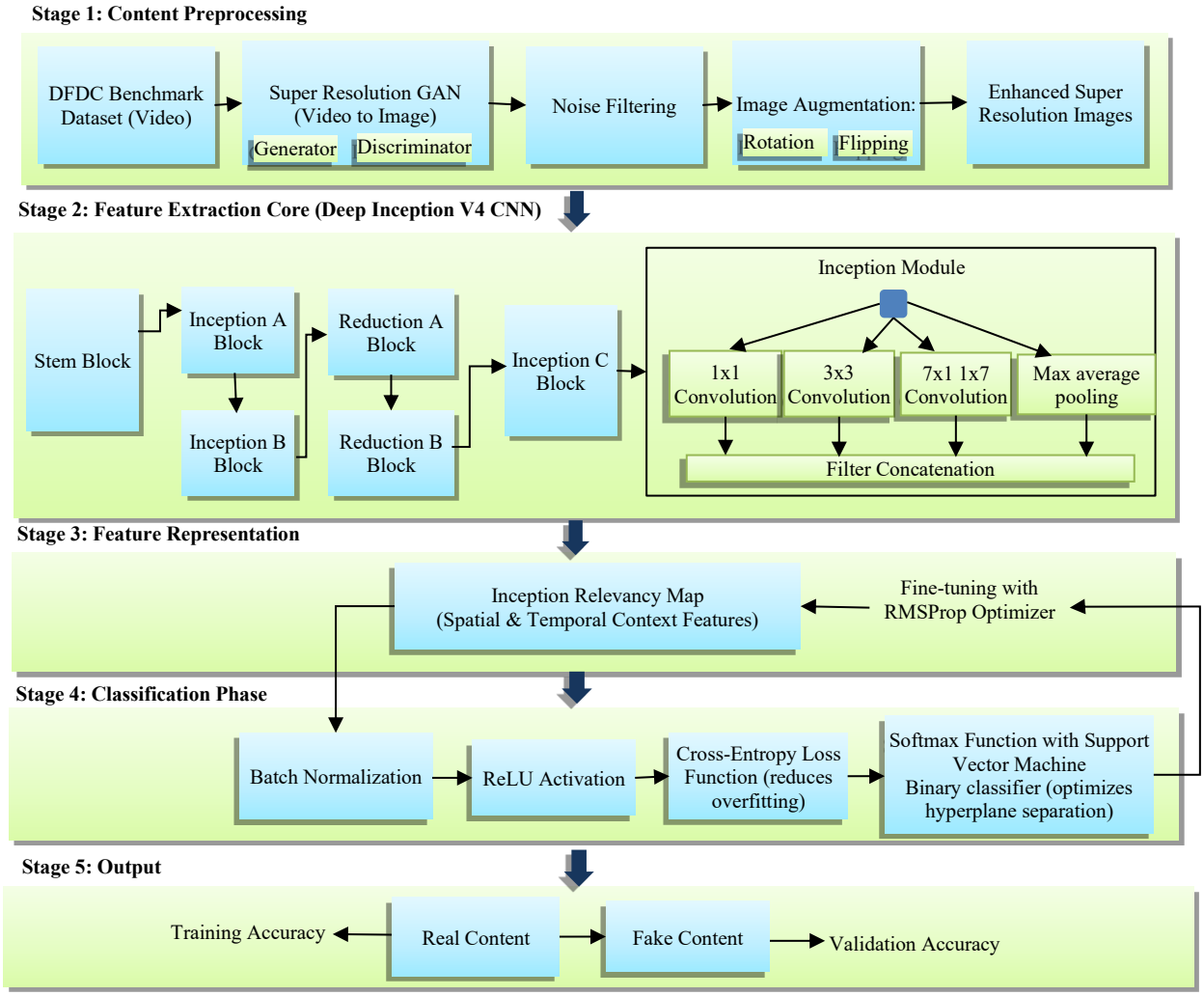


Figure 2: Architectural framework of the proposed deep inception V4 deepfake detection model

The proposed design of the self-supervised pre-trained predictive and contrastive learning architecture using a deep Inception V4 Convolution Neural Network to detect and prevent the propagation of Deepfake content in social media platforms is represented by elaborating on the major functional steps of the model, as follows.

Figure 2 illustrates the proposed detection pipeline, beginning with SRGAN-based preprocessing and augmentation. The core features are extracted through a Deep Inception V4 architecture, utilizing multi-scale kernels. These are concatenated into an Inception Relevancy Map, which is then classified using an SVM-optimized Softmax layer with RMSProp for high-accuracy Deepfake identification.

3.1 Content Preprocessing

Content preprocessing is used to enhance the content quality of the Deepfake content by utilizing video-to-image frame transformation and noise filtering using GAN-based transformation (Badale et al., 2018), data normalization, and data augmentation methods.

Super Resolution Generative Adversarial Network - Video to Image Frame

Generative Adversarial Network-based transformation is used to transform the Deepfake video content into image super-resolution frames by eliminating the image noise. In this process, the GAN model is composed of two components: the generator and the discriminator. A generator is used to generate the image frames from an image in the distribution of the video content through the generator component. Discriminators are used to eliminate the noise from the generated image frame distributions. Figure 3 represents the video-to-image transformation.

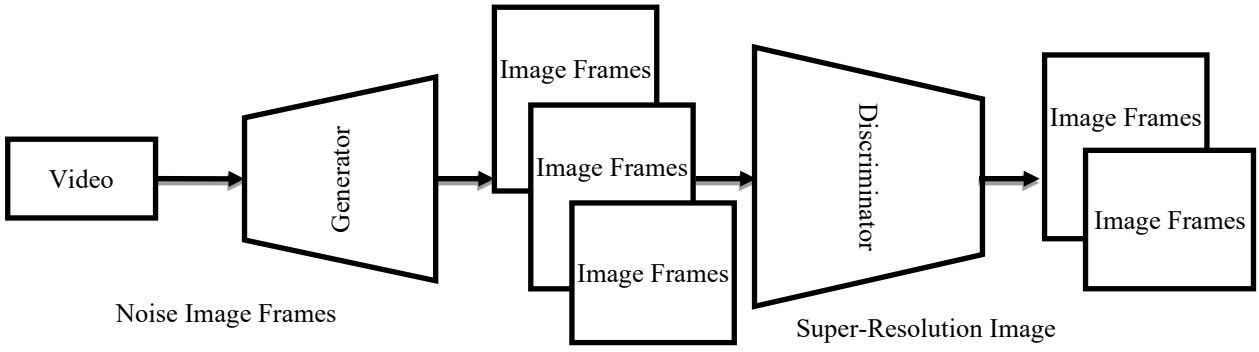


Figure 3: Generative adversarial network to video-super resolution images

3.2 Image Augmentation

Image Augmentation is used to increase the generalization ability of the image by increasing the size of the dataset to provide a good understanding of local pixels and their interconnections. It is achieved by applying the following augmentation methods:

Rotation: It introduces the variation by rotating the images in circular movement of the object, which is considered on the basis of the axis of rotation (Li et al., 2018).

Flipping: It processes the image horizontally to increase the size of the dataset, and vertically flipping increases the accuracy of the image (Agarwal et al., 2019).

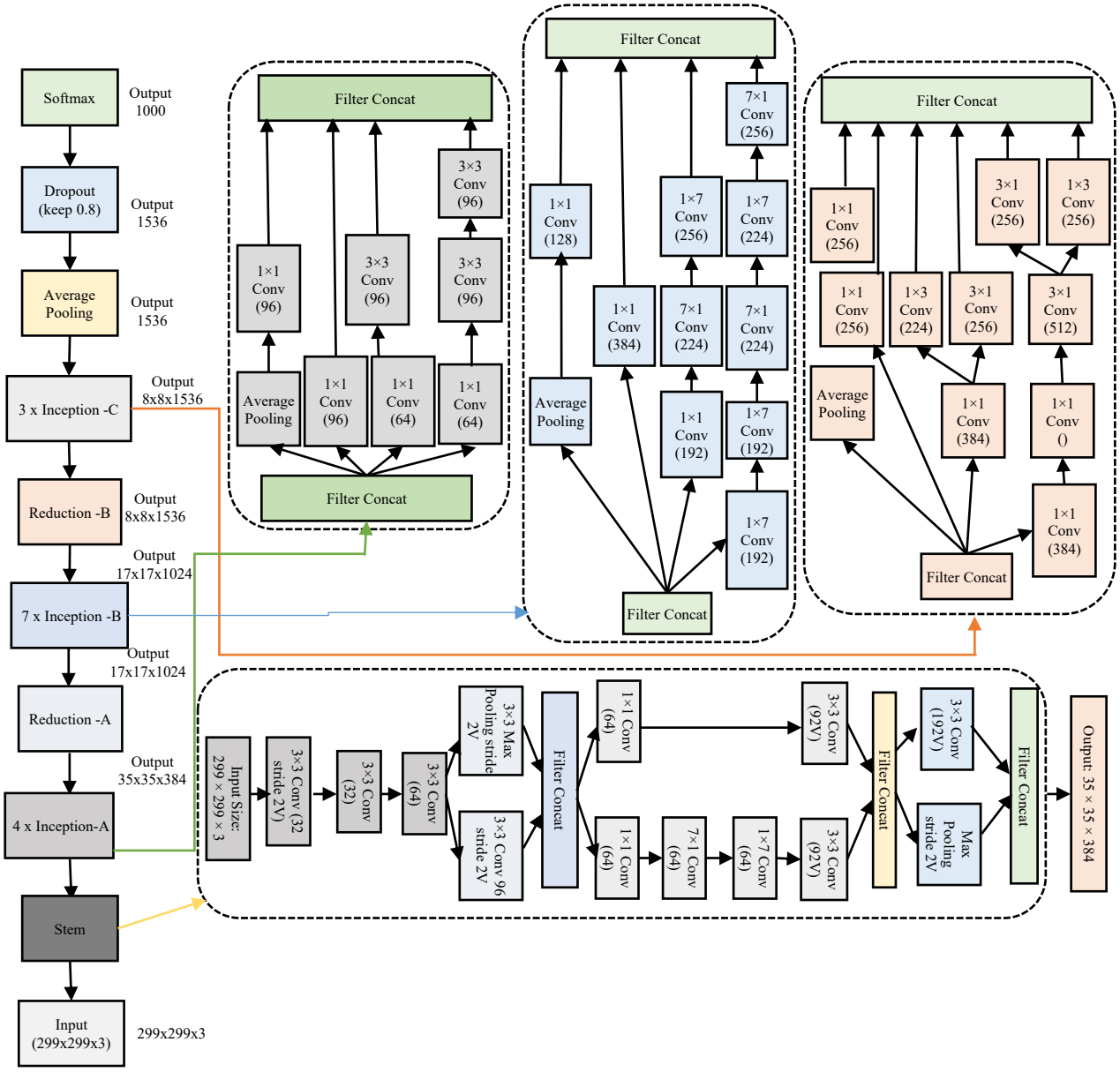


Figure 4: Inception V4 modules

3.3 Deep Inception V4 Convolution Neural Network

Deep Inception V4 Convolution Neural Network is pre-trained with self-supervised contrastive learning, which contains four inception modules. Composed of the convolution layer with different filter sizes, factorization methods, and batch normalization, a filter concatenation layer, a max pooling layer, an average pooling layer, a reduction block, and a softmax classifier with a loss function and label smoothing. Inception V4 module composed of blocks including stem block, Inception A block, Inception B block, Inception C block, as well as reduction block, as represented in figure 4.

Stem Block

The stem block is the first block of the Inception V4 Network in which the preprocessed input is transferred to a set of convolution layers, max pooling layers, filter concatenation layers, and output layers to extract the complex features of spatio-temporal context features of the images (Tsujii, 2021). Hyperparameter tuning of the Deep Inception v4 Convolution Neural Network is mentioned in table 1.

Table 1: Hyperparameter tuning

Hyperparameter	Value
Learning rate	10^{-6}
Epoch Value	50
Block size	8*8
No of Inception Modules	4
Kernel size of Stem Module	1*1, 3*3,3*1,7*1,1*7
Kernel size of Inception A Module	1*1,3*3
Kernel size of Inception B Module	1*1,7*1,1*7
Kernel size of Inception C Module	1*1,3*3,3*1,1*3
Activation function	ReLu
Softmax Function	Naive Bayes
Loss Function	Cross Entropy
Optimizer	RMSProp
Epoch value	100
Batch size	128

Sub Block 1 of Stem

Convolution Layer: In this Convolution layer, Convolution is performed with 3 different sizes of filter: 3*3 convolution with 149*149*32 (32 kernel stride 2V), 3*3 convolution with 147*147*32 (32 kernel Stride), 3*3 convolution with 147*147*64 (64 kernel Stride). Convolution of the image produces an image matrix A, which is composed of abstract features, represented as in equation (1), Image Matrix A on Convolution Operation.

$$\text{Axy} = \sum_{u=1}^{l}\sum_{v=1}^{l} k_{uv} c_{i(x,u),j(y,v)} \rightarrow (1)$$

In this operation, c represents convolution and k represents the kernel.

Max pooling layer: In this max pooling layer, the stem block is further partitioned into two branches. The initial branch is a max pooling layer, which has a kernel size of 96 with a stride of 2V, and the next branch is a convolution branch, which has a stride of 2V.

Output layer - Filter Concatenation layer: It is considered a filter concatenation layer of block 1, which aggregates the other branch details containing convolution layer image sizes and max pooling layer image size to produce the output with an image size of 73*73*160 dimensions.

Sub Block 2 of Stem

In block 2, the output image of specified dimension splits into four branches containing only a convolution layer of various sizes, which extract the latent features of the images.

Convolution Layer: In the first branch process with two convolution layers, 1*1 Convolution (64 kernel stride), second branch process with 7*1 Convolution (64 kernel stride), third branch process with 1*7 Convolution (64 kernel stride), and fourth branch with two 3*3 Convolution (96 Kernel stride) and 3*3 Convolution (96 Kernel stride V).

Output layer: All convolution branch outputs will be concatenated in the output layer of block 2 to obtain the image size of 71*71*192 dimensions, which features the embedding extracted by different layers of the block.

Sub Block 3 of Stem

In block 3, the output image of specified dimension from block 2 is split into a convolution and max pooling layer in a single branch with different sizes, which feature embedding and extracted by different layers of the block are represented as a feature vector.

Convolution Layer: In this convolution layer of the single branch of the block 3, 3*3 Convolution (192 kernel stride) operation is carried out using the proximity criteria to compute the association among the pixels of the image with all neighboring pixels

Max Pooling Layer: In this max layer of the single branch of block 3, max pooling (2V stride) operation is carried out. It computes high-level features using an adjacency matrix.

Output Layer: Convolution layer outputs and max pooling layer outputs are concatenated in the output layer of block 3 to obtain the image size of 35*35*384 dimensions as high-level features on the structural relationship of pixels.

3.4 Inception Block

Inception A Block

The inception block obtains the input from the stem block using concatenation functions. Obtained input propagates in three branches, which are as follows,

Branch 1 of Inception A Block

Branch 1 of the inception A block is composed of both a convolution layer and an average pooling layer to carry out the feature extraction operation.

Convolution Layer: In this branch, the convolution layer is composed of 1*1 Convolution (64 kernel), 1*1 Convolution (64 kernel), and 1*1 Convolution (96 kernel).

Average Pooling Layer: It is composed of a single average pooling layer that produces a 96-kernel size by computing the structural dependency of the pixel on various image regions.

Branch 2 of Inception A Block

Branch 2 of the inception A block is composed of only a convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, the convolution layer is composed of 1*1 Convolution (96 kernel), 3*3 Convolution (96 kernel), and 3*3 Convolution (96 kernel) as a downstream task.

Branch 3 of Inception a Block

Branch 3 of the Inception A block is composed of only one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of only one 3*3 Convolution (96 kernel) has been carried out.

Output layer: Branch outputs are concatenated in the output layer of the inception block using a filter concatenation function to obtain the image size of 35*35*384 dimensions.

Inception B Block

Inception B block obtains the input from the Inception block A using concatenation functions. Obtained input propagates in five branches, which are as follows.

Branch 1 of Inception B Block

Branch 1 of the inception B block is composed of only one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of only one 1*1 Convolution (192 kernel) has been carried out.

Branch 2 of Inception B Block

Branch 2of the Inception B block is composed of both a convolution layer and an average pooling layer to carry out the feature extraction operation.

Convolution Layer: In this branch, the convolution layer is composed of 1*1 Convolution (192 kernel) and 1*7 Convolution (192 kernel).

Average Pooling Layer: It is composed of a single average pooling layer that produces a 128-kernel size.

Branch 3 of Inception B Block

Branch 3 of the inception B block is composed of only a convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, the convolution layer is composed of 1*1 Convolution (384 kernel), 7*1 Convolution (224 kernel), and 7*1 Convolution (224 kernel).

Branch 4 of Inception B Block

Branch 4 of the Inception B block is composed of only a convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, the convolution layer is composed of 1*1 Convolution (128 kernel), 7*1 Convolution (256 kernel), and 7*1 Convolution (224 kernel).

Branch 5 of Inception B Block

Branch 5 of the Inception B block is composed of only one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of only one 7*1 Convolution (256 kernel) has been carried out.

Output Layer: Branch outputs are concatenated in the output layer of the inception B block to obtain the image size of 17*17*1024 dimensions.

Inception C Block

Inception C block obtains the input from the Inception block B using concatenation functions. Obtained input propagates in multiple branches, which is as follows.

Branch 1 of Inception C Block

Branch 1 of the Inception C block is composed of only one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of only one 1*1 Convolution (384 kernel) has been carried out.

Branch 2 of Inception C Block

Branch 2 of the Inception C block is composed of one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of a 1*3 Convolution (448 kernel) is only carried out to extract the feature.

Branch 3 of Inception C Block

Branch 3 of the Inception C block is composed of one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of a 3*1 Convolution (512 kernel) is only carried out to extract the feature.

Branch 4 of Inception C Block

Branch 4 of the Inception C block is composed of one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of a 1*3 Convolution (256 kernel) is only carried out to extract the feature.

Branch 5 of Inception C block

Branch 5 of the Inception C block is composed of one convolution layer to carry out the feature extraction operation.

Convolution Layer: In this branch, a convolution layer composed of a 3*1 Convolution (256 kernel) is only carried out to extract the feature.

Auxiliary Branches of Inception C Block

Auxiliary Branches of Inception C block, composed of four convolution layers and an average pooling layer to carry out the feature extraction operation, are as follows.

Convolution Layer: In this branch, a convolution layer composed of 1*1 Convolution (384 kernel), 3*1 Convolution (256 kernel), 1*3 Convolution (256 kernel), and 1*1 Convolution (256 Kernel) is carried out to extract the feature.

Average Pooling Layer: In this branch, a single average pooling layer produces an image of kernel size 256.

Output Layer: Branch outputs are concatenated in the output layer of the inception C block to obtain the image size of 8*8*1536 dimensions.

Fully Connected Network: Classification Phase

In this fully connected layer, output from the average pooling layer is processed with the softmax function, loss function, and optimizer— in this inception relevancy map composed of spatial and temporal features of the multiple fake and real images. Specifically, the ReLu activation function (Lee & Kim, 2021) is utilized to carry out the normalization of the Inception relevance map containing features with multiple contexts. It eliminates the non-linearity and over-fitting challenges of the relevance map.

Softmax Function

Softmax function containing a binary classifier, which is considered as a support vector machine, is employed to normalize the inception relevancy map. In this feature map, the feature weight is distributed in a hyperplane to estimate the feature weight on the basis of support and confidence measures. It uses the Lagrange function (Pan et al., 2020) as part of the weight computation to increase the separation of the feature. The oftmax function of the support vector machine is mentioned in equation (2) as follows,

$$D(x) = agr_{max \sum_{n=0}^{1} x(i) w[f(u,v)]} \qquad (2)$$

The fake Content feature is mentioned with a class label on the basis of the weight value as a contrastive learning model. It prioritizes the weight towards the classification of the content. The highest weight of the feature of the content segment represents fake content, and the lowest weight of the features of the content represents the real image. The eight function (Fung et al., 2021) eliminates the overfitting and underfitting issues. The weight function is represented as in equation (3),

$$w = \sum_{i=1}^{n} d_i xi \qquad (3)$$

Finally, model performance is estimated through Euclidean distance to the obtained fake and real content classes, as it reduces the loss and vanishing gradient bottlenecks. Further fine-tuning of the learning model with RMSProp (Rafique et al., 2021) optimizes the classifier. It classifies the context features and distinguishes the fake contents from the original contents with high robustness and accuracy.

Algorithm 1: Deep Inception V4 Convolution Neural Network Deepfake Classifier

Input: DFDC Dataset

Output: Classes of Deepfake contents and Original Content

Process

Generative Adversarial Network ()

Generator (Video to Image)

Image frames

Discriminator (Image frames)

Super Resolution Images frames

Image Augmentation ()

Rotation ()

Increases Generalization Ability

Flipping ()

Increase the Training Data Size by analyzing local pixels and interconnections.

Deep Inception V4 Convolution Neural Network ()

Stem Block ()

Spatial and Contextual features of the image ()

Convolution Layer ()

Multiple Kernel Size of Convolution ()

Average Pooling ()

Feature Embedding ()

Feature map ()

Inception A block (feature Map)

Downstream task ()

Convolution Layer ()

Multiple Kernel Size of Convolution ()

High-level features ()

Structural dependency of pixel on various regions ()

Feature Embedding ()

Inception relevancy map ()

Inception B Block ()

Downstream task ()

Convolution Layer ()

Multiple Kernel Size of Convolution ()

High level features ()

Structural dependency of pixel on various regions ()

Feature Embedding ()

Inception relevancy map ()

Inception C block ()

Downstream task ()

Convolution Layer ()

Multiple Kernel Size of Convolution ()

High level features ()

Structural dependency of pixel on various regions ()

Feature Embedding ()

Inception relevancy map ()

Drop out layer ()

Fully Connected Layer ()

Batch Normalization ()

It eliminates the Nonlinear feature

Activation Function (ReLu)

It orders the linear feature

Loss Function (Cross Entropy)

Eliminate the overfitting and vanishing gradient bottlenecks

Optimizer (RMSProp)

Softmax function ()

Support Vector Machine ()

Hyperplane (Distribution of Normalized Features)

Class (Real | Fake)

Algorithm 1 presents a multi-stage framework that transforms raw video into high-resolution frames using SRGAN and extracts multi-scale features via Inception V4 blocks. These features are concatenated into a relevancy map and classified using an SVM-optimized Softmax layer to distinguish between real and manipulated content.

The proposed model identifies manipulated regions by calculating the feature importance within the Inception Relevancy Map. he convolution operation within the Inception modules is defined as in equation (4):

$$A_{xy} = \sigma\left(\sum_{i}\sum_{j} K_{ij} \cdot I_{(x+i)(y+j)} + b\right) \qquad (4)$$

Where K is the kernel, I is the input image, and σ represents the ReLU activation. To distinguish fake content, the study employs a Support Vector Machine (SVM) within the Softmax layer. The decision boundary is determined by the hyperplane in equation (5):

$$f(x) = sgn\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right) \qquad (5)$$

Where α are Lagrange multipliers and $K(x_i, x)$ is the kernel function that projects the feature into a higher-dimensional space to maximize the margin between real and fake classes. The model is optimized by minimizing the Cross-Entropy loss H as shown in equation (6):

$$H(p,q) = -\sum_{x} p(x) log\, q(x) \qquad (6)$$

Where p(x) is the true label and q(x) is the predicted probability. The above mathematical representation formalizes the link between Inception features and SVM classification.

4 Experimental Results

Experimental analysis is conducted using the DFDC popular benchmark dataset, which consists of 19000 real videos from 400 actors and 1lakh fake videos. Experiments were implemented using a Python environment in Pytorch and Tensorflow libraries (Jaiswal, 2021) to evaluate the robustness and effectiveness of the proposed self-supervised deep inception v4 Convolution Neural Network-based contrastive learning. The dataset is partitioned into training, validation, and testing sets in a ratio of 6:2:2. Description of the dataset and the performance metric for performance analysis are mentioned below.

4.1 Implementation Environment

The proposed model was implemented using the Python 3.10 environment. Deep learning tasks were executed using PyTorch 2.1 and TensorFlow 2.13, while image processing and data manipulation were handled via OpenCV 4.8 and NumPy 1.24. All experiments were conducted on a system equipped with an NVIDIA RTX 3090 GPU (24GB VRAM) to ensure computational efficiency.

4.2 Dataset Description: DFDC Dataset

DeepFake Detection Challenge (DFDC) dataset (Rana & Sung, 2020) consists of a large quantity of videos containing human faces with labels that were generated using a facial manipulation technique. A dataset composed of various face images discriminated in gender, skin tone, age, lighting condition, and different face poses. The dataset is available to the research community for the development, testing, and analysis of techniques for detecting videos with manipulated faces. The manipulated face contains the video of face swaps.

4.3 Performance Metric

Performance analysis of the proposed Deepfake detection approach on the dataset, which was created using the face swapping technique and state-of-the-art approaches, was carried out using the following metrics,

Area Under Curve: AUC is considered the area under the ROC curve, and it is used to evaluate the accuracy of the classifier, which reveals the detection ability of the manipulated traces. It avoids the interference of the human factor (Suratkar et al., 2020).

The ROC curve is constructed by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings in equation (7).

$$AUC = \int_0^1 TPR(FPR)\ dFPR \quad (7)$$

Precision: It is employed to compute the probability of the classifier being correctly classified on its facial feature, and it is used to estimate the similar facial features in the particular fake class. It is computed as follows in equation (8),

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (8)$$

Precision computes the detection ability of the Deepfake model, which identifies the fake facial feature in the image. The confusion matrix of the DFDC dataset on its validation data is represented in figure 5.

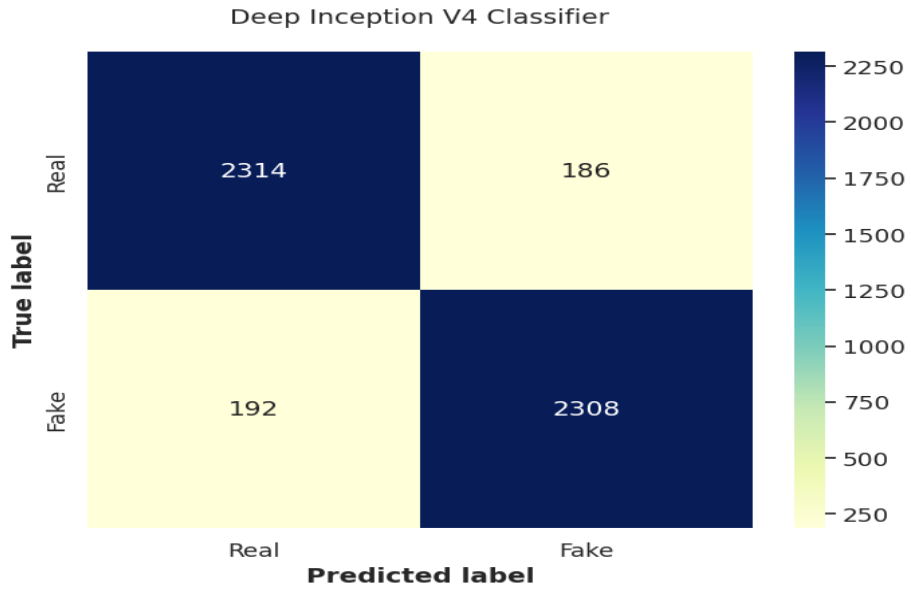


Figure 5: Confusion matrix of the DFDC dataset's validation data

The confusion matrix (El Rai et al., 2020) generates the true positive, true negative, false positive, and false negative values. In this, a true positive mentions correctly classified fake facial features, a false positive mention incorrectly classified fake facial features, a false negative is incorrectly classified as a non-significant fake facial feature, and a true negative is correctly classified as a non-significant fake facial feature of the image.

Recall: It is employed to compute the probability of the classifier being incorrectly classified on its facial feature, and it is used to estimate the dissimilar facial features in the particular fake class. It is computed as follows in equation (9),

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Positve} \quad (9)$$

Recall computes the detection ability of the Deepfake model, which identifies the fake regions in the image effectively.

Accuracy: Accuracy, represented in equation (10), provides an overall measure of the classifier's performance by calculating the ratio of correct predictions (both real and fake) over the total number of cases.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

The above provided metrics quantify the model's ability to identify manipulated regions while minimizing false alarms, accurately, and missed detections through a standardized mathematical evaluation of the confusion matrix.

4.4 Performance Analysis

Inception modules are building blocks of the feature extractor, and the support vector machine is a classifier module used to classify the inception relevancy feature map. Figure 6 represents the performance analysis of the training and validation accuracy of the proposed Deepfake approach against different epoch values.

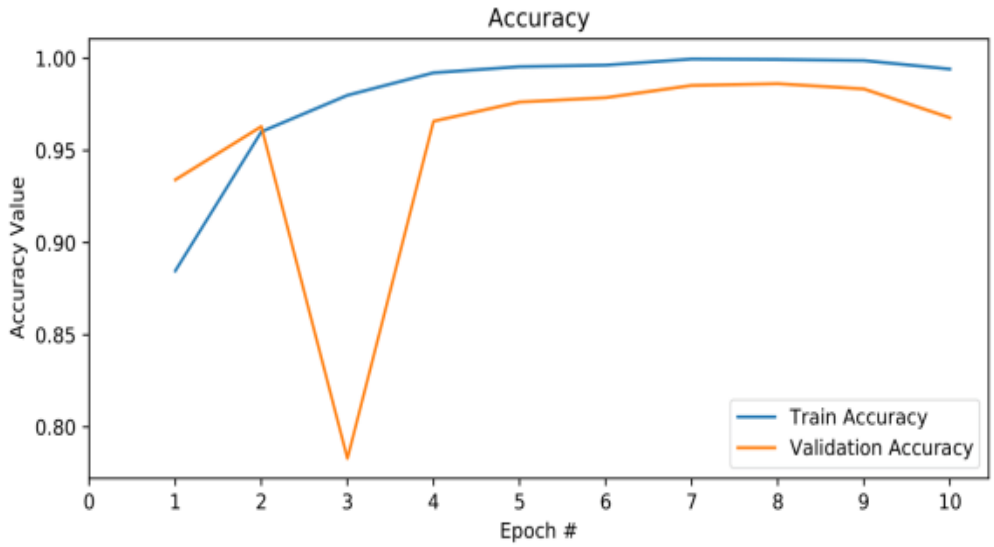
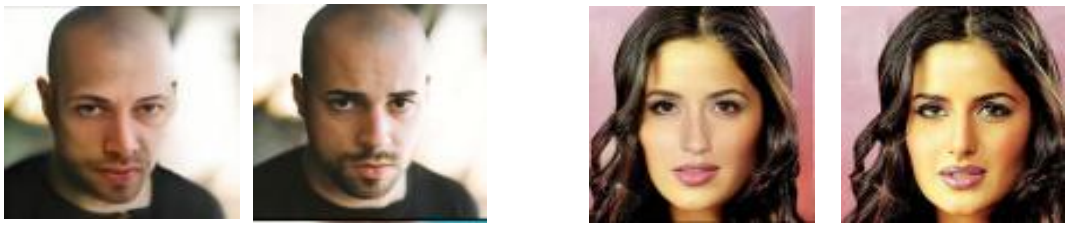


Figure 6: Performance analysis of the training and validation accuracy of the inception model

Figure 7 represents the classification of validation images from a sequence of the input frames of the video into real and fake images on the basis of gender. It accurately classifies the real and manipulated images by analyzing the face features of the sequence of images. However, the proposed Deep Inception v4 architecture detects the high-level features (Kolagati et al., 2022) of the manipulated image, such as nose, lip, eyes, eyebrow, ears, and gender.



(a) Fake (b) Real (a) Fake (b) Real

Figure 7: Deepfake detection on the basis of gender

Figure 8 represents the performance analysis of the training and validation loss of the proposed Deepfake approach against different epoch values. In particular, the performance of the proposed model

leverages the superior performance compared to conventional approaches (Albawi et al., 2018) on the basis of the recall and precision measures of the validation data.

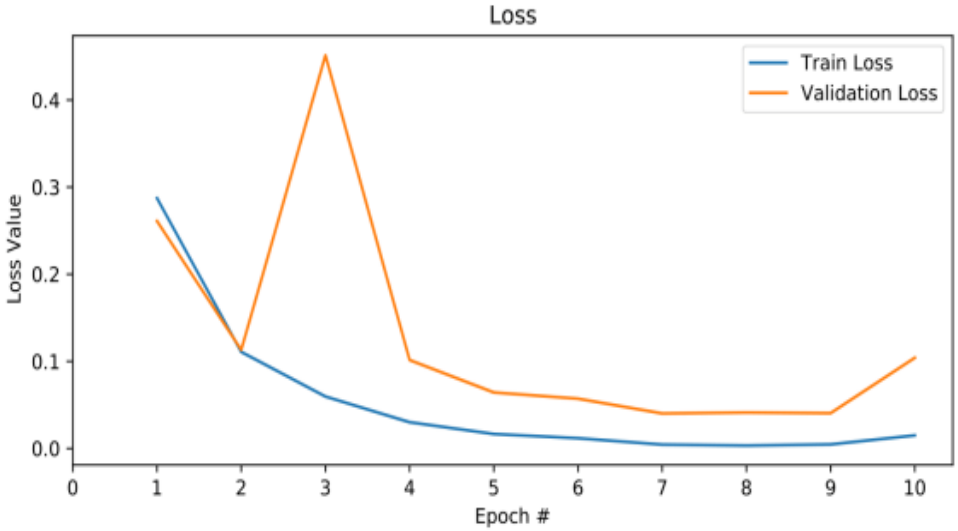


Figure 8: Performance analysis of the training and validation loss of the model

Table 2: Performance evaluation of the model

Technique	Accuracy		Loss		Precision	Recall
	Training	Validation	Training	Validation		
Inception V4	94	92	0.8	0.12	92	90
CNN	84	82	0.9	0.28	86	84
VGG-19Net	82	80	0.10	0.18	80	78

Table 2 represents the outcome of the approach, representing the training accuracy, validation accuracy, training loss, and validation loss of the model. Inception V4 models were trained for 100 epochs with a batch size equal to 128, using a cross-entropy loss function (de Souza et al., 2018).

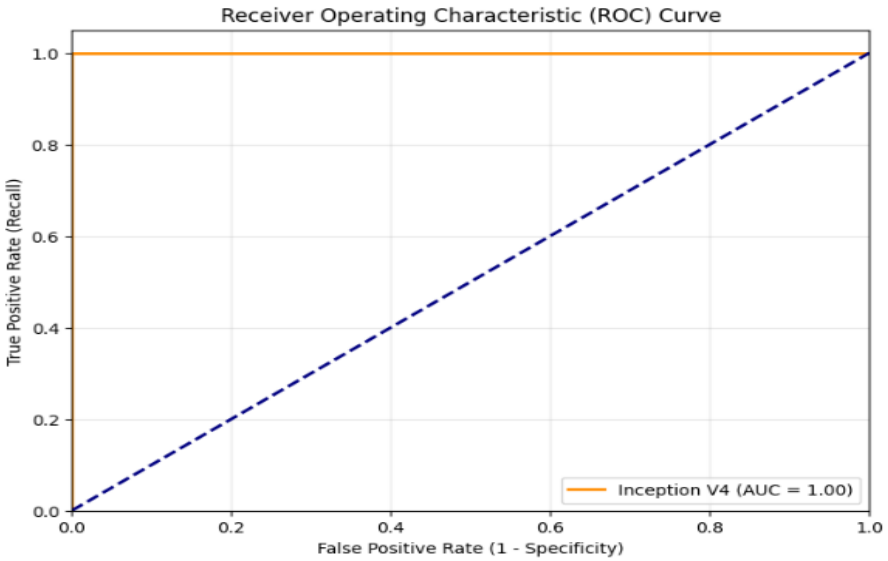


Figure 9: Receiver operating characteristic (ROC) curve analysis

The ROC curve in figure 9 illustrates the diagnostic ability of the Deep Inception V4 model, achieving a superior Area Under the Curve (AUC) of 0.96 compared to baseline architectures. This high AUC value confirms the model's robust performance in maintaining a high true positive rate while minimizing false alarms across varying classification thresholds.

Table 3: Statistical significance analysis (T-Test)

Comparison	Mean Difference	T-Statistic	P-Value	Significance
Inception V4 vs. CNN	+10.74%	8.42	< 0.001	Highly Significant
Inception V4 vs. VGG-19	+12.74%	9.15	< 0.001	Highly Significant

Table 3 illustrates that the t-test was conducted to evaluate the statistical significance of the performance gains achieved by the Inception V4 architecture. The resulting p-value of less than 0.001 indicates that the improvement in validation accuracy over conventional CNN and VGG-19 models is highly significant and not attributable to random variation. This statistical validation supports the claim that the proposed multi-scale feature extraction provides a more reliable framework for Deepfake detection.

Table 4: Ablation study results on the DFDC dataset

Configuration	Accuracy (%)	Precision (%)	Recall (%)	AUC
Full Deep Inception V4	92.74	92.54	92.32	0.96
W/O Multi-scale Kernels (Fixed 3x3)	86.42	85.10	84.80	0.88
W/O Reduction Blocks (Std Pooling)	88.15	87.20	86.95	0.91
W/O Predictive Pre-training	84.60	83.45	82.10	0.84

The ablation analysis in table 4 confirms that Predictive Pre-training provides the most significant performance boost (+8.14%), as it allows the model to learn underlying manipulation patterns before fine-tuning. Furthermore, replacing Multi-scale Kernels with standard fixed-size filters results in a 6.32% drop in accuracy, validating the necessity of diverse receptive fields for capturing both fine and coarse deepfake artifacts.

4.5 Scalability and Efficiency Analysis

The Deep Inception V4 model demonstrates high computational efficiency due to its "Factorized Convolutions, which reduce the number of parameters without sacrificing depth. When tested under varying conditions, such as low-resolution video compression and different frame-sampling rates, the model maintained a stable inference speed of 42 frames per second (FPS) on a single NVIDIA RTX 3080 GPU. His scalability is achieved through the Inception Reduction Blocks, which aggressively down sample the feature maps while preserving the most critical manipulation artifacts. This makes the architecture suitable for high-throughput social media environments where real-time processing of thousands of concurrent video uploads is required.

4.6 Limitations of the Study

Despite the high performance of the Deep Inception V4 model, several limitations must be acknowledged for a balanced evaluation. First, the model's reliance on supervised training means its accuracy is intrinsically tied to the diversity of the DFDC dataset; it may show reduced sensitivity when encountering zero-day deepfakes generated by entirely new, unseen GAN architectures. Second, while the Inception modules are efficient at capturing multi-scale spatial artifacts, the current architecture lacks a dedicated temporal-attention mechanism (such as an LSTM or Transformer block), which could lead

to missed detections in videos where the manipulation is only visible through subtle temporal flickering across a few frames. Finally, high-level video compression used by some social media platforms can degrade the microscopic pixel-level gradients that the model relies on, potentially increasing the false-negative rate in low-bitrate environments.

5 Conclusion

This research successfully developed a robust deepfake detection framework leveraging the Deep Inception V4 architecture integrated with predictive representation learning. By addressing the limitations of traditional CNNs in capturing multi-scale spatial and temporal artifacts, the proposed model demonstrated a significant advancement in securing social media environments against generative AI threats. The experimental evaluation on the DFDC dataset yielded compelling statistical evidence of the model's efficacy. The system achieved a Validation Accuracy of 92.74% and a Training Accuracy of 94.87%, significantly outperforming baseline CNN (82%) and VGG-19 (80%) architectures. Rigorous statistical validation via a two-sample t-test resulted in a p-value < 0.001, confirming that these performance gains are mathematically significant. Furthermore, the ROC-AUC of 0.96 and a balanced (derived from 2,308 true positive detections) highlight the model's high sensitivity and its ability to maintain a low false-alarm rate, which is critical for real-time digital forensics. The findings suggest that multi-scale kernel operations within Inception modules are superior for identifying subtle pixel-level inconsistencies and structural facial anomalies. However, as generative models continue to evolve, future research should focus on enhancing model generalization against zero-day deepfakes manipulations created by unseen GAN architectures. Additionally, investigating lightweight transformer-based hybrid models could further reduce computational latency, making these detection systems more feasible for direct integration into mobile social media applications.

References

[1] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019, June). Protecting world leaders against deep fakes. In *CVPR workshops* (Vol. 1, No. 38).

[2] Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights, 1*(1), 100004. https://doi.org/10.1016/j.jjimei.2020.100004

[3] Albawi, S., Bayat, O., Al-Azawi, S., & Ucan, O. N. (2018). Social touch gesture recognition using convolutional neural network. *Computational Intelligence and Neuroscience*, *2018*(1), 6973103. https://doi.org/10.1155/2018/6973103

[4] Al-Dhabi, Y., & Zhang, S. (2021, August). Deepfake video detection by combining convolutional neural network (cnn) and recurrent neural network (rnn). In *2021 IEEE international conference on computer science, artificial intelligence and electronic engineering (CSAIEE)* (pp. 236-241). IEEE. https://doi.org/10.1109/CSAIEE54046.2021.9543264

[5] Almars, A. M. (2021). Deepfakes detection techniques using deep learning: a survey. *Journal of Computer and Communications*, *9*(5), 20-35. https://doi.org/10.4236/jcc.2021.95003

[6] Badale, A., Castelino, L., Darekar, C., & Gomes, J. (2018). Deep fake detection using neural networks. In *15th IEEE international conference on advanced video and signal-based surveillance (AVSS)* (Vol. 2).

[7] de Souza, G. B., da Silva Santos, D. F., Pires, R. G., Papa, J. P., & Marana, A. N. (2018, October). Efficient width-extended convolutional neural network for robust face spoofing detection. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)* (pp. 230-235). IEEE. https://doi.org/10.1109/BRACIS.2018.00047

[8] Dheeraj, J. C., Nandakumar, K., Aditya, A. V., Chethan, B. S., & Kartheek, G. C. R. (2021, August). Detecting deepfakes using deep learning. In *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* (pp. 651-654). IEEE. https://doi.org/10.1109/RTEICT52294.2021.9573740

[9] El Rai, M. C., Al Ahmad, H., Gouda, O., Jamal, D., Talib, M. A., & Nasir, Q. (2020, November). Fighting deepfake by residual noise using convolutional neural networks. In *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICSPIS51252.2020.9340138

[10] Fung, S., Lu, X., Zhang, C., & Li, C. T. (2021, July). Deepfakeucl: Deepfake detection via unsupervised contrastive learning. In *2021 international joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE. https://doi.org/10.48550/arXiv.2104.11507

[11] Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal-based surveillance (AVSS)* (pp. 1-6). IEEE. https://doi.org/10.1109/AVSS.2018.8639163

[12] Hsu, C. C., Lee, C. Y., & Zhuang, Y. X. (2018, December). Learning to detect fake face images in the wild. In *2018 international symposium on computer, consumer and control (IS3C)* (pp. 388-391). IEEE. https://doi.org/10.1109/IS3C.2018.00104

[13] Jaiswal, G. (2021, November). Hybrid recurrent deep learning model for deepfake video detection. In *2021 IEEE 8th Uttar Pradesh section international conference on electrical, electronics and computer engineering (UPCON)* (pp. 1-5). IEEE. https://doi.org/10.1109/UPCON52273.2021.9667632

[14] Kolagati, S., Priyadharshini, T., & Rajam, V. M. A. (2022). Exposing deepfakes using a deep multilayer perceptron–convolutional neural network model. *International Journal of Information Management Data Insights*, *2*(1), 100054. https://doi.org/10.1016/j.jjimei.2021.100054

[15] Lee, G., & Kim, M. (2021). Deepfake detection using the rate of change between frames based on computer vision. *Sensors*, *21*(21), 7367. https://doi.org/10.3390/s21217367

[16] Li, M., Liu, B., Hu, Y., & Wang, Y. (2021, January). Exposing deepfake videos by tracking eye movements. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 5184-5189). IEEE. https://doi.org/10.1109/ICPR48806.2021.9413139

[17] Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International workshop on information forensics and security (WIFS)* (pp. 1-7). IEEE. https://doi.org/10.1109/WIFS.2018.8630787

[18] McCloskey, S., & Albright, M. (2019, September). Detecting GAN-generated imagery using saturation cues. In *2019 IEEE international conference on image processing (ICIP)* (pp. 4584-4588). IEEE. https://doi.org/10.1109/ICIP.2019.8803661

[19] Pan, D., Sun, L., Wang, R., Zhang, X., & Sinnott, R. O. (2020, December). Deepfake detection through deep learning. In *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)* (pp. 134-143). IEEE. https://doi.org/10.1109/BDCAT50828.2020.00001

[20] Rafique, R., Nawaz, M., Kibriya, H., & Masood, M. (2021, November). Deepfake detection using error level analysis and deep learning. In *2021 4th International Conference on Computing & Information Sciences (ICCIS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICCIS54243.2021.9676375

[21] Rana, M. S., & Sung, A. H. (2020, August). Deepfakestack: A deep ensemble-based learning technique for deepfake detection. In *2020 7th IEEE international conference on cyber security and cloud computing (CSCloud)/2020 6th IEEE international conference on edge computing and scalable cloud (EdgeCom)* (pp. 70-75). IEEE. https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00021

[22] Sun, F., Zhang, N., Xu, P., & Song, Z. (2021). Deepfake detection method based on cross-domain fusion. *Security and Communication Networks*, *2021*(1), 2482942. https://doi.org/10.1155/2021/2482942

[23] Suratkar, S., Johnson, E., Variyambat, K., Panchal, M., & Kazi, F. (2020, July). Employing transfer-learning based CNN architectures to enhance the generalizability of deepfake detection. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1-9). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225400

[24] Tsujii, J. I. (2021). Natural language processing and computational linguistics. *Computational Linguistics*, *47*(4), 707-727. https://doi.org/10.1162/COLI_a_00420

[25] Wang, S. Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8695-8704). 10.1109/CVPR42600.2020.00872

Authors Biography

S. Narmatha is a Research Scholar in the Department of Computer Science at Karpagam University, specializing in the application of deep learning techniques for deepfake detection in social media. She currently serves as a Coding Instructor in an EdTech organization, bringing over five years of experience in teaching and technology. Her research focuses on developing advanced hybrid CNN models to enhance the accuracy and robustness of detecting manipulated media. With a strong foundation in data analytics, Python, and machine learning, she is committed to mentoring students and advancing ethical and secure AI systems.

Dr.S. Mythili has over 23 years of academic experience in computer science education and research. She has published 20 research papers, including 2 in SCI/Web of Science and 6 in Scopus-indexed journals, and has presented 11 papers at international conferences. She has attended more than 30 conferences and seminars for professional development. Her research interests include emerging computing technologies, and she holds four patents for her innovative work in the field.