Journal of Information Systems Engineering and Management
2025, 10(18s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Multi Cascaded Face Artefact Detection with Xception Convoluted LSTM Network for Deep Fake Detection

Aishwarya Rajeev[1]*, Raviraj P[2]
[1]*Research Scholar, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, INDIA.*
[2]*Professor and Head, Department of Computer Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, INDIA.*
**Corresponding Email: aishwaryarajeev@gmail.com*
Co-author Email: raviraj@gsss.edu.in

ARTICLE INFO

Received: 21 Dec 2024

Revised: 31 Jan 2025

Accepted: 10 Feb 2025

ABSTRACT

Introduction: Facial deepfakes are becoming increasingly realistic, creating it difficult for humans to distinguish between fake and real videos. This technology poses significant risks across various sectors, including politics, entertainment, and cybersecurity.

Objectives: To address these challenges, deepfake detection systems must enhance their detection capabilities, ensure temporal consistency, and improve face detection techniques. Existing systems often struggle with subtle manipulations, necessitating a combination of spatial and temporal information.

Methods: This paper introduces a novel methodology employing a Multi-cascaded Face Artefact Detection approach combined with an Xception Convoluted Long Short-Term Memory (LSTM) Network to overcome existing limitations.

Results: The method begins with pre-processing the input video by converting it into frames at a consistent rate. Face detection is conducted using Multi-Task Cascaded Convolutional Networks (MTCNN), which identifies as well as resizes faces in each frame. Key facial landmarks are then extracted using Dlib to capture intricate manipulations.

Conclusions: The Xception Convoluted LSTM Network detects spatial features and temporal dependencies to identify inconsistencies in manipulated videos. The system was evaluated using the FaceForensics++ dataset, achieving impressive performance metrics: 94.72% accuracy, 92.09% precision, 95.06% recall, 93.55% F1-score, 94.50% specificity, and 94.78% AUC, underscoring the effectiveness of the proposed approach compared to state-of-the-art models.

Keywords: Deep fake detection, Multi-cascaded Face Artefact Detection, Xception Convoluted Long Short-Term Memory, Multi-Task Cascaded Convolutional Networks, FaceForensics++ dataset.

INTRODUCTION

Nowadays, images and videos play an important part in digital communication, and regardless of whether they are of private (social network), juridical (trial), or security (surveillance, police investigation) origin, they can be used as evidence [1]. As a result, confirming their origin and legitimacy is critical to preventing harmful use. However, because modifying software is easily accessible and used, falsified information is becoming more widespread and increasingly difficult for people to recognise [2]. Although the capacity to produce or alter facial clues using artificial intelligence has potential benefits in fields such as art, video games, face anonymization, and cinematography, there are also various uses that might be destructive to individuals, groups, and society as a whole [3].

Initially, the name "DeepFake" suggested a DL-based method for altering media by swapping the faces of two people. It first debuted in 2017, when famous faces were turned into pornographic flicks using a machine learning system [4]. Aside from pornography, some of the most destructive applications of this technology include online deception and financial fraud [5]. However, the term "DeepFakes" has increasingly become associated with most forms of face

and/or audio manipulation. Face swapping, face creation from scratch, facial attribute modification, and facial expression manipulation/re-enactment are common examples of such operations [6]. Because Deepfake takes just a small number of face images to allow video face-swapping, some malevolent individuals have used the internet's data to create a large number of fake videos. The pornography business was the first to use this technique, with several face-swapping porn movies depicting female celebrities circulating on the internet [7]. Replacing pornographic movie heroines with female stars, as well as forging video footage for politicians, corporate leaders, and other powerful figures, in order to mislead public opinion, gain selection, and manipulate stock prices. Many false face-swapping movies represent a significant risk to national security, societal stability, and personal privacy. Verifying the authenticity of internet movies is quickly becoming one of the most popular concerns in digital culture [8].

The deepfake creation process produces artefacts in both the spatial [9] and frequency domains [10], particularly in certain areas of the face [11]. As a result, numerous research focuses their study on facial image components. Targeting motions of certain face features enables us to detect artefacts caused by deep learning algorithms [12]. These "temporal artefacts" alter the dynamic fluctuation of the pixel in specific locations, affecting the view's quality. As a result, deep learning-based detection techniques have become widely used due to their exceptional performance on cutting-edge face deepfake datasets [13]. CNN architectures are particularly popular, generally employing a clipped picture of a subject's face from a video frame as input to determine if it is genuine or false [14]. Transformer-type architectures have recently been repurposed for deepfake detection. Some deep architectures leverage the temporal dimension by classifying images since existing facial deepfakes are frequently made frame-by-frame and can therefore exhibit temporal consistency in a deepfake movie [15]. Deepfake detection involves combining traditional methods with advanced deep learning techniques. Traditional methods involve checking facial expressions, eye movements, and lip sync for anomalies. Visual face artefacts, like unrealistic reflections or white blobs, are used for detection. However, downsampling images for computational constraints makes it difficult to catch these artefacts. This study aims to overcome this issue by using pre-processing approaches and collecting various visual artifacts to improve model generalization and overcome hurdles in deepfake detection. The primary contribution of the proposed methodology as follows:

1. The proposed methodology ensures temporal consistency by converting input videos into frames at a consistent frame rate. This step is crucial for accurate temporal analysis, which is often lacking in current deepfake detection systems.

2. The method employs MTCNN for precise face detection and resizing, followed by Dlib for extracting key facial landmarks. This approach enhances the model's ability to capture intricate manipulations commonly found in deepfakes, addressing the sensitivity issue present in existing systems.

3. The core of the methodology involves the Xception Convoluted LSTM Network, which combines spatial feature extraction using Xception convolution layers with temporal dependency capture through LSTM networks. The model can identify temporal irregularities and fine-grained features in edited videos because to its integration.

4. The inclusion of a temporal attention mechanism allows the network to focus on the most informative frames, enhancing detection accuracy. This feature ensures that the model can efficiently aggregate features across frames, improving its robustness against sophisticated manipulations.

5. The approach employs a dense layer with a single neuron and a sigmoid activation function for robustness, dense layers with ReLU activation and dropout to minimize overfitting, and a max-pooling layer for dimensionality reduction, making it ideal for binary classification tasks, indicating video authenticity.

The manuscript's remaining section is arranged as follows: The second section looks at the body of existing literature. In the third section, the research technique was covered in detail. The fourth section discusses the outcomes and implementation of the suggested strategy. A synopsis of the key findings is provided in the conclusion.

LITERATURE REVIEW

This section summarizes various categorization techniques developed by researchers for deep fake detection.

This paper proposed by Ciamarra et al [16] how deepfake creation affected scene characteristics, suggesting that the overall geometry of the scene could be altered by the deepfake generation process. A descriptive method called SurFake was utilized to train a CNN for deepfake detection by analyzing surface characteristics. Experimental results

on that SurFake could discriminate between pristine and altered images and improve detection accuracy. However, it had limitations, lacked data augmentation techniques, and missed local surface geometry. Further experimentation on deepfake datasets was needed to improve the model's performance and leverage other geometric information.

Saikia et al. [17] examined deep fakes, which are digitally produced videos that are extremely lifelike and hard to identify with conventional detection techniques. In order to recognise such data, discriminators based on Convolutional Neural Networks (CNNs) were frequently employed; nonetheless, their primary focus was on the spatial characteristics of individual video frames. In this work, temporal characteristics were extracted using an optical flow-based feature extraction technique and then input into a hybrid model for classification. However, the proposed method still struggled with variations in deepfake techniques and unseen data, potentially affecting its overall reliability.

Khan et al. [18] explored the generalization challenge of DL-based detection systems for deepfake detection. They assessed several datasets, pre-training techniques, and deep learning model architectures. Using four distinct deepfake detection benchmarks, the study evaluated two transformer-based models and eight supervised deep learning. The goal of the investigation was to determine which datasets had the best generalisation capabilities, which models performed the best, and how picture augmentations affected model performance. The study also looked into the trade-off between performance, efficiency, and model size. The findings demonstrated that in deepfake identification, Transformer models performed better than CNN models. Additionally, the study demonstrated that image augmentations could improve performance, particularly for Transformer models. However, it was limited due to its reliance on specific deepfake detection benchmarks, which could affect the effectiveness of the models.

Liu et al. [19] Detecting artefacts was the mainstay of previous techniques, but as deep forgeries technology advanced, high-quality synthetic pictures and reconstruction techniques advanced as well. They addressed this by introducing a deep forgery detection technique that combined fine-grained artefact characteristics with deep neural networks. The technique used face mask deformation and blurring, facial colour conversion, and facial frequency domain conversion to replicate a variety of facial synthesis data. Multiple perturbations of real photos were used to train the classifier model, and stability was ensured via fine-grained artefact characteristics. However, it was limited by specific perturbations and fine-grained artifact features, and as deepfake technologies evolved, new synthesis methods could emerge that the model hadn't been trained to detect effectively.

Hasanaath et al [20] Deepfakes are artificially produced films or images that are produced by deep neural networks, posing threats like social media disinformation and fraud. Existing detection algorithms have trouble generalizing across different deepfakes' generating methods and across different corpora. The efficient-capsule network (E-Cap Net), a unique deep learning model, is suggested for the classification of face photos produced by various deepfake methods. The E-Cap Net is strong and lightweight since it employs a cheap max-feature-map activation algorithm. However, it may struggle to adapt to various types of deep fakes techniques, it may require frequent updates and retraining for optimal performance.

Ilyas et al. [21] Existing detection models, such as convolutional neural networks, struggled to generalize across multiple deepfake generation techniques and cross-corpora settings. To address this, suggested the efficient-capsule network (E-Cap Net), to categorise facial photos produced by various deepfake generation techniques. Lightweight and durable, the E-Cap Net employed a low-cost max-feature-map (MFM) activation mechanism in each principal capsule. However, overfitting reduced the model's ability to accurately detect deepfakes in diverse and unseen data.

Al Dulaimi et al. [22] demonstrated a hybrid feature extraction strategy for identifying deepfakes. The model made use of strong 10-PCA features from clipped faces of 10 frames each video, as well as 128-identity features from FaceNet CNN. The study demonstrated that merging these two methods for feature extraction yielded better results for detecting fake videos than using each method alone. The proposed method outperformed traditional CNN models in terms of feature extraction and dimensionality reduction. However, processing features from multiple methods required more computational resources, potentially making the model less efficient.

As a result, the existing method for detecting deepfakes required further experimentation on deepfake datasets to improve performance and leverage geometric information. However, it struggled with variations in deepfake techniques and unseen data, potentially affecting its reliability. The model's reliance on specific detection benchmarks limited its effectiveness. Additionally, it was constrained by specific perturbations and fine-grained artifact features. The model introduced significant computational overhead, potentially affecting its efficiency in

deepfake detection applications. It also struggled to adapt to different deepfake techniques and required frequent updates and retraining. Overfitting reduced the model's ability to accurately detect deepfakes in diverse and unseen data.

PROPOSED METHODOLOGY:

Deep fake technology poses significant risks to sectors like politics, entertainment, and cybersecurity. It can deceive viewers into believing false information, leading to social, economic, and political consequences. To mitigate these negative impacts, deep fake detection systems need to improve detection capabilities, address temporal consistency in video analysis, and improve pre-processing techniques for accurate face detection and facial feature extraction. Current systems lack sensitivity to intricate facial changes, resulting in inability to capture perplexing manipulations. Incorporating both spatial and temporal information with collaborative attention from facial features is essential for improving the robustness, accuracy, and adaptability of deep fake detection systems.

The proposed methodology enhances deep fake detection by implementing a Multi-Cascaded Face Artefact Detection approach with an Xception Convoluted LSTM Network, effectively addressing gaps in robustness and adaptability to new deep fake techniques. The process begins with preprocessing, where the input video is converted to frames at a consistent frame rate, establishing temporal consistency essential for accurate analysis. Face detection is initiated using the Viola-Jones Algorithm, which rapidly identifies face regions in each frame using Haar-like features, integral images, and cascade classifiers. This efficient face detection significantly reduces computational overhead by ensuring only frames with detected faces proceed for further processing. To refine the accuracy, especially in frames with challenging poses or partial occlusions, Multi-Task Cascaded Convolutional Networks (MTCNN) further localize and align faces, resizing them to a uniform size for reliable feature extraction. After precise face detection, key facial landmarks including the mouth, nose, and eyes are extracted by Dlib. These landmarks are critical for capturing the subtle manipulations often present in deep fake videos, ensuring comprehensive facial region representation. The core of the methodology, the Xception Convoluted LSTM Network, then takes over. Xception's convolution layers leverage depthwise separable convolutions to detect fine-grained spatial artefacts essential for identifying forged content. Simultaneously, LSTM layers detect irregularities characteristic of modified sequences by capturing temporal relationships across frames. A temporal attention mechanism focuses dynamically on the most informative frames, aggregating crucial features for enhanced detection accuracy. The aggregated features are then dimensionally reduced through max pooling, followed by a series of dense layers with ReLU activations to capture complex patterns, with dropout layers preventing overfitting.

Binary classification is made possible by the final dense layer, which utilize a sigmoid activation function to identify if the video is real or fake. The model's training process leverages backpropagation and the Adam optimizer for efficient parameter updates. Overall, the integration of Viola-Jones for efficient preprocessing alongside Xception Convoluted LSTM Network for spatial and temporal analysis significantly enhances the detection pipeline. This combination ensures the methodology is not only efficient but also robust and adaptable, capable of maintaining its effectiveness against various sophisticated and emerging deep fake techniques. Figure 1 below displays the architectural diagram for the proposed method.

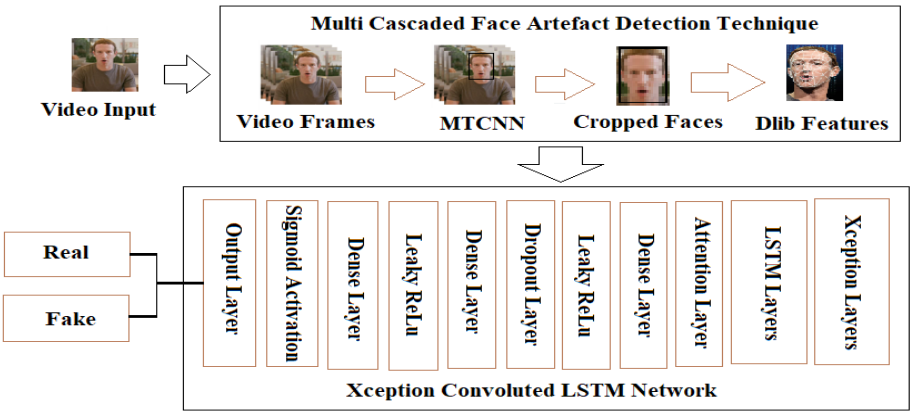

Figure 1: Architecture of the proposed methodology

Multi cascaded Face Artefact Detection

The process starts with pre-processing, converting video into frames, detecting faces using Multi-Task Cascaded Convolutional Networks, and extracting key facial landmarks using Dlib, capturing intricate manipulations in deep fakes.

Video Frames:

Pre-processing is crucial for model training and inference, ensuring data is in a suitable format. Input video is converted to frames at a consistent rate, ensuring temporal consistency for accurate analysis. Frames are then standardized in size, color format, and attributes to simplify the processing pipeline. Normalizing pixel values can speed up convergence during training and improve overall performance. Random cropping, rotation, flipping, and colour modifications are examples of data augmentation approaches that can increase the resilience of the detection system. In some cases, not all frames from the video are necessary for analysis. Frame sampling or key frames selection can reduce data while retaining essential information, making the pre-processing pipeline more efficient and reducing computational load. These steps transform data into a consistent and standardized format, ensuring high performance, accuracy, and reliability in the detection system.

Multi-Task Cascaded Convolutional Networks (MTCNN):

MTCNN was utilized for face detection, identifying faces in each frame and resizing them uniformly for reliable feature extraction. Face detection using Viola-Jones Algorithm is utilized as an initial face detection mechanism. This algorithm rapidly detects candidate face regions in each frame using Haar-like features, an integral image, and cascade classifiers. Viola-Jones efficiently identifies potential face regions, allowing MTCNN to focus on refining these regions for high precision. Viola-Jones serves as an initial filter, reducing computational load by selecting only regions with potential faces, which then proceed to MTCNN for detailed detection and alignment. MTCNN comprises a three-stage cascaded architecture that leverages three neural networks, each performing specific tasks for accurate face detection. Figure 2 illustrates the MTCNN structure. Fully convolutional networks make up the first network, P-Net; standard CNNs make up the other two networks, R-Net and O-Net. MTCNN is made up of three networks. Any size picture may be used as the MTCNN's input. The following three-stage cascaded architecture uses an image pyramid that is created by regularly resizing photos to different sizes.

Three tasks must be completed in order to train the networks: facial landmark localization, bounding box regression, and face categorization. Equation (1) illustrates that the loss for face categorization is cross-entropy loss. where $y_i^{det} \epsilon \{0,1\}$ is the ground truth label and P_i is the probability of the face produced by the network. A human face could be present in the image's bounding box; thus, during training, the offset between it and the closest ground truth must be kept to a minimum. Equation (2) illustrates the Euclidean loss for the bounding box, where y_i^{box} is the closest ground truth and y_i^{box} is the bounding box result derived from the network. The enclosing box's dimensions are four and include the height, width, and left top coordinates. Euclidean loss is also used in facial landmark regression, as seen in Equation (3), where $y_i^{landmark}$ is the ground truth coordinate and $y_i^{landmark}$ is the coordinate of the face landmark retrieved from the network. To prevent overfitting, the RMSProp optimizer is utilized during training, ReLU is used as the activation function, Xavier is used as the weight initializer, and L2 weight regularizes are set for each convolutional filter.

$$L_i^{del} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i) \quad (1)$$

$$L_i^{box} = ||y_i^{box} - y_i^{box}||_2^2 \quad (2)$$

$$L_i^{landmark} = ||y_i^{landmark} - y_i^{landmark}||_2^2 \quad (3)$$

The P-Net is a fully convolutional network that can classify each 12x12 area in the input image as having a chance of having a human face. By setting a threshold t_1, regions with probabilities exceeding t_1 are selected for non-maximum suppression (NMS). Following NMS, the remaining boxes are sent as input to the R-Net after being resized to 24 x 24 pixels. The R-Net then evaluates these boxes, assigning probabilities for the presence of a human face. A second threshold t_2 is applied, and boxes with probabilities greater than t_2 undergo another round of NMS. The resulting boxes are resized to 48x48 pixels and fed into the O-Net. Similar to the R-Net, the O-Net assesses these boxes, assigning face probabilities. A final threshold t_3 is applied, and boxes with probabilities exceeding t_3 go through NMS. The MTCNN's final outputs are the boxes that are left behind after this operation. These three networks are

trained sequentially. First, the P-Net was trained. The training set is then used as input to the trained P-Net, generating regions with potential faces. The R-Net's training set consists of these areas. Likewise, the outputs of the trained P-Net and R-Net are sent into the O-Net's training set. P-Net, R-Net, and O-Net training is hence sequential and interconnected rather than independent.

It is a common strategy used in object detection. The MTCNN's NMS merges boxes with large regions of overlap. The Greedy-NMS algorithm's stages are as follows:

Step 0: Set the empty set S_e, the input box set S_0, and the IOU threshold δ.

Step 1: Select the box with the highest probability value, B_1. Remove B_1 from S_0 and add $B_1$to S_e.

Step 2: Using B_1, compute the IOU of each remaining box as you go through the remaining boxes in S_0. Eliminate this last box from S_0 if the IOU is bigger than δ.

Step 3: NMS is complete if there are still boxes in S_0; if not, go to step 1. These are the output boxes in S_e. Eq. (4) shows the IOU computation procedure.

$$IOU = \frac{Region\ 1 \cap Region\ 2}{Region\ 1 \cup Region\ 2} \tag{4}$$

After the MTCNN process, Dlib is utilized to extract facial features, identifying key landmarks in the cropped and resized face images. This step is essential for detecting subtle manipulations present in deep fake videos.

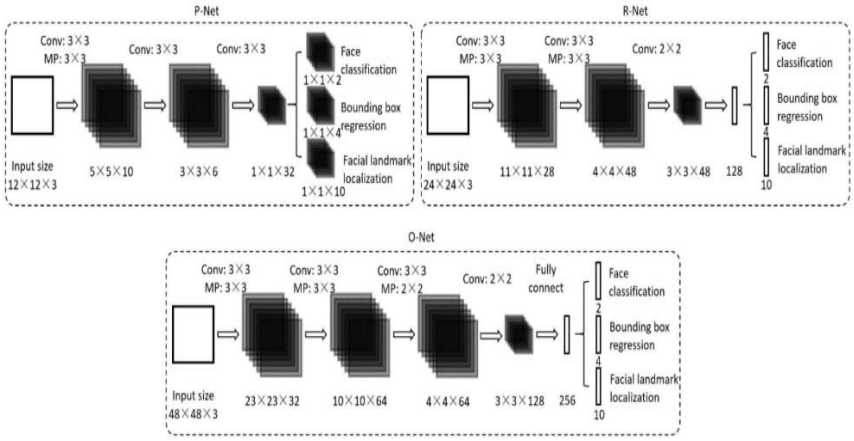

Figure 2: Structure of MTCNN [23]

Cropped Faces

Let X be a video series of T frames, represented as $X^{(t)}, t = 1,2, \dots, T$. Every $X^{(t)} \in \mathbb{R}^{H \times W \times C}$ is a 2D picture, where C is the number of colour channels and $(H \times W)$is the image resolution. Every frame $X^{(t)}$ is preprocessed before being fed into the CNN. Face identification and landmark localization are done using the well-known Dlib program. After that, the face is trimmed to 224 x 224 resolution and facial alignment is carried out. The total accuracy of deepfake detection is strongly influenced by the cropped face's quality. By concentrating just on the regions that could be altered rather than utilizing the entire frame, the cropping stage helps reduce background noise. Lastly, the CNN model receives the T clipped face areas, represented as $X'^{(t)}, t = 1,2, \dots, T$, in order to extract features.

The extracted facial features are then processed through a novel Xception Convoluted LSTM Network to extract both spatial and temporal information.

Xception Convoluted LSTM Network

The Xception network design is used as the feature extraction backbone network once cropped face images have been obtained. This is made possible by removing the fully-connected (FC) layer from the top of the Xception network, which enables the network to produce feature maps—a 2D deep representation of each cropped face image directly. The output feature maps have dimensions of 2048 × 7 × 7, which are then flattened for further processing. The Xception architecture is a neural network that relies solely on deep, separable convolution layers. The decoupling of cross-chain correlations and spatial correlations inside convolutional neural network feature maps is the

fundamental premise of this design. The Xception model comprises 36 convolutional layers organized into 14 modules, each connected by linear residual links, except for the initial and final modules. These depth wise separable convolution layers enhance the network's efficiency and effectiveness in feature extraction.

The Xception model builds upon and refines the principles of the Inception architecture, leading to its designation as "Extreme Inception" or Xception. This approach improves the network's ability to map and separate spatial and cross-channel correlations, making it highly suitable for tasks includes image classification and face recognition.

Then it captures the temporal dependencies across frames, employing LSTM models, as they remember long-term dependencies and patterns in sequences, making them ideal for detecting temporal inconsistencies that are common in deep fake videos. A gated recursive neural network, the LSTM network is perfect for analyzing and forecasting important events with lengthy time series data intervals. The issue of temporal interdependence across frames is resolved by incorporating a gating mechanism to regulate information transmission. This is achieved by enhancing linear dependencies through three control units: the input gate, output gate, and forget gate. The input gate establishes the amount of network state data that must be stored to the internal state.

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i) \tag{5}$$

where the input gate's weight matrices W_i and U_i its bias term is b_i, its logistic function is σ, and the output of the memory block at time t-1 and the input vector at time t are indicated, respectively, by h_{t-1} and x_t. The threshold for deleting prior data is set by the forget gate.

$$f_t = \sigma(U_f h_{t-1} + W_f x_t + b_f) \tag{6}$$

where b_f is the forget gate's bias term and W_f and U_f are its weight matrix. The output gate controls the amount of data that the internal state must now output to the external state.

$$o_t = \sigma(U_o h_{t-1} + W_o x_1 + b_0) \tag{7}$$

where b_0 is the output gate's bias term and W_o and U_o are its weight matrix.

Temporal relationships between frames may be efficiently resolved using LSTM. When the input parameter dimension is quite big, an attention method is used to enhance the performance of the LSTM by concentrating on influential factors. The aggregated features from the attention layer are then processed through a max pooling layer to reduce dimensionality while retaining the most critical information. In the attention mechanism, the input sequence was signified by the intermediate output of the LSTM encoder, a model is selectively trained to learn these inputs, and the output sequence was associated with these inputs. The output $[h_1, h_2, h_3, \dots, h_n]$ I the LSTM was transformed nonlinearity to obtain $[u_1, u_2, u_3, \dots, u_n]$. The attitude and position of shield tunnelling are significantly influenced by certain operating parameters, hence they should be given more importance. This is followed by a series of dense layers that further process the features.

Classification

Each neurone in a neural network was connected to every other neuron in the network's first dense layer, which is a completely connected layer. This layer's output is subjected to the ReLU activation function, which introduces non-linearity and is essential for learning intricate patterns and representations. A portion of neurons are randomly dropped out throughout each training cycle by the Dropout layer, which serves as regularisation to avoid overfitting. The model becomes more resilient to unseen input as a result of the network being forced to learn redundant representations and enhanced generalisation. The second dense layer with ReLU activation and dropout mirrors the first layer and adds another dense layer with ReLU activation to capture even more complex features and interactions within the data. The subsequent dropout layer ensures that learning is distributed across the network. For binary classification tasks, such as identifying if a video is real or false, the final output layer is a dense layer with a single neurone and sigmoid activation. By mapping the input to a value between 0 and 1, the sigmoid activation function generates a probability score that shows how likely it is that the input belongs to the positive class.

RESULT AND DISCUSSION

This section provides thorough analysis of the results and performance indicators produced by the proposed method. It also has a comprehensive evaluation that shows how well the model works in comparison to other approaches or accepted standards.

System and Tool Configuration:

Tool	:	PYTHON 3.10
OS	:	Windows 10 (64-bit)
Processor	:	Intel R core ™ i-5
RAM	:	16 GB RAM

Dataset Description

The FaceForensics++ dataset is a comprehensive and invaluable resource designed for the study and detection of facial manipulation techniques. The FaceForensics++ dataset, which includes 1000 original movies, is one of the biggest deepfakes datasets. It builds upon the original FaceForensics dataset by incorporating a broader range of manipulations and higher-quality video data. This dataset features video sequences that have been manipulated using advanced techniques such as DeepFakes, which involve deep learning models for face swapping; Face2Face, a real-time facial reenactment method; FaceSwap, a traditional computer graphics-based face-swapping technique; and NeuralTextures, which utilize neural networks to generate realistic textures and facial expressions. The FaceForensics++ dataset offers videos in three quality levels to facilitate robust evaluation under a range of conditions: RAW, which are high-quality videos that are uncompressed; HQ, which are compressed with little loss of quality; and LQ, which are heavily compressed to mimic low-bandwidth situations. Researchers can evaluate the robustness and precision of their detection algorithms across varying video quality and compression settings because to this variation. The dataset is widely employed in the research community for multiple purposes, including the improvement and benchmarking of deepfake detection algorithms, robustness testing of these algorithms under different compression and noise conditions, and forensic analysis to understand the artifacts and patterns introduced by various manipulation techniques. For practical use, the FaceForensics++ dataset is meticulously organized into an 80% training set and a 20% testing set, facilitating the development and validation of machine learning models. While maintaining a distinct set for objective assessment, this separation guarantees that researchers may train their models on a significant amount of the data. The FF++ dataset model is shown in figure 3 below.

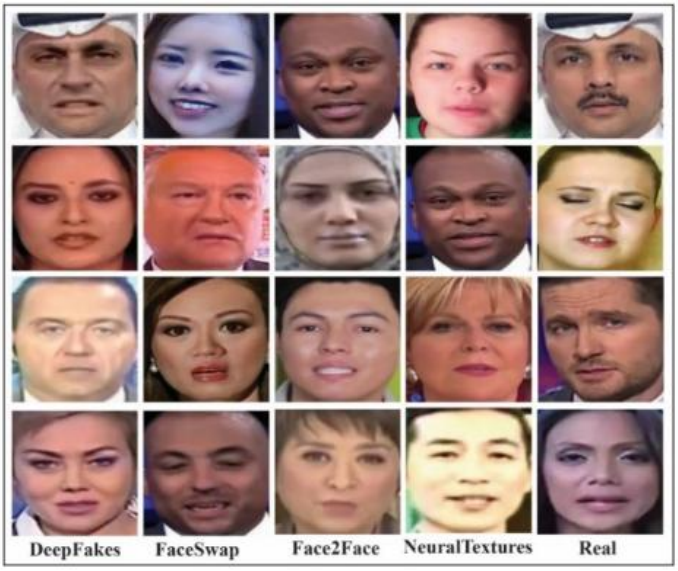

Figure 3: FaceForensics++ dataset

Experimental result:

The input video is first converted into frames at a consistent frame rate to ensure temporal consistency. Subsequently, MTCNN are employed to detect faces within each frame, providing precise bounding boxes for accurate feature extraction. The workflow demonstrates how these steps collectively enhance the accuracy and reliability of the deepfake detection system. The figure 4 and 5 illustrates the pre-processing steps and the Multi-Cascaded Face Artefact Detection technique used for deepfake detection.

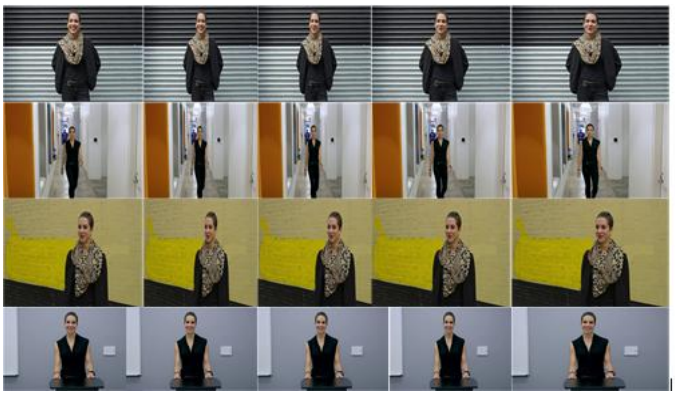

Figure 4: Preprocessed sample frames

Figure 5: Multi-Cascaded Face Artefact Detection technique used for deepfake detection

Subsequently, Dlib facial features are extracted to identify key facial landmarks from the cropped and resized face images. These landmarks are crucial for capturing the subtle and perplexing manipulations often present in deepfake videos, increasing the model's ability to detect such forgeries accurately. The following figure 6 shows the Facial Landmark Extraction using Dlib.

Figure 6: Facial Landmark Extraction using Dlib

The model outputs a probability score that reflects the likelihood of the video being real. This score is derived from the features extracted during the pre-processing and facial landmark identification stages, allowing the model to make informed predictions. A video is more likely to be authentic if its score is closer to 1, and it is more likely to be fraudulent if its score is closer to 0. By providing these probability scores, the model enables a nuanced assessment of each video, facilitating better decision-making in applications such as deepfake detection and content verification. The following figure 7 represents the classification process of the proposed model.

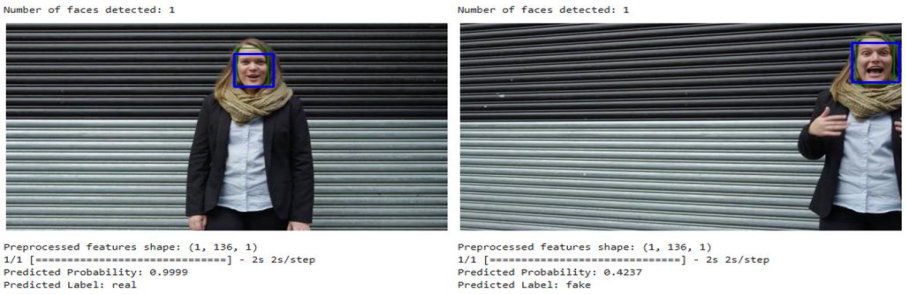

Figure 7: Classification process of the proposed model

Performance Evaluation:

This section evaluates the performance of the proposed deep fake detection model, focusing on its ability to accurately differentiate between real and manipulated videos. The model's accuracy and loss over 500 training epochs are monitored to understand its convergence speed. Additionally, potential trade-offs between generalizability, computational efficiency, and model accuracy are discussed, noting that improvements in accuracy may come at the expense of other factors. Overall, this section highlights the model's effectiveness in identifying real versus fake videos. The accuracy and validation results of the proposed approach are illustrated in Figure 8, demonstrating its performance during the training and validation phases.

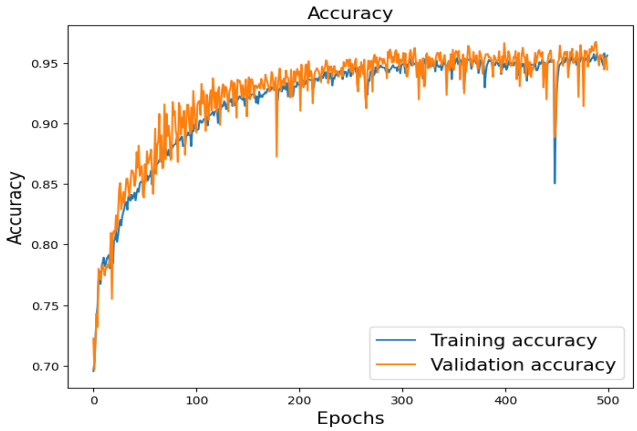

Figure 8: Training and validation accuracy of the proposed model

The proposed approach involves training a deep fake detection model over 500 epochs using the Adam optimizer, a popular and efficient method known for its adaptive learning rate capabilities. The model consistently achieves an accuracy of 94.72%, demonstrating its resilience and efficacy in recognizing and analyzing patterns in the data. This high accuracy shows how well the algorithm distinguishes between edited and actual videos. The model's remarkable performance highlights its reliability and efficiency, making it highly suitable for practical applications due to its ability to consistently maintain a high level of accuracy.

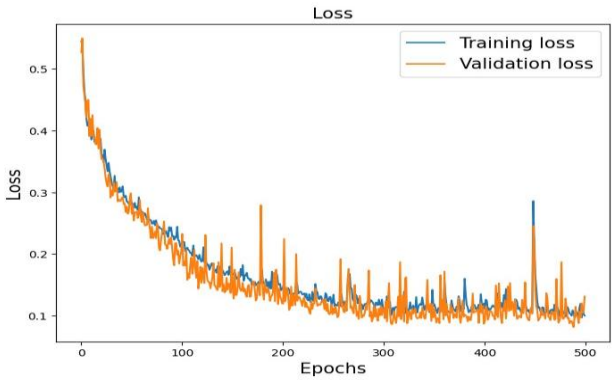

Figure 9: Training and validation loss of the proposed model

The proposed deep fake detection model, trained over 500 epochs with the Adam optimizer, is illustrated in Figure 9, along with its training and validation losses. The model consistently decreased its loss during training, indicating its increasing proficiency in minimizing the differences between predicted and actual outcomes. The model achieved high accuracy, with an average loss level of 0.1591. The significant reduction in loss is likely attributed to the Adam optimizer's adaptive learning rates and parameter updates, which enabled the model to converge effectively to the optimal solution.

Performance evaluation metrics:

The deep fake detection model is assessed using the following performance metric: Accuracy, Precision, Recall, F1-score, Specificity, and Area Under Curve (AUC).

i) Accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+TN+FP} \tag{8}$$

The percentage of true positive (TP) and true negative (TN) outcomes across all examined cases is known as accuracy. It offers a broad indication of the model's effectiveness.

ii) Precision:

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Precision indicates the percentage of positive predictions that are actually correct. The accuracy of the model's positive predictions is its main focus.

iii) Recall

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{10}$$

Recall, also known as the True Positive Rate or Sensitivity, gauges how well the model can detect positive occurrences. It is theproportion of all actual positive observations that are correctly anticipated.

iv) F1-Score

$$F1 - score = 2 \times \frac{FN}{TP+FN} \tag{11}$$

A compromise between precision and recall is offered by the F1-score. It is the two measurements' harmonic mean, and it is especially helpful when the costs of false positives and false negatives vary.

v) Specificity

$$Specificity = \frac{TN}{TN+FP} \tag{12}$$

The percentage of real negative outcomes that are true negatives is known as specificity. Also referred to as the True Negative Rate, it shows how well the model can detect negative situations.

vi) AUC

The True Positive Rate is plotted against the False Positive Rate using a ROC curve, and AUC is measured. With larger values signifying greater performance, it offers a single metric to assess the model's performance across various threshold settings.

$$AUC = \sum \frac{(TPR[i]+TPR[i+1])}{2} * FPR[i+1] - FPR[i]) \tag{13}$$

The overall effectiveness of the deep fake detection model is illustrated in Figure 10 below.

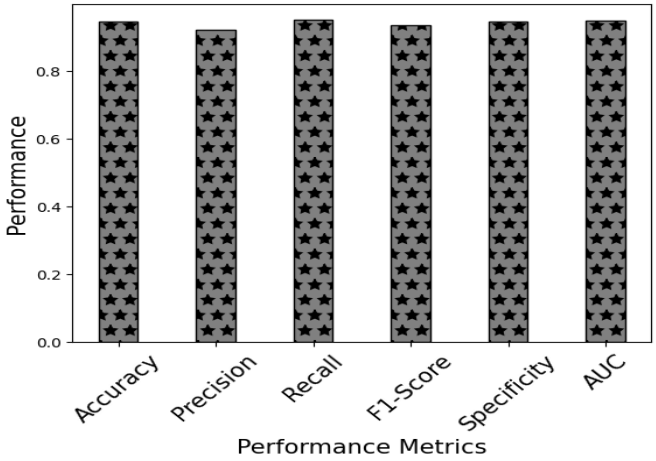

Figure 10: Overall performance metrics of the proposed model

The proposed deep fake detection strategy boasts an impressive accuracy rate of 94.72%, demonstrating its capability to differentiate between real and manipulated content. The precision of 92.09% specifies that most positive predictions made by the model are correct. With a recall rate of 95.06%, the model successfully identifies the majority of actual positive cases. The model's performance is thoroughly assessed by the F1-score of 93.55%, which strikes a balance between recall and accuracy. The model also attains an AUC of 94.78% and a specificity of 94.50%, highlighting its accuracy and reliability. These metrics highlight the method's effectiveness and reliability in deep fake detection. The detailed performance features of the proposed approach are accessible in table 1 below.

Table 1: Performance metrics of the proposed approach

Parameters	Performance (%)
Accuracy	94.72
Precision	92.09
Recall	95.06
F1-score	93.55
Specificity	94.50
AUC	94.78

A 2x2 confusion matrix is employed to access the model performance of a deepfake detection model in figure 11. It compares the actual ground truth labels with the model's predictions. The rows indicate the true labels (Fake or Real), while the columns display the predicted labels. The diagonal entries (4655 for True Positives and 3154 for True Negatives) reflect the correct predictions. In contrast, the off-diagonal values (271 for False Positives and 164 for False Negatives) indicate incorrect predictions. By examining the confusion matrix, one can evaluate the model's accuracy, precision, recall, and other relevant performance metrics to gauge its effectiveness in detecting deepfake images.

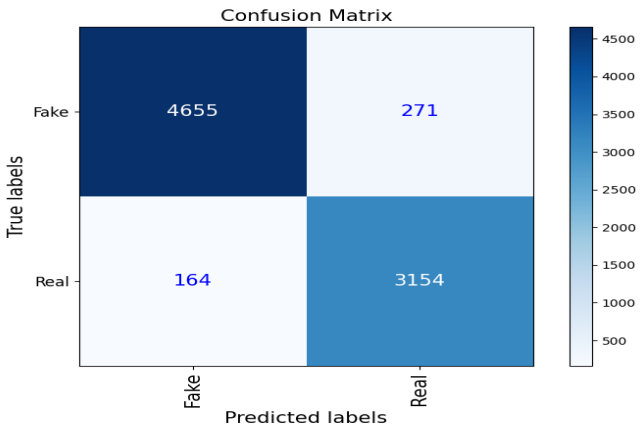

Figure 11: Confusion matrix

One popular visualisation technique in machine learning, especially for binary classification tasks like identifying deepfakes, is the Receiver Operating Characteristic (ROC) curve. The curve shows the False Positive Rate (FPR) on the x-axis, which is the percentage of real photos that are incorrectly identified as deepfakes. As a proportion of correctly detected deep fake images, the True Positive Rate (TPR) is represented on the y-axis. A diagonal line on the graph represents a random classifier, while a curve that approaches the top-left corner indicates better model performance. In this scenario, the ROC curve has an AUC of 0.95, signifying that the model is highly effective at differentiating between deepfake and real images. The following figure 12 shows the ROC of the proposed methodology.

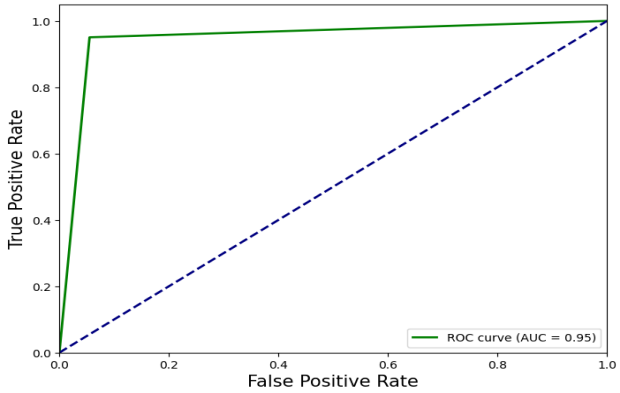

Figure 12: ROC of the proposed methodology

Comparative Analysis:

This section provides a comprehensive comparative analysis of the proposed method in relation to several existing approaches, specifically ResNet [24], Inception V3 [24], Vision Transformer (ViT) [24], and a custom Convolutional Neural Network (custom-CNN) [24]. The primary focus of this comparison is on evaluating the performance of these methods using the accuracy metric, which is a critical indicator of their effectiveness in the context of deepfake detection.

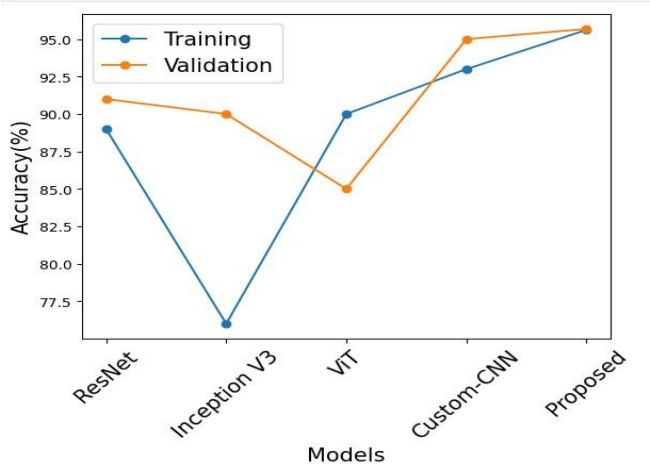

Figure 13: Comparison of Accuracy

The proposed method has demonstrated superior performance in deepfake detection, achieving an impressive accuracy rate of training and validation is 94.72%. It outperformed several established models, including ResNet, Inception V3, Vision Transformer (ViT), and a Custom CNN. Specifically, ResNet achieved 91% validation accuracy and 89% training accuracy. The accuracy of Inception V3 was 90% for validation and 76% for training. In contrast to the Custom CNN, which obtained training accuracy of 93% and validation accuracy of 95%, ViT reported training accuracy of 90% and validation accuracy of 85%. Visual comparisons with the other models further emphasise the benefits of the suggested approach, demonstrating its potential as a trustworthy tool for deepfake identification, especially in situations where high classification accuracy is essential. The enhancements in performance are also

illustrated in the accompanying figure 13, providing a clear representation of the method's effectiveness compared to its peers.

Discussion

The proposed method has demonstrated exceptional performance in deepfake detection, achieving an impressive accuracy rate of 94.72% for both training and validation. This marks a significant improvement over several established models, such as ResNet, Inception V3, Vision Transformer (ViT), and a Custom CNN. In particular, ResNet achieved 91% validation accuracy and 89% training accuracy. By comparison, Inception V3 had a 90% validation accuracy and a 76% training accuracy. While the Custom CNN fared somewhat better with a training accuracy of 93% and a validation accuracy of 95%, ViT obtained 90% training accuracy and 85% validation accuracy. The advantages of the proposed method are further emphasized through visual comparisons with these models, illustrating its potential as a reliable tool for deepfake detection, especially in applications where high classification accuracy is essential. The accompanying visuals provide a clear representation of the method's effectiveness, showcasing the performance enhancements achieved through this approach. Overall, the results underscore the proposed method's capability to outperform existing models, positioning it as a strong candidate for future implementations in deepfake detection systems.

CONCLUSION

In conclusion, the proposed methodology for detecting facial deepfakes significantly enhances detection capabilities in the face of increasingly sophisticated manipulation techniques. By integrating a Multi-cascaded Face Artefact Detection approach with an Xception Convoluted LSTM Network, this study effectively addresses the limitations of existing systems. The systematic pre-processing of input videos, combined with advanced face detection and landmark extraction methods, allows for accurate identification of subtle manipulations often overlooked by traditional detection systems. The outstanding performance metrics achieved on the FaceForensics++ dataset 94.72% accuracy, 92.09% precision, 95.06% recall, 93.55% F1-score, 94.50% specificity, and 94.78% AUC establish the robustness and reliability of the proposed approach. These results affirm its potential as a valuable tool for safeguarding against the risks posed by deepfake technology across various sectors. Future studies can concentrate on improving the model's capacity to adjust to fresh and developing deepfake methods, making sure it continues to work well against ever-more-advanced manipulations.

REFERENCES

[1] Delfino, R. A. (2022). Deepfakes on trial: a call to expand the trial judge's gatekeeping role to protect legal proceedings from technological fakery. Hastings LJ, 74, 293.

[2] Botha, J., & Pieterse, H. (2020, March). Fake news and deepfakes: A dangerous threat for 21st century information security. In ICCWS 2020 15th International Conference on Cyber Warfare and Security. Academic Conferences and publishing limited (p. 57).

[3] Gramigna, R. (2024). Preserving Anonymity: Deep-Fake as an Identity-Protection Device and as a Digital Camouflage. International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique, 37(3), 729-751.

[4] Sharma, J., Sharma, S., Kumar, V., Hussein, H. S., & Alshazly, H. (2022). Deepfakes Classification of Faces Using Convolutional Neural Networks. Traitement du Signal, 39(3).

[5] O'Halloran, A. (2021). The Technical, Legal, and Ethical Landscape of Deepfake Pornography. Cs. Brown. Edu.

[6] 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.

[7] Masood, M., Nawaz, M., Malik, K. M., Javed, A., Irtaza, A., & Malik, H. (2023). Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. Applied intelligence, 53(4), 3974-4026.

[8] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. Information Fusion, 64, 131-148.

[9] Huang, Y., Juefei-Xu, F., Guo, Q., Liu, Y., & Pu, G. (2023). Dodging deepfake detection via implicit spatial-domain notch filtering. IEEE Transactions on Circuits and Systems for Video Technology.

[10] Maltby, H., Wall, J., Glackin, C., Moniri, M., Cannings, N., & Salami, I. (2024). A Frequency Bin Analysis of Distinctive Ranges between Human and Deepfake Generated Voices.

[11] Gao, J., Micheletto, M., Orrù, G., Concas, S., Feng, X., Marcialis, G. L., & Roli, F. (2024). Texture and artifact decomposition for improving generalization in deep-learning-based deepfake detection. Engineering Applications of Artificial Intelligence, 133, 108450.
[12] Kim, E., & Cho, S. (2021). Exposing fake faces through deep neural networks combining content and trace feature extractors. IEEE Access, 9, 123493-123503.
[13] Jung, T., Kim, S., & Kim, K. (2020). Deepvision: Deepfakes detection using human eye blinking pattern. IEEE Access, 8, 83144-83154.
[14] Alnaim, N. M., Almutairi, Z. M., Alsuwat, M. S., Alalawi, H. H., Alshobaili, A., & Alenezi, F. S. (2023). DFFMD: a deepfake face mask dataset for infectious disease era with deepfake detection algorithms. IEEE Access, 11, 16711-16722.
[15] Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A., & Alshehri, A. H. (2023). Deep fake detection and classification using error-level analysis and deep learning. Scientific Reports, 13(1), 7422.
[16] Ciamarra, A., Caldelli, R., Becattini, F., Seidenari, L., & Del Bimbo, A. (2024). Deepfake detection by exploiting surface anomalies: the SurFake approach. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1024-1033).
[17] Saikia, P., Dholaria, D., Yadav, P., Patel, V., & Roy, M. (2022, July). A hybrid CNN-LSTM model for video deepfake detection by leveraging optical flow features. In 2022 international joint conference on neural networks (IJCNN) (pp. 1-7). IEEE.
[18] Khan, S. A., & Dang-Nguyen, D. T. (2023). Deepfake Detection: Analysing Model Generalisation Across Architectures, Datasets and Pre-Training Paradigms. IEEE Access.
[19] Liu, Q., Xue, Z., Liu, H., & Liu, J. (2024). Enhancing Deepfake Detection with Diversified Self-Blending Images and Residuals. IEEE Access.
[20] Hasanaath, A. A., Luqman, H., Katib, R., & Anwar, S. (2024). FSBI: Deepfakes Detection with Frequency Enhanced Self-Blended Images. arXiv preprint arXiv:2406.08625.
[21] Ilyas, H., Javed, A., Malik, K. M., & Irtaza, A. (2023). E-Cap Net: an efficient-capsule network for shallow and deepfakes forgery detection. Multimedia Systems, 29(4), 2165-2180.
[22] Al_Dulaimi, D. A., & Ibrahim, L. M. (2023). Deepfake Detection Model Based on Combined Features Extracted from Facenet and PCA Techniques. AL-Rafidain Journal of Computer Sciences and Mathematics, 17(2), 19-27.
[23] Ma, L. H., Fan, H. Y., Lu, Z. M., & Tian, D. (2020). Acceleration of multi-task cascaded convolutional networks. IET Image Processing, 14(11), 2435-2441.
[24] Rafiqee, M. M., Qaiser, Z. H., Fuzail, M., Aslam, N., & Maqbool, M. S. (2023). Implementation of Efficient Deep Fake Detection Technique on Videos Dataset Using Deep Learning Method. Journal of Computing & Biomedical Informatics, 5(01), 345-357.