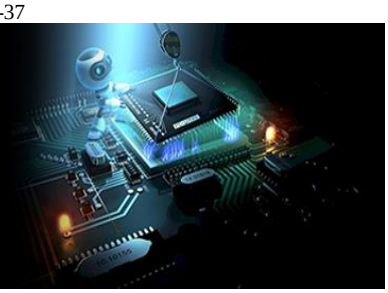
International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
Impact Factor (RJIF): 5.52
www.computersciencejournals.com/ijecs
IJECS 2025; 7(2): 30-37
Received: 05-05-2025
Accepted: 12-06-2025

Jaspreet Singh
Department of Computer Science and Engineering, Punjabi University, Patiala, Punjab, India

Dr. Madan Lal
Department of Computer Science and Engineering, Punjabi University, Patiala, Punjab, India

Dr. Kanwal Preet Singh Attwal
Department of Computer Science and Engineering, Punjabi University, Patiala, Punjab, India

Corresponding Author:
Jaspreet Singh
Department of Computer Science and Engineering, Punjabi University, Patiala, Punjab, India

A hybrid deep learning approach for deepfake detection using spatial and temporal features with attention mechanisms

Jaspreet Singh, Madan Lal and Kanwal Preet Singh Attwal

DOI: https://www.doi.org/10.33545/26633582.2025.v7.i2a.196

Abstract
Deepfake attacks threaten the authenticity of digital media, requiring strong detection methodologies to counter them. We therefore propose a deepfake detection system in which EfficientNetB3 acts as a spatial feature extractor, BiLSTM allows for temporal sequence modeling, and the self-attention mechanism creates attention on discriminative frames. The method is tested against the highly challenging Celeb-DF dataset, in which it achieves an accuracy of 85% on the test split. This also suggests that the proposed method successfully captures spatial and temporal discrepancies inside deepfake videos and therefore, is a viable candidate to analyze high-quality synthesized content. Early stop has been applied to prevent the model from overfitting the training data and enhance generalization to unseen data. The future aims of this research are to improve the robustness of the face detector and explore multimodal approaches to improve the inference accuracy further.

Keywords: Deepfake detection, EfficientNet, BiLSTM, self-attention, celeb-DF

1. Introduction

1.1 Background and Motivation

Deepfakes are synthetic media created by means of artificial intelligence-aided techniques like Generative Adversarial Networks (GANs); they have become a matter of concern in digital media authentication [3, 21]. Often, such fake videos can show individuals performing actions or making statements that they never actually did. Often, such videos are very difficult to distinguish from the actual ones with the untrained eye. With the growing popularity of deepfakes and accessibility of tools for their generation, there have been graphic developments in their proliferation, undermining trust in visual media. Trying to Deepfake detection is made difficult because generation methods keep morphing, while several manipulation techniques are being deployed [4, 7]. Older approaches to detection would not be able to lead advances in new-generation generation methods, which displays the necessity to have new-generation machine-learning-based models that are able to track and scrutinize video inconsistencies of spatial and temporal nature [6, 9]. The generation of deepfake content does not stop with the visual alone; advanced Text-to-Speech technologies threaten an equal realm of harm with deepfake-like fake voices [12]. This underlines the simultaneous need for detection techniques that apply to both synthetic visual and auditory media. In line with the stated purpose, here is an implementation of the deepfake detection system, attempting to face the issue through combinations of those deep learning approaches currently considered state-of-art.

1.2 Importance of Deepfake Detection

Deepfakes generate complex hazards worldwide, making their detection urgent. Such deepfakes could be wielded as means to promulgate misinformation and disinformation, thereby swaying human will on transient societal issues such as elections [15]. On an individual level, there is the possibility of execution of grievous loss of reputation on those who put someone into situations in which the one portrayed may not have done whatever the persona has been 'depicted' as doing (some of this has been analyzed under the banner of ethics of synthetic media) [14]. From a criminal viewpoint, they can be used to scam people by impersonating some senior member or legitimate personnel and soliciting

sensitive information or resources [7]. In an even graver national-security context, deepfakes could be misused for espionage, blackmail, or social uproar [4]. Setting up the production and distribution of deepfakes has already raised severe ethical and legal concerns, including some aspects of privacy and intellectual-property-related offenses [13]. Therefore, rendering a robust well-grounded detection mechanism.

1.3 Overview of the Approach

This paper proposes a novel deepfake detection framework comprising face detection, feature extraction, and sequence modeling to make a correct classification as real or fake. First, MTCNN performs face detection to localize the faces, thereby concentrating on the region's most prone to manipulation [10, 11]. Frames are extracted from the video with a maximum of 40 and consequently resized to 224x224 pixels for further consistent processing. Each frame is then fed to a pre-trained EfficientNetB3 network to extract high-dimensional features. The rationale behind this is EfficientNetB3 is considered one of the most efficient and accurate networks to extract complex visual patterns. Then, at the temporal modeling stage, the frame features get processed by a BiLSTM network to take into account both the past and the future context [1, 2]. Furthermore, in order to highlight the discriminative frames and thereby detecting fine-grained inconsistencies, an attention mechanism is integrated within the model [6, 9]. The final output of a dense layer with a sigmoid function then indicates the probability score for the question of whether it is a deepfake. The proposed method shows an 85% accuracy over the test set of the Celeb-DF dataset consisting of real and synthetic celebrity videos.

2. Related Work

2.1 Deepfake Generation Techniques

Deepfakes can arguably be considered a type of synthetic media creation, mostly from a GAN perspective: the generator network produces the altered content with a different facial expression or scenes, while the discriminator network attempts to judge their authenticity. In the process of winning each other over, the networks eventually engender very convincing deepfakes [3, 4]. Alternatively, an autoencoder-based face-swapping procedure can be used for deepfaking, wherein a face image is encoded to a latent representation and then decoded to swap the identity of a second input face image, thus perfectly applied in the face replacement process [13, 21]. More advanced methods consider a 3D model of a face or multimodal inputs such as audio to improve the realism and thus make detection much harder [3]. The sheer availability of these tools and their ever improvement make it an urgent matter for us to work on their detection [13, 21]. Another interesting area is facial image inpainting methods such as proposed by Wei *et al.* [16]; these methods operate with deep generative models to reconstruct missing facial regions, thus paving the way for the generation of perfectly fake deepfakes.

2.2 Deepfake Detection Methods

Born in evolving times for inventions, deepfakes stood as the center of a massive debate about detection. The age-old methods, hitherto widely known, included some sort of visual artifacts: light in consistency and color mismatch between the face and background or unrealistic movements during rotation [4, 7, 8, 13]. With successive threatening manifestations of deepfakes, the deep-learning-based detection got another push. Deep learning neural networks employ subtle spatial and temporal inconsistencies in the video to present a few desperately needed strong approaches facing challenges posed by deepfakes.

2.2.1 CNN-based Approaches

Because it extracts hierarchical spatial features present in video frames, CNN has been considered important for deepfake detection. Omar *et al.* suggested ensemble CNN-based architecture with a self-attention mechanism for better detection by assigning importance to facial regions [9]. On the contrary, Suratkar and Kazi used transfer learning to fine-tune CNN models trained beforehand on very large image datasets for deepfake classification [19]. CNN-based deepfake detection techniques primarily follow frame-wise approaches for detecting spatial inconsistencies and temporal modeling for detecting temporal inconsistencies. Ganguly *et al.* [20] incorporated visual attention into the CNN to focus on the areas that were manipulated for better detection. Ikram *et al.* [18] proposed a hybrid CNN that merges different architectures to extract a variety of features that would help in further detection. Implicit in these methods is the idea that CNNs can responsively detect spatial inconsistencies that arise probably through deepfake manipulations.

2.2.2 RNN and LSTM-based Approaches

However, temporal inconsistencies are the hallmark of these deepfake videos. Thus, for the modeling of frame-to-frame dependencies, RNN and LSTM have been accepted. Tariq *et al.* [1] described a convolutional LSTM residual network to process frame sequences so that they could better learn spatial and temporal patterns for detection. Saikia *et al.* [2] designed a CNN-LSTM-based approach, using optical transition features to identify motion inconsistencies and lend long-term dependency support to LSTMs. The methods emphasize the importance of temporal modeling in identifying dynamic artifacts in manipulated videos.

2.2.3 Attention Mechanisms

Attention goes into deepfake detection through focusing on regions or frames most discriminative of forgery. Omar *et al.* [9] introduced self-attention into their CNN ensemble in order to emphasize the primary facial features for improved detection. The authors Gu, *et al.* [6] might have applied an attention mechanism in their spatiotemporal inconsistency learning method to detect inconsistencies of space and time. Ganguly *et al.* [20] employ some form of visual attention for locating the forgery and helping the model to identify very subtle interventions. Thus, attention methods allow the model to attend to relevant information and, as a result, help them in locating deepfake artifacts.

2.2.4 Other Techniques

Emerging architectures for deepfake detection beyond CNNs/RNNs/attention mechanisms have been investigated in the literature. Wodajo and Atnafu [5] proposed a convolutional vision transformer that combines convolutional feature extractions along with transformer-based sequence modeling to create a novel means for video analysis. Yan *et al.* [17] proposed a plug-and-play framework where in video-level blending and spatiotemporal adapter

tuning take place so as to provide a framework for flexible and generalizable detection models. These new-generation methods attempt to enlarge the scope of deepfake detection beyond that of traditional arenas, concurrently with the trend of newly evolving manipulation techniques.

Deepfake detection has undergone rapid transformation with the use of deep learning to contain the menace of growing synthetic media. CNN methods perform well in extracting spatial features; RNNs and LSTMs can model temporal dynamics; attention mechanisms focus on important regions; newer transformer- or plug-and-play framework-based architectures provide a fresh perspective. The wide variety of methods truly sets a solid foundation for our implementation, synergizing these concepts for a robust deepfake detection that we will discuss next.

3. Methodology

The deepfake detection system proposed follows a structured pipeline as shown in Figure 1. The video is input to be processed for face detection, frame extraction, feature extraction, sequence modeling, and classification regarding the realness of it. This section describes each component of the pipeline, with an overview of the process presented in Figure 1 and an algorithmic description in Algorithm 1.

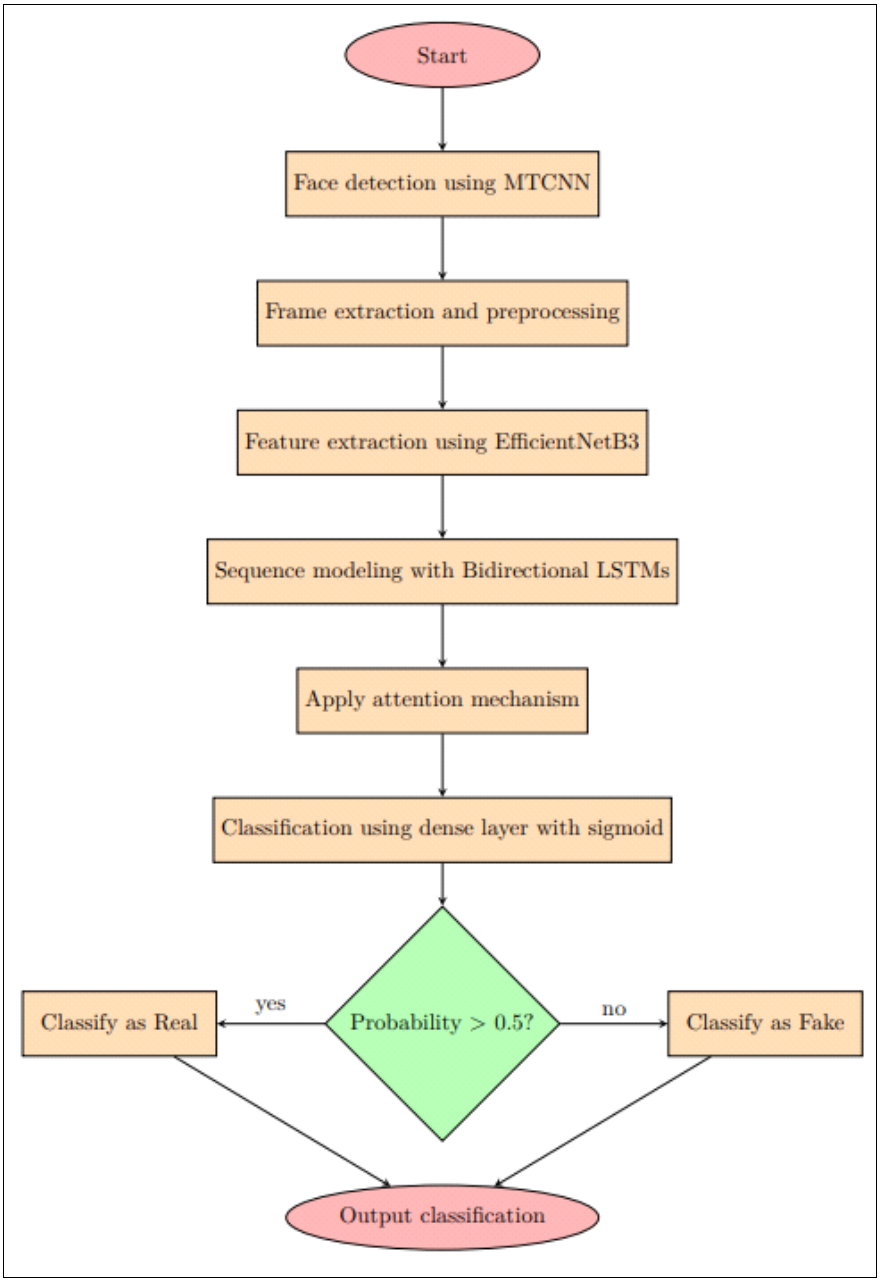


Fig 1: System Flowchart

The flowchart consists of the following steps

Start: Input video file.

- **Step 1**: Face detection using MTCNN.
- **Step 2**: Frame extraction and preprocessing (crop face or center, resize to 224x224).
- **Step 3**: Feature extraction using EfficientNetB3.
- **Step 4**: Sequence modeling with Bidirectional LSTMs.
- **Step 5**: Apply attention mechanism.
- **Step 6**: Classification using a dense layer with sigmoid activation.
- **Step 7**: Decision: If probability > 0.5, classify as Real; else, classify as Fake.
- **End**: Output classification (Real or Fake).

3.1 Dataset

3.1.1 Description of Celeb-DF Dataset

For the present study, the Celeb-DF dataset was used, wherein 408 original celebrity videos were downloaded from YouTube and 795 DeepFake videos were produced from those originals [3, 21]. Real videos include a range of people, with demographic variances in age group, ethnicity, and gender to provide a widespread representation. DeepFake videos were crafted using very sophisticated synthesis techniques, offering very visual fidelity levels and close impressions of real ones. So, in that manner, Celeb-DF provides a very tough benchmark for deepfake detection research [3, 21]. Being high-quality synthetic videos from the perspective of detection, these projects common visual artifacts such as splicing boundary, color mismatch, and so forth [3].

3.1.2 Data Splitting

Thus, datasets are stratified into train and test with an 80-20 ratio, keeping in mind the real versus fake class balance. This way, each set receives equal representation of its classes and, therefore, the model can be judged fairly while being tested. Approximately 20% of the data or about 241 videos make it to the test set, while 80%, or about 962 videos, stay with the train set. Keeping it stratified keeps real/fake label distribution intact, and this acts as an advantage for the model to generalize across the dataset.

3.2 Face Detection and Frame Extraction

3.2.1 MTCNN for Face Detection

The detection of the face is done by Multi-task Cascaded Convolutional Networks (MTCNN), a very robust method capable of detecting faces at various scales, with different poses and illumination. It is this procedure that allows MTCNN to detect faces at facial regions in video frames considered highly vulnerable to deepfake manipulations, e.g., facial expressions or lip movements. For every video, MTCNN attempts to find a face in the first few frames. Once detected, the coordinates of the bounding box are stored and reused in the coming frames to maintain consistency and keep computational overhead intact.

3.2.2 Handling Videos with No Faces Detected

The backup mechanism, meanwhile, is invoked in the unlikely event that a face goes undetected in the frame: the center square of that frame gets cropped to maintain input size consistency. Once one or more faces are detected in any frame, its bounding box is gated to extract the faces from all the succeeding frames under the assumption of minimal movement in the video sequence. This serves as an interesting trade-off for greater accuracy and efficiency, avoiding the redundancy of target detection during frame extraction. Extraction is limited to 40 frames per video; if under 40 are available, the last extracted frame is repeated to make up for 40 total. This guarantees equal inputs into the later feature extraction stage.

3.3 Feature Extraction

3.3.1 EfficientNetB3 as Feature Extractor

Every frame extracted from a video sequence is treated as a visual sample of 224x224 pixels, and the pre-trained EfficientNetB3 is used to generate 1536-dimensional embeddings [1, 2]. EfficientNetB3 captures complex visual patterns with efficiency and high accuracy and is hence, selected for deepfake detection. The network is initialized with ImageNet weights and left frozen throughout training in order to encourage the model to learn from those representations. Thus, features are average pooled across spatial dimensions to describe each frame compactly, and these serve as the input to the sequence modeling phase.

3.4 Sequence Modeling

3.4.1 Bidirectional LSTM Architecture

For capturing temporal dependency across the sequence of frames, a BiLSTM neural network is used. Two stacked BiLSTM layers-with 256 and 128 units, respectively-process a sequence of 40 frame features each of dimension 1536, returning sequences [1, 2]. The bidirection scheme will enable the model to consider past and future contexts within the video, thereby allowing it to determine temporal inconsistencies, such as unnatural movements and discontinuities, which are telltale signs of deepfake manipulations.

3.4.2 Attention Mechanism

The idea of self-attention is applied, allowing the model to focus on the discriminatory frames in the sequence [6, 9]. From the BiLSTM layer 2 output, the self-attention layer calculates attentional weights across the sequence, which form the basis of a weighted sum of the features. This attended representation is then concatenated with the BiLSTM output, so that the feature set contains sequential information as well as attentional information. Next, the combined features are averaged using global average pooling across the sequence dimension to get a fixed-size vector representing temporal dynamics and contributions by key frames.

3.5 Model Training

3.5.1 Loss Function and Optimizer

Binary cross-entropy loss functions for training purposes as it provides the optimizer with a manner of distinguishing between real and fake video data. Adam optimizer is therefore chosen, beginning with an aggressive learning rate of 1e-4, such that it converges fast and steadily. For training, a minibatch setup with a batch size 8 shall be used. This is deemed small enough not to slow the training speed while, on the other hand, being large enough not to affect the performance of the model very much.

3.5.2 Callbacks and Early Stopping

There are further callbacks to increase training optimization and prevent overfitting:

- **ModelCheckpoint:** saves the weights of a model that have the highest validation accuracy, so that the best-performing model is saved.
- **ReduceLROnPlateau:** reduces the learning rate in steps by a factor of 0.5 for 3 consecutive epochs after a plateau of validation accuracy, but until the learning rate is not lower than 1e-6.
- **EarlyStopping:** stops training if the validation accuracy hasn't improved for 7 consecutive epochs and restores weights of the best epoch.

A total of 25 epochs are allowed to run during training, though reduced by early stopping, saving computational time and potentially promoting better model performance.

3.6 Algorithm Summary

The inference process of the deepfake detection system is summarized in Algorithm 1, which outlines the steps from video input to classification output.

Algorithm 1
Deepfake Detection Inference
Input: Video file
Output: Classification (Real or Fake)

Step	Description
1	Load the video file
2	Initialize face detector (MTCNN)
3	For each frame in the video (up to 40 frames):
3a	Detect face using MTCNN
3b	If face detected, crop the face region
3c	Else, crop the center of the frame
3d	Resize the cropped frame to 224×224
3e	Store the processed frame
4	If fewer than 40 frames, pad with the last frame
5	Initialize feature extractor (EfficientNetB3)
6	For each processed frame:
6a	Extract features using EfficientNetB3
6b	Store the feature vector (1536 dimensions)
7	Initialize sequence model (BiLSTM with attention)
8	Pass the sequence of feature vectors through the BiLSTM layers
9	Apply attention mechanism to the LSTM outputs
10	Concatenate attended features with original LSTM outputs
11	Apply global average pooling over the sequence dimension
12	Pass the pooled features through a dense layer with sigmoid activation
13	Output the classification probability
14	If probability > 0.5 → classify as Real; else → classify as Fake

This algorithm utilizes the latest deep learning technologies for robust deepfake detection, whereby EfficientNetB3 is used for spatial feature extraction, BiLSTM is used for temporal modeling, and the attention mechanism is used to focus on important frames.

The components of the proposed deepfake detection system are summarized in Table 1.

Table 1: System Components

Component	Description	Parameters
Dataset	Celeb-DF	241 real, 962 fake videos; 80-20 train-test split
Face Detection	MTCNN	Detects faces; center crop if no face detected
Frame Extraction	Up to 40 frames	Padded with last frame if needed; resized to 224x224
Feature Extractor	EfficientNetB3	1536-dimensional features; ImageNet weights, frozen
Sequence Model	BiLSTM	256 and 128 units, returns sequences
Attention	Self-attention	Applied to BiLSTM output; concatenated with sequence
Training	Binary cross-entropy, Adam	Batch size: 8; learning rate: 1e-4; max 25 epochs
Callbacks	ModelCheckpoint, ReduceLROnPlateau, EarlyStopping	Save best weights; reduce LR on plateau; stop after 7 epochs

The architecture of each component, along with a list of parameters, is provided in this table to give an overview of the system design.

4. Experiments and Results

Setting Up Experiments and Security Protocols-Various experiments had to be conducted for training and evaluation of the system intended for deepfake detection. The experiments were performed in the Celeb-DF environment for the task of assigning a particular video as real or fake. The results of the experiments indicated that the pipeline was highly efficient in testing with an accuracy of 85%.

4.1 Experimental Setup: The experiments were carried out on the Celeb-DF, which consisted of 408 Real videos of celebrities from YouTube and 795 corresponding DeepFake videos. The classes of the dataset were divided into train and test sets by stratified sampling on an 80-20 basis to avoid any undesired distribution of classes in real and fake videos. Thus, the training set contained approximately 408 real and 795 fake videos (a total of 1,203), whereas the test set consisted of almost 408 real and 795 fake videos (a total of 1,203). This stratified sampling procedure ensured the classes were distributed proportionally, thereby providing better generalization for the model.

4.1.1 Hardware and Software Environment

The experiment was set to cover convolution at depth and was deployed at the Kaggle cloud-computing setup that provides free GPU acceleration for intensive computations in deep learning. Depending on the availability of the machine at the particular time, it either provided the NVIDIA GPU P100 or the V100. It ran with Python 3.8 with Tensorflow 2.4 compiled with Keras API (version 2.4); OpenCV for video processing; NumPy for numerical computations; Pandas for data manipulations; and MTCNN for face detection. Therefore, it was concluded that it is good to have something reproducible for the easy acceptance and compatibility of the deep-learning community.

4.1.2 Hyperparameters
The carefully chosen set of hyperparameters allowed the training to strike a balance between computational efficiency and performance, as summarized in Table 1.

Table 2: Hyperparameters Used in Model Training

Hyperparame	Value
Batch size	8
Maximum epochs	25
Initial learning rate	1e-4
LSTM units (first layer)	256
LSTM units (second layer)	128
Feature extractor	EfficientNetB3
Feature dimension	1536
Loss function	Binary cross-entropy
Optimizer	Adam

4.2 Training Process
Initially, batch size was chosen mainly for computational efficiency and memory constraints of the Kaggle environment. When these settings were chosen initially, the maximum number of epochs was set to 25, with early stopping trials enforced every 7 epochs, mainly to save on resources; early stopping would occur whenever validation accuracy ceased to improve.

4.2.1 Batch Size and Epochs
Also, a parameter modifying strategy was used for learning rate scheduling: the initial learning rate, 1e-4, was reduced to half via the Reduce LR On Plateau call back if validation accuracy failed to improve for 3 epochs and would continue to halve until it reached the minimum of 1e-6. This allowed for the model to learn finer representations towards the final steps of training for better convergence.

4.2.2 Learning Rate Scheduling
Given a testing accuracy rate of 0.85, the trained pipeline has achieved a fine deep classification of real versus fake videos. This type of accuracy suggests that the system is capable of catching the spatial and temporal inconsistencies being forced into deepfake videos through the help of EfficientNetB3 for the feature building and BiLSTM + attention for sequence building.

4.3 Accuracy on Test Set
The trained model reaches an 85% accuracy for the test set, showing the ability to discriminate real and fake videos. The technique performs poorly with detection of temporal and spatial inconsistencies in deepfake videos; however, EfficientNetB3 works well in feature extraction, and BiLSTM with attention works well in sequence modeling.

4.4 Training and Validation Curves
Figures 2 and 3 show the deepfake detection model performance for training and validation, respectively, over 25 epochs. Referring to Figure 2, the training accuracy increases faster than the validation accuracy, starting at about 0.65 at the initial epoch and reaching nearly 0.95 by epoch 25, demonstrating a strong learning capability of the model on training data. The validation accuracy on the other hand increases slowly from 0.65 to 0.85 at epoch 25 but remains more or less constant at 0.84-0.85 in the initial epochs.

Nonetheless, the final test accuracy of 85% serves to prove that it generalizes reasonably well on unseen samples, holding up well to perform competitively on the challenging Celeb-DF datasets. Early stopping (after 7 epochs of no improvement in validation accuracy) may have diminished the risk of overfitting, thus ensuring model robustness.

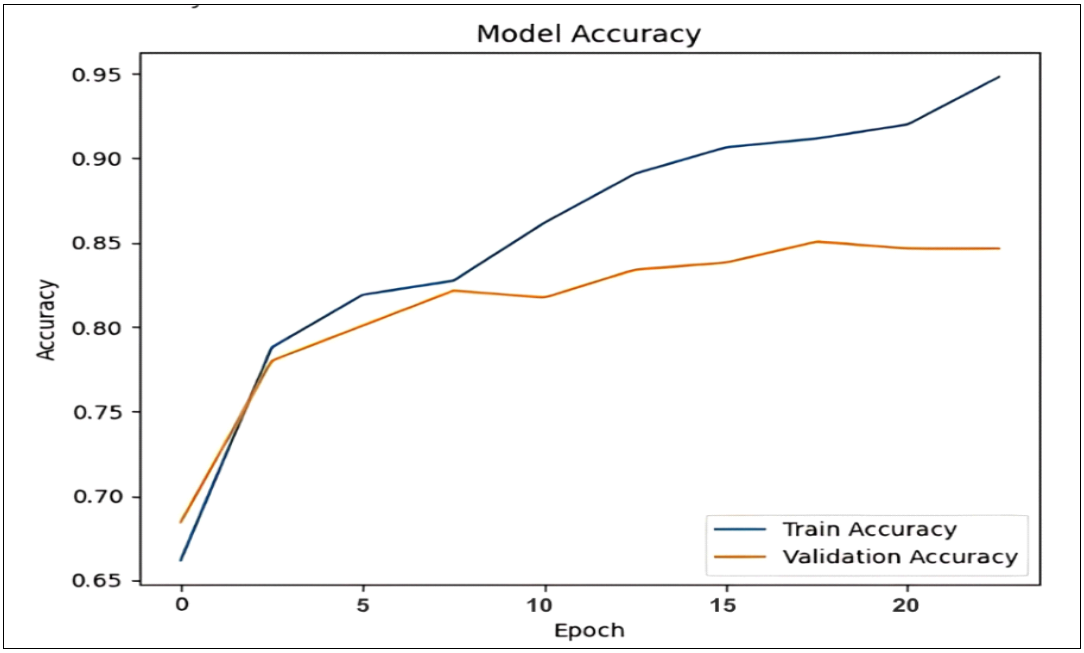


Fig 2: Training and Validation Accuracy over Epochs

The stabilization of validation accuracy for the model at around 85% suggests that the model has learned to detect deepfakes, which is also confirmed by the test set performance.

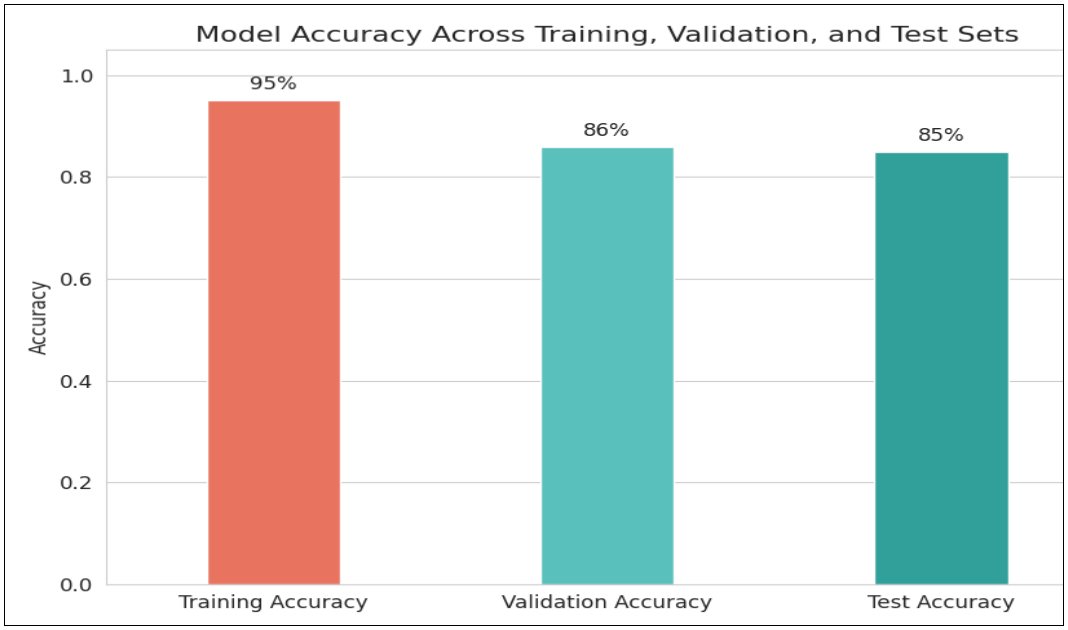


Fig 3: Bar Chart of Model Accuracy across Training, Validation, and Test Sets

The bar chart compares the accuracy performance of a deepfake detection model throughout various phases: training, validation, and test set. The training accuracy is about 95%, which shows a strong learning capability from the training data. The validation and test accuracies stabilize to close to about 85%, indicating that the model well generalizes to fresh data.

5. Discussion

The proposed deepfake detection system shows efficacious performance with an accuracy of 85% on the Celeb-DF test set. This indicates how well EfficientNetB3 acts in spatial feature extraction, whereas Bidirectional Long Short-Term Memory (BiLSTM) layers and a self-attention mechanism are used for temporal sequence modeling. This system exploits static visual cues and dynamic inconsistencies simultaneously to detect suspicious features that indicate deepfake manipulations, thus making it a reliable tool to detect high-quality synthesized videos from the Celeb-DF dataset.

In other words, the system has yielded wonderful results, yielding test accuracies marring some 85%: I say a real epochal feat because Celeb-DF happens to be one of the toughest testing sets with high-fidelity deepfakes hardly carrying any identifiable visual distortive artifact. But in consideration of a real-life scenario, a 14% margin for errors: some videos are misclassified, and then the very existence of these videos' tau their discussion. In contrast with these errors, the argument against building an even better system capable of being relied upon in real-world applications, such as journalism, legal investigation, or online content verification, can only be regarded as weak. In social media, a false positive censoring a real video is tantamount either to censorship or unjust damage to a person's reputation; a false negative allows a misleading video to permeate.

First and foremost, a principal drawback of the system being heavily reliant on Multi-task Cascaded Convolutional Networks (MTCNN) for face detection. When there are partially visible or obscured faces, or the faces are completely absent in the video, cropping is resorted to at the center of the frame, which may not have relevant features for correct classification. Hence, relying on this would hinder or prevent the system from operating properly for peculiar videos, for instance, the surveillance videos or multiple person subject videos. Moreover, since the system was trained and tested on the Celeb-DF dataset containing mostly videos of celebrities, this feature could constrain this system from generalizing to other types, like non-celebrity videos filmed under different conditions or the new emerging deepfake styles. Deepfake generation techniques are rapidly evolving, further hampering the situation; as soon as any manipulation appears in the training data, the system will not be able to detect it.

The created system offers a highly structured and reproducible framework into which the deepfake-detection research can be thrust. Incorporating modern and yet established deep-learning techniques with the real-world problem of missing faces, this serves as a sound basis for further enhancements. The ability to and efficiently handle sequences of videos and classify them with good accuracy on a truly arduous dataset is proof of the strong practicality potential the system holds, should the limitations be resolved.

6. Future Work

Some other considerations should be analyzed more extensively in order to theoretically improve the applicability of a deepfake detection system. Ideally, more emphasis should be placed on bettering face detection. For example, should a good face detection algorithm fail to capture the face more than occasionally, one might want to consider hybrid solutions for these problematic cases where the occluded or partially visible faces are shown, with low resolution, in the videos. Other types of preprocessing could be developed instead of relying solely on the facial regions; for instance, full-frame content or contextual elements might be exploited for application to other video types.

Second, multiple modalities can help increase detection accuracy. Usually, when a deepfake is real, visual inconsistencies may be introduced in conjunction with the audio; lip movements do not match or the voice does not

sound quite right. Thus, audio analysis would basically make for a multi-modal detection scheme where the two types of analyses complement each other, relying on discrepancies in audio and visual cues as evidence of manipulation.

Third, because of the rapid-forward nature of deepfake generation technologies, constant updating of models is required. A system re-training with the relatively recent dataset containing more novel deepfake methods or by means of adversarial training will arm the system adequately to counter new manipulations, and maintaining the adaptation must be carried on for the system to be relevant in the shifting carpet of deepfake technology.

With these methods endorsed, deepfake detection might resist its limitations.

References

1. Tariq S, Lee S, Woo SS. A convolutional LSTM based residual network for deepfake video detection. arXiv preprint arXiv:2009.07480. 2020.
2. Saikia P, Dholaria D, Yadav P, Patel V, Roy M. A hybrid CNN-LSTM model for video deepfake detection by leveraging optical flow features. In: 2022 International Joint Conference on Neural Networks (IJCNN). IEEE; 2022. p. 1-7.
3. Zhang T. Deepfake generation and detection: a survey. Multimed Tools Appl. 2022;81(5):6259-6276.
4. Yu P, Xia Z, Fei J, Lu Y. A survey on deepfake video detection. IET Biom. 2021;10(6):607-624.
5. Wodajo D, Atnafu S. Deepfake video detection using convolutional vision transformer. arXiv preprint arXiv:2102.11126. 2021.
6. Gu Z, Chen Y, Yao T, Ding S, Li J, Huang F, Ma L. Spatiotemporal inconsistency learning for deepfake video detection. In: Proc 29th ACM Int Conf on Multimedia. 2021. p. 3473-3481.
7. Heidari A, Jafari Navimipour N, Dag H, Unal M. Deepfake detection using deep learning methods: A systematic and comprehensive review. WIREs Data Min Knowl Discov. 2024;14(2):e1520. doi:10.1002/widm.1520.
8. Kaur A, Noori Hoshyar A, Saikrishna V, Firmin S, Xia F. Deepfake video detection: challenges and opportunities. Artif Intell Rev. 2024;57(6):159.
9. Omar K, Sakr RH, Alrahmawy MF. An ensemble of CNNs with self-attention mechanism for DeepFake video detection. Neural Comput Appl. 2024;36(6):2749-2765.
10. Lim S, Gwak Y, Kim W, Roh JH, Cho S. One-class learning method based on live correlation loss for face anti-spoofing. IEEE Access. 2020;8:201635-1648.
11. Liu W, Wei X, Lei T, Wang X, Meng H, Nandi AK. Data-fusion-based two-stage cascade framework for multimodality face anti-spoofing. IEEE Trans Cogn Dev Syst. 2021;14(2):672-683.
12. Mahum R, Irtaza A, Javed A. EDL-Det: a robust TTS synthesis detector using VGG19-based YAMNet and ensemble learning block. IEEE Access. 2023;11:134701-16.
13. Malik A, Kuribayashi M, Abdullahi SM, Khan AN. DeepFake detection for human face images and videos: a survey. IEEE Access. 2022;10:18757-18775.
14. Oulad-Kaddour M, Haddadou H, Vilda CC, Palacios-Alonso D, Benatchba K, Cabello E. Deep learning-based gender classification by training with fake data. IEEE Access. 2023;11:120766-120779.
15. Shahid W, Li Y, Staples D, Amin G, Hakak S, Ghorbani A. Are you a cyborg, bot or human?—a survey on detecting fake news spreaders. IEEE Access. 2022;10:27069-27083.
16. Wei J, Lu G, Liu H, Yan J. Facial image inpainting with deep generative model and patch search using region weight. IEEE Access. 2019;7:67456-67468.
17. Yan Z, Zhao Y, Chen S, Guo M, Fu X, Yao T, *et al.* Generalizing deepfake video detection with plug-and-play: video-level blending and spatiotemporal adapter tuning. In: Proc CVPR Conf Comput Vis Pattern Recognit. 2025. p. 12615-12625.
18. Ikram ST, Chambial S, Sood D. A performance enhancement of deepfake video detection through the use of a hybrid CNN deep learning model. Int J Electr Comput Eng Syst. 2023;14(2):169-178.
19. Suratkar S, Kazi F. Deep fake video detection using transfer learning approach. Arab J Sci Eng. 2023;48(8):9727-9737.
20. Ganguly S, Mohiuddin S, Malakar S, Cuevas E, Sarkar R. Visual attention-based deepfake video forgery detection. Pattern Anal Appl. 2022;25(4):981-992.
21. Singh J, Lal M, Attwal KPS. Deep learning techniques for deep fake identification: a review. Int J Comput Artif Intell. 2025;6(1):246-256. doi:10.33545/27076571.2025.v6.i1d.159.