Babylonian Journal of Machine Learning
Vol.2026, **pp**. 110–142
DOI: https://doi.org/10.58496/BJML/2026/012; ISSN: 3006-5429
https://mesopotamian.press/journals/index.php/BJML

Research Article

Deep Learning-Based Pre-Processing Pipeline and Hybrid Vision Transformer Model for Deepfake Video Detection

Harith A. Hussein[1,2*], Khalid Shaker [3], Salwani Abdullah[4]

[1] *Department of Computer Science, College of Computer Science and IT, University of Anbar, Iraq.*

[2] *Department of Computer Science, College of Computer Science and Mathematics, Tikrit University, Iraq.*

[3] *Department of Information Systems, College of Computer Science and IT, University of Anbar, Iraq.*

[4] *Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia.*

ARTICLE INFO

Article History
Received 10 Jun 2026
Revised 23 Jul 2026
Accepted 13 Aug 2026
Published 19 Sep 2026

Keywords
Deepfake Detection,
Multi-scale ResNet,
Hypergraph Convolutional,
Multi-scale Attention Network,
Transformation,
Siberian Tiger Optimization,
Hypergraph Convolutional Vision Transformer Networks.

ABSTRACT

Deepfake technology has increased the generation of hyper-realistic synthetic images and videos using AI that pose a threat to personal privacy, integrity in business and national security. Machine Learning (ML) and Deep Learning (DL) have been extensively used in Deepfake generation and detection due their feature learning abilities and precise predictions. While DL models overtake ML in Deepfake detection, model complexity, and high resource requirements are the major limitations. Considering such limitations, this paper developed an advanced pre-processing pipeline and the detection model for improving the Deepfake detection in videos. The pre-processing pipeline model includes Adaptive Cascaded Temporal Convolutional Networks (ACTCN) for face detection and alignment, Multi-scale ResNet-based feature fusion technique for up-scaling detection face region, CNN-based Auto-Encoder for color space transformation, Deep CNN with Fast Fourier Transformation for frequency and spatial feature enhancement, 3D-CNN for temporal consistency and ResNet for noise and texture anomaly detection. Then, the Hybrid Localized Artifact Adaptively Weighted Multi-scale Attention Network-Spatio-Temporal Hypergraph Optimized Convolutional Vision Transformer Networks (HLAAWMSAN-STHGOCVT) is developed for Deepfake video Detection. This method combined the HLAAWMSAN for feature extraction from Deepfake frames and the STHGOCVT module for Deepfake detection . This framework ensures high robustness against evolving Deepfake manipulation techniques while maintaining computational efficiency. Evaluated using benchmark and real-time video data, this proposed model has achieved 99.15% accuracy and 98.21% AUC.

1. INTRODUCTION

Deepfake technology employs Artificial Intelligence (AI)-based synthetic technologies to manipulate videos and images by generating deceptive representations of real individuals. These generated videos and images are indistinguishable from true individual images. The Deepfake techniques are applied in creative and environment fields for special effects and virtual avatars [1]. Deepfake technologies have both positive and negative applications. For instance, in entertainment, it is used for dubbing and CGI enhancements in movies, AI avatars create historical re-enactments for education purposes and also generate virtual characters for gaming purposes. However, this Deepfake has been misused for malicious purposes, including spreading false information, theft of identity, reputational damage, ethical and security concerns and cyber harassment [2]. The dissemination of unverified information through online social platforms has become a critical issue and differentiating fake and real image content is challenging. These limitations necessitate the need for a Deepfake detection model to identify subtle inconsistencies in facial landmarks to ensure prompt detection and mitigate harmful content.

The traditional methods for Deepfake images include forensic analysis, Metadata analysis and visual artifacts techniques. The forensic analysis examines inconsistencies in lighting, shadows and reflections; a metadata analysis checks file metadata for evidence of tampering and visual artifacts and identifies unnatural facial movements, irregular blinking patterns and mismatched shadows. The conventional method relies on watermarking and handcrafted feature-based detection. However, these methods are ineffective due to the inability to generalize to diverse generations of Deepfake

*Corresponding author. Email: har22c1015@uoanbar.edu.iq

content, fail to gather intricate patterns and are time-consuming with labour-intensive analysis [3]. Due to these limitations, the advent of ML and DL methods are employed to automate and enhance Deepfake detection. ML methods such as Support Vector Machines (SVM), Random Forests, and Decision Trees are applied to gather the handcrafted features to differentiate Deepfake from real data. The DL models such as RNN and transformer-based networks, are applied to analyse the Deepfake-generated data at various levels of abstraction [4]. Yet, the ML and DL techniques have high false positive rates, fails to identify subtle manipulation artifacts and lack performance due to noisy and unseen data. The pre-processing technique is important in detecting the Deepfake but filtering and alignment techniques eliminate the important details and add noise that degrades the detection models [5]. To address these limitations, a hybrid method that increases pre-processing, feature extraction, robustness and computational efficiency is developed for Deepfake detection.

This paper developed a DL-based framework called Hybrid Localized Artefact Adaptively Weighted Multi-scale Attention Network-Spatio-Temporal Hypergraph Optimized Convolutional Vision Transformer Networks (HLAAWMSAN-STHGOCVT) model. The pre-processing pipeline method improves the input data quality and helps in the effective identification of manipulated content for detection. The pre-processing pipeline model used CNN with the ResNet model as a backbone network to increase the spatial, temporal and frequency-based learning. The manipulated facial regions are detected using the Adaptive Cascaded Temporal Convolutional Networks (ACTCN) method. The ACTCN model focuses on manipulated regions and ignores irrelevant background details by capturing spatial and temporal details for face detection and alignment. The Multi-scale ResNet-based Feature Fusion (MSRFF) model is applied to enhance the image resolution, which helps to expose pixel-level inconsistencies and synthetic artifacts. For colour space transformation, the CNN-based Auto-Encoder for RGB-to-YCbCr Transformation is applied to transform the images to different colours and identify colour inconsistencies in the generated image. The CNN-based Enhancement using Laplacian filters and Fourier Transform (FFT) techniques is employed for feature enhancement by identifying the spatial features like edges and textures. The 3D-CNN for Frame-to-Frame Analysis maintains the temporal consistency of the video frames and analysis of the movements. Deepfake generators leave unnatural noise patterns and texture artifacts, which are detected by using the ResNet model. The detection model includes two components HLAAWMSAN for feature extraction and the STHGOCVT for detection. The HLAAWMSAN framework captures subtle localized Deepfake artifacts using an adaptive multi-scale attention mechanism. The STHGOCVT uses Hypergraph Convolutional Networks (HGCN) and Vision Transformers (ViT) to model spatial and temporal Deepfake inconsistencies. The model leverages the strengths of CNN, Transformers, and Hypergraph-based learning to improve detection accuracy and ensure high robustness against evolving Deepfake manipulation techniques. The contributions of the research includes

1. A HLAAWMSAN model selectively emphasizes vulnerable manipulated regions and suppresses identity and background bias, which improves generalization.
2. A STHGOCVT jointly models higher-order spatial relationships and long-range temporal dependencies using hypergraph convolution and ViT mechanisms.
3. An optimization-driven detection strategy employing STO to automatically tune model hyperparameters, which increases convergence stability and detection robustness.
4. A modular and scalable framework balances detection accuracy and computational efficiency that is suitable for benchmark datasets and real-time Deepfake analysis.

2. RELATED WORKS

Deep learning-based methods have been a major trend for the effective detection of Deepfakes video manipulation in recent years. Zheng [6] proposed a novel MesoNet framework for Deepfake video detection with a novel pre-processing module for better feature extraction. They concentrated on improving the facial image inputs and increasing color channel separation to improve the ability to deal with compression artifacts in the videos, especially on Celeb-DF and FaceForensics++ (FF++) datasets. The system uses spatial level feature representation, frame extraction, chi-distance calculation and constructing histograms to differentiate real content from manipulated content. The model exhibited higher reliability and competitiveness in the benchmark datasets, with remarkable accuracy and AUC metrics; it is still challenging with learning to manipulate patterns that are highly complex and subtle with the lightweight MesoNet backbone, which has limited convolutional depth.

In 2022, Deng et al. [7] introduced a study “Deepfake Video Detection Based on EfficientNet-V2 Network”, which provides an effective Net-V2 model to capture the features and recognize the videos. This model is focused on identifying whether the video is fake or real with facial cropping, pixel quality filtering, and size adjustment. It achieved an accuracy of 97.90% and an error rate of 6.76% on the FF++ dataset. It handled issues to discover hidden facial areas.

In 2022, Kuang et al. [8] presented a paper titled “A dual-branch neural network for Deepfake video detection by detecting spatial and temporal inconsistencies”, which provides an EfficientNet-BiLSTM model for feature extraction and an SVM algorithm. This model is applied to detect Deepfake with face cropping, frame extraction, resizing, and optical flow methods. The suggested method obtained accuracy of 98.21% and 98.93% on the FF++ and Celeb-DF

datasets, respectively. Yet, the algorithm incorporates EfficientNet and Bi-LSTM, which might cause additional computational complexity.

In 2022, RAZA et al. [9] presented a study entitled "A novel deep learning approach for deepfake image detection", which utilized a novel Deepfake Predictor (DFP). This model incorporates VGG-16 and CNN to extract features and detect videos to identify whether they are fake or original. The DFP method ensured 94% accuracy, 95% precision, 91.5% recall and 93.46% F-score on a Deepfake dataset from the Department of Computer Science at Yonsei University, that consist of 960 images. But, it obtains high training and inference time, which raises the computational complexity. In 2022, Chen et al. [10] suggested a study entitled "Detecting Deepfake videos based on spatiotemporal attention and convolutional LSTM", which presents an Xception network with convolutional layers for feature extraction, and an LSTM is applied to detect Deepfake through frame extraction and face cropping methods. The Xception-ConvLSTM model obtained accuracy of 99%, 99.93%, and 92.43% and AUC of 0.9920, 0.9991, and 0.9398 on FF++, CelebDF, and DFDC datasets, respectively. But, the method incorporates various DL frameworks, which increases the computational complexity.

In 2023, Pang et al. [11] suggested a study titled "MRE-Net: Multi-rate excitation network for deepfake video detection", which presents a Multi-Rate Excitation Network (MRE-Net) that contains Momentary inconsistencies in data (MIE) and Longstanding inconsistencies in data fully connected layers to capture features for classifying the videos. It detects whether it is original or fake along with frame extraction, face recognition, image resizing, bipartite resampling, and noise removal models. The MRE-Net attained 97.76%, 91.60%, 85.61%, 97.35%, and 85.33% accuracy and 99.57%, 96.55%, 91.23%, 99.75%, and 86.59% AUC on FF++ (c30 and c40), Celeb-DF, DFDC, and Wild Deepfake datasets, respectively. But, it utilizes large pre-processing procedures, and it increases the execution time. In 2023, Yu et al. [12] introduced a study titled "MSVT: Multiple spatiotemporal views transformer for Deepfake video detection". This model suggested the Multiple Spatiotemporal View Transformers (MSVT) model for capturing features and identifying videos, recognizing Deepfake through frame extraction methods. The MSVT achieved 98.91%, 98.83%, 95.95%, 99.73%, and 98.96% accuracy and 99.95%, 99.86%, 99.17%, 99.87%, and 99.32% AUC on FF++, DFD, DFDC, Celeb-DF-v2, DF 1.0 Raw, and WildDeepfake datasets, respectively. But the GLT hybridizes both the global transformer and local transformer, and it provides the memory cost.

In 2023, Zaho et al. [13] presented a study titled "ISTVT: interpretable spatial-temporal video transformer for deepfake detection", which provides an interpretable spatial-temporal video transformer model. The model achieved 93.8%, 84.1%, 74.2%, 99.3%, and 98.6% AUC-ROC on FF++, Celeb-DF, DFDC, FSh, and DFo datasets, respectively. Yet, the method lacks diversity and suffers from overfitting problems. In 2023, Deng et al. [14] suggested a study titled "Cascaded network based on EfficientNet and transformer for Deepfake video detection", which presents an improved EfficientNetV2S to capture the features and a vision transformer to detect the videos to check whether it is fake or original, along with frame extraction and facial cropping. This algorithm obtained 0.9784 and 0.9956 AUC, 96.75% and 92.16% accuracy, and 96.721% and 92.22% F1 scores on the FF++ and DFDC datasets, respectively. In 2023, Lu et al. [15] recommended a study titled "Deepfake Video Detection Based on Improved CapsNet and Temporal-Spatial Features", which presents CapsNet and Temporal-Spatial Features (iCapsNet-TSF). It provides the VGG-16 model to extract facial features and identify the videos to categorize the Deepfake. This algorithm ensured 94.07% accuracy, 93.65% precision, 94.22% recall, 93.3% F1 value, and 0.1769 cross-entropy loss on the Celeb-16 dataset. Still, it has a problem with inference time.

In 2023, Masud et al. [16] presented a study titled "LW-DeepFakeNet: a lightweight time-distributed CNN-LSTM network for real-time DeepFake video detection", which designs the LW-DeepfakeNet model. It incorporates the VGG-16 model for feature extraction, and the LSTM utilized the features to classify the videos for Deepfake detection. The LW-DeepfakeNet obtained 99.24% accuracy and 99.52% AUC on the Celeb-DF dataset. In 2024, Qadir et al. [17] designed a study titled "An efficient Deepfake video detection using robust deep learning", which provides ResNet along with a Swish activation function. It is utilized to mine the features, and the Bi-LSTM model is focused on detecting Deepfake. The ResNet-Swish-BiLSTM scheme obtained 99.25%, 98.23%, 95.8%, and 99% precision, 97.89%, 97.8%, 97.66, and 99.2% recall, 96.3%, 92.9%, 89.56%, and 98.8% F1 score, and 98.9%, 98.0%, 96.2%, and 98.6% accuracy on FF++(FS), FF++(F2F), FF++(NT), and DFDC datasets, respectively.

In 2024, Javed et al. [18] developed a study titled "Real-time deepfake video detection using eye movement analysis with a hybrid deep learning approach", which provides a hybrid DL method via MesoNet4 and ResNet-101. It collected attributes and classified the Deepfake from the specified dataset. It ensured accuracy by 0.9873, 0.9689, and 0.9790, 0.9871, 0.9754, and 0.9797 precision, 0.9944, 0.9729, and 0.9923 recall, and 0.9907, 0.9742, and 0.9859 F1 score on FF++, v1, and Celeb-DF v2 datasets, respectively. Yet, it has a problem with convergence speed. In 2024, Alhaji et al. [19] presented a study entitled "An approach to deepfake video detection based on ACO-PSO features and deep learning", which provides an Ant Colony Optimization with Particle Swarm Optimization (ACO-PSO) for feature selection. The models include Support Vector Machine (SVM), decision tree (DT), and k-nearest Neighbour (KNN)

analyzed to categorize the Deepfake. The ACO-PSO scheme provided a sensitivity of 98.20%, specificity of 98.10%, precision of 97.90%, accuracy of 98.30%, and an F1 score of 98.70% on the DFDC dataset.

In 2024, Omar et al. [20] introduced a study entitled "An ensemble of CNNs with self-attention mechanism for DeepFake video detection", which designed the CoAtNet model. It is focused on Deepfake classification and ensured 97.70%, 91.69%, 95.74%, 79.32%, and 97.81% accuracy and 99.70%, 97.49%, 98.90%, 87.62%, and 99.74% on DF, F2F, FS, NT, and Celeb-DF datasets, respectively. However, this model has an issue with the risk of overfitting. In 2024, Ramadhani et al. [21] suggested a study entitled "Improving video vision transformer for deepfake video detection using facial landmark, depth-wise separable convolution and self-attention," which improves video vision transformers (ViViT). This model focused on extracting features and detecting deepfakes using face landmark detection and data augmentation models. The ViViT method ensured 88.03%, 88.03%, 94.14%, and 90.27% accuracy and 92.5%, 90.23%, 97.39%, and 93.54% F1 scores on the Celeb-DF V2, DFDC, Deepfake TIMIT, and FF++ datasets, respectively.

In 2024, Cunha et al. [22] suggested a study titled "Video deepfake detection using Particle Swarm Optimization improved deep neural networks", which provides EfficientNet B0 and EfficientNet-GRU. This model is applied to extract the features to classify the videos for detecting deepfakes, and PSO is used to enhance the performance. This method ensured 0.9382 and 0.9512 accuracy, 0.9208 and 0.9886 precision, 0.9912 and 0.9566 recall, and 0.9141 and 0.9346 AUC on Celeb-DFv2 and DFDC datasets, respectively. In 2024, Kaddar et al. [23] suggested a study titled "Deepfake detection using spatiotemporal transformer", which provides a ViT model with cropping models. Xception is focused on capturing the features, and the vision transformer uses these features. This model is focused on classifying videos for detecting deepfakes with face-cropping models. The HCiT ensured 96.00%, 97.82%, 95.85%, 86.29%, 76.03%, and 76.03% accuracy and 93.86%, 94.92%, 92.65%, 83.50%, 73.96%, and 73.96% on DF, FS, F2F, NT, Celeb-DF, and DFDC-p datasets, respectively. However, this model integrates Xception and ViT, which increase the model complexity and training cost.

In 2024, Tipper et al. [24] suggested a study titled "An investigation into the utilisation of CNN with LSTM for video deepfake detection", which provides a CNN-LSTM model. CNN is applied to extract features, and LSTM uses the features to classify the video to detect Deepfake. However, this model has computational complexity. In 2024, Boongasame et al. [25] suggested a study titled "Design and Implement Deepfake Video Detection Using VGG-16 and Long Short-Term Memory", which provides VGG-16, VGG-19, and ResNet 101 models. LSTM was applied for Deepfake detection. Comparatively, the VGG-16-LSTM achieved the best performance, with 97.01% accuracy, 99.07% precision, and 93.04% recall. But this model utilizes multiple DL models, which increases model complexity.

In 2024, Gong et al. [26] suggested a study titled "Swin-fake: A consistency learning transformer-based deepfake video detector", which provides the Swin-fake model to capture the spatial and temporal features. The reliability model is applied to identify the Deepfake with Sample and Computation Redistribution for Face Detection (SCRFD), aligning the face and cropping the face models. This model obtained accuracy of 99.9%, 99.6%, 100%, 98.8%, and 93.7% on the DF, F2F, FS, NT, and DFDC datasets, respectively. In 2024, Liu et al. [27] presented a study entitled "MeST-Former: Motion-enhanced spatiotemporal transformer for generalizable deepfake detection", which used a motion-enhanced spatiotemporal transformer (MeST-Former). This model integrates an RGB and motion encoder to extract attributes, and the Swin transformer is applied to categorize the image and detect deepfakes along with AFCS for facial cropping. The MeST-Former achieved 99.6%, 99.95%, 72.16%, 76.93%, 75.46%, 76.33%, and 76.89% accuracy on DFGC, FF++, DFD, Deeper, CDF1, CDF2, and UADFV datasets, respectively. But, this model consumes higher inference time and it provides higher computational complexity.

In 2024, Al-Adwan et al. [28] presented a study titled "Detection of deepfake media using a hybrid CNN–RNN model and particle swarm optimization (PSO) algorithm", which provides a CNN and a Recurrent Neural Network (RNN), along with PSO for efficient Deepfake detection. This algorithm ensured 97.35% accuracy, 98.36% sensitivity, 96.15% specificity, and 97.56% F1 score on the Celeb-DF dataset, and 94.02% accuracy, 94.53% sensitivity, 93.40% specificity, and 94.53% F1 score on the DFDC dataset. But this model requires a significant amount of memory space for video analysis. In 2025, Xu et al. [29] suggested a study titled "Localization and detection of deepfake videos based on self-blending method", which provides a novel spatial method. This model maintains Swin-UNet for feature extraction and categorizes videos to detect Deepfake. It ensured 99.95%, 98.82%, 99.88%, 99.13%, 87.52%, and 80.34% AUC on DF, F2F, FS, NT, Celeb-DF and DFDC datasets, respectively. Yet, the model has overfitting bias issues and ignores temporal modeling.

In 2025, Sohali et al. [30] suggested a study titled "Deepfake image forensics for privacy protection and authenticity using deep learning", which provides CNN-LSTM, CNN-GRU, and Temporal Convolutional Network (TCN) schemes. It is focused on capturing attributes and uses a GAN model to detect deepfakes. It ensured 0.96 of accuracy, 0.98 of precision, an F1 score of 0.98, and an AUC of 0.93 on the FF++ dataset. But it has a problem with an imbalanced database. In 2025, Zafar et al. [31] introduced a study titled "A hybrid deep learning framework for deepfake detection using temporal and spatial features", which provides a hybrid DL. It includes an enhanced EfficientNetB0, Feature

Pyramid Network (FPN), and TCN to extract features to detect the Deepfake. The model obtained a precision of 0.9268, a recall of 0.9245, and an F1 score of 0.9244 on the FFIW 10K dataset. Yet, it needs high processing time and memory, which increases the computational complexity. In 2025, Agarwal et al. [32] suggested a study, "Eye blinking feature processing using convolutional generative adversarial network for deepfake video detection", which presents a Depthwise Separable Residual Network (DSRes) to capture attributes and a convolutional GAN (Con-GAN) for Deepfake identification. This scheme obtained accuracy of 98.91%, 98.89%, 98.91%, and 98.89% on FF++, DFDC, WildDeepfake, and CelebDFv2 Datasets, respectively.

In 2025, Siddiqui et al. [33] presented a study entitled "Enhanced deepfake detection with DenseNet and Cross-ViT", which suggested a hybrid Vision Transformer-DenseNet framework. The model combined a vision transformer (CrossViT) along with DenseNet121-based convolution to collect global and local pattern recognition. This method ensured an AUC of 99.99%, F1 score of 99% for DeeperForensics, a 97.4% AUC and a 95.1% F1 score for Celeb-DF. But it included ViT and DenseNet, which gives inference time. In 2025, Usmani et al. [34] presented a study titled "Spatio-temporal knowledge distilled video vision transformer (STKD-VViT) for multimodal deepfake detection", which developed a Spatiotemporal Knowledge Distillation-Video Vision Transformer (STKD-VViT) model. The model utilized FakeAVCeleb and obtained an accuracy of 97.49% for the video stream, 98.65% for the audio stream and 96% for the combined stream. However, it lacked performance in quality conditions and generalizability. In 2025, Xiong et al. [35] presented a study titled "BMNet: Enhancing deepfake detection through BiLSTM and multi-head self-attention mechanism", which provided the BiLSTM Multi-Head Self-Attention Network (BMNet) model for deepfake video detection. The BMNet model ensured 95.54%, 92.18%, 80.20%, and 84.72% accuracy and AUCs of 98.60%, 97.21%, 75.72%, and 88.51% on the FF++, UADFV, Celeb-DF, and DFDC datasets, respectively. Yet, this framework handled issues of detecting low-quality and occluded videos.

In 2025, Birla et al. [36] presented a study titled "AVENUE: A novel deepfake detection method based on temporal convolutional network and rPPG information". This work provides a novel Deepfake detection method based on TCN and rPPG information (AVENUE) for image pre-processing, stable clip extraction, and face distortion attention method. This model obtained 98.32%, 90.14%, 93.48%, and 94.24% accuracy and 92.58%, 87.80%, 86.80%, and 84.35% AUC on FF++ (c23), FF++ (c40), CDF, and DFDC datasets, respectively. But this method handled the issues of detecting occluded faces. In 2025, Long et al. [37] presented a study titled "LGDF-Net: Local and global feature-based dual-branch fusion networks for deepfake detection", which provides a Local and Global feature-based Dual-branch fusion network (LGDF-Net). This model contains the Local Compression Model (LCM), Global Expansion Module (GEM), and Multi-Scale Feature Extraction module (MSFE) for feature extraction and the Multi-Scale Feature Fusion module (MSFF) for combining the features to detect deepfakes. This model ensured 99.53%, 100%, 97.64%, and 96.81% accuracy and 1.000, 1.000, 0.997, and 0.996 AUC on DFDC, Celeb-DF, FF++ (HQ), and FF++ (LQ) datasets. It provides computational complexity.

In 2025, Pintelas et al. [38] suggested a study entitled "Quantization-based 3D-CNNs through circular gradual unfreezing for DeepFake detection", which presents a Quantization of Different Augmentation Networks (QDANet). It contains a ResNet3D-18 model to capture features and circular gradual unfreezing (G-CU) to detect the Deepfake. It ensured 0.8641 and 0.99465 accuracy, 0.8368 and 0.9893 sensitivity, 0.8956 and 0.9997 specificity, and 0.9358 and 0.9998 AUC on the cDFDC and DFMnist datasets, respectively. However, the model requires high memory due to quantization. In 2025, Sagar & Arukonda [39] presented a study titled "" proposed that ResNetXt-50 extracts the features and LSTM with frame extraction, face detection and segmentation techniques and achieved 96.67% and 91.80% accuracy for FF++ and Celeb-DF datasets, respectively. However, the model has a slower processing speed and higher computational complexity and cost. Shi et al. presented a Face Reconstruction-Based Generalized Deepfake Detection (FRG2D) model combined with a Residual Outlook Attention (ROA) model. However, the model has increased computational overhead due to multiple attention mechanisms. Manikanta Maguluri et al. proposed a CNN for feature extraction and an LSTM for the detection of deepfakes. This CNN-LSTM achieved 96.4% accuracy, 94.8% precision, 97.2% recall, and 96.0% F1 score on the FF++ dataset. However, the model reduces detection performance due to a lack of feature fusion techniques.

In 2025, Katı et al. [40] presented a study titled "Enhancing deepfake detection through quantum transfer learning and class-attention vision transformer architecture", which provides Quantum Transfer Learning (QTL) with Class-Attention Vision Transformer (CaiT) architectures. This method ensured 90% accuracy, 91% precision, 90% recall, an F1 score and 94% AUC for the DFDC dataset. In 2025, Uddin et al. [41] presented a study titled "Deepfake face detection via multi-level discrete wavelet transform and vision transformer", which provides a Multi-Level Frequency Feature Extraction with Vision Transformer (ML-FFE-ViT) model. It ensured accuracy of 98.43%, 99.92%, and 99.26% for FF++, Celeb-DF and DFID datasets, respectively.

In 2025, Jin et al. [42] presented a study titled "DFD-NAS: General Deepfake detection via efficient neural architecture search", which provides Deepfake Detection via the Neural Architecture Search (DFD-NAS) model. This model ensured accuracy of 94.34%, 78.55% and 79.57% and AUC of 99.75%, 86.41% and 88.70% for Celeb-DF, WildDeepfake and

DFDC, respectively. Yet, the NAS is computationally expensive and relies on GPU-intensive operations, limiting accessibility. In 2025, Kurniawan et al. [43] presented a study titled "Real or deepfake face detection in images and video data using YOLOv11 algorithm", which provides the YOLOv11 model. This model is focused on classifying the facial regions for Deepfake detection and ensured 58.1% precision, 84.9% recall, and 57.8% MAP on the EC2-Deepfake Dataset. However, the model fails to detect faces in stylized and artistic representations.

In 2025, Zheng et al. [44] suggested a study titled "A spatio-frequency cross-fusion model for Deepfake detection and segmentation", which provides a Spatio-frequency cross-fusion (SFCF) model. This model obtained 98.3% AUC and 97.1% F1 score for the FF++ dataset. It provides lower sensitivity and specificity values. In 2025, Pintelas & Livieris [45] developed a study titled "Convolutional neural network framework for deepfake detection: A diffusion-based approach", which provides a diffusion-based convolution network (DiffconvNet). This model obtained an accuracy of 92.9%, 88.5% and 71.2%, and an AUC of 96.2%, 95.3%, and 79.1% for DFDC, DFMNIST+ and EXP datasets, respectively. However, the diffusion process adds computational complexity.

In 2025, Sudarshana, K., & Vamsidhar, Y. [46] presented a study titled "UAM-Net: Robust Deepfake Detection through Hybrid Attention into Scalable Convolutional Network", which provides the Unified Attention Mechanism Network (UAM-Net) model. This model obtained 98.07% accuracy, 97.91% precision, 98.23% recall, 98.07% F1 score and 99.82% AUC for the DFDC-preview dataset. Yet, the model has high computational costs due to attention mechanisms.

In 2025, Mashetty et al. [47] presented a study titled "Deep Fake Detection with Hybrid Activation Function Enabled Adaptive Milvus Optimization-Based Deep Convolutional Neural Network", which provides a hybrid activation function-enabled adaptive Milvus optimization-based Deep Convolutional Neural Network (HA2MilO-DCN) model. It ensured 95.72% accuracy, 94.91% recall and 96.52% precision for the FF++ dataset. Yet, the model has a higher computational overhead. In 2025, Shi et al. [48] presented a study titled "Face reconstruction-based generalized deepfake detection model with residual outlook attention", which provides the Face Reconstruction-Based Generalized Deepfake Detection (FRG2D) model. This model is combined with a Residual Outlook Attention (ROA) scheme. But the model incurs increased computational overhead since it has multiple attention mechanisms.

In 2024, Zakkam et al. [49] developed a study titled "Codeit: contrastive data-efficient transformers for deepfake detection", which provides the Contrastive Data-efficient Transformers (CoDeiT) model. This model obtained 91.5% accuracy for FF++, 86.9% accuracy, 0.95 AUC for DFDC and 78.5% accuracy and 0.89 AUC for CelebDF. Yet, the hierarchical attention mechanism and contrastive learning framework require computational resources. In 2026, Bahadure, S., & Mane, V [50] suggested a study titled "Deepfake detection using deep convolutional neural network and long short-term memory", which provides a strong and consistent DeepFake Detection Network (DFDN). This model is used for improving ability, interpretability, accuracy, and minimalize the computation difficulty of the scheme. It is used to represent the frequency-based, time-based, and phonetic features of the audio. This model has 97.80% accuracy, 98.29% recall, 97.10% precision, and 97.70% F1-score on the FakeAVCeleb dataset.

In 2026, Siagian et al. [51] suggested a study entitled "Improving Vision Transformer for Deepfake Detection," which provides a ViT method for a larger database of Deepfake detection. This model detects more reliably and provides security against threats effectively. It achieves an accuracy of 85.39%, precision of 85.40% and recall of 85.39% for the ImageNet-1k dataset. However, it has issues with error rates. In 2026, Adinarayana et al. [52] presented a study titled "Multi-model Deepfake Detection using Vision Transformers (ViT) and Hybrid Deep Learning Techniques", which provides a Mel-frequency Cepstral Coefficient (MFCC)-based CNN model for audio and video analysis. It ensures accuracy of 95.1% for the FaceForensics++ dataset. In 2026, Raghavajosyula & Pawar [53] presented a study titled "Hybrid deep learning architecture for comprehensive deepfake video facial forgery detection and segmentation", which provides ViT with a CNN for segmenting and detecting facial forgery over the DFDC dataset. This model ensured accuracy of 89.8%, precision of 90.3%, recall of 91.4% and F-measure of 91% for the DFDC dataset. But it has a problem with computational complexity.

In 2026, Tilagul et al. [54] presented a study titled "DeepFake Face Detection Using Machine Learning: A Hybrid CNN-LSTM Approach", which provides CNN and LSTM models. This model is focused on extracting temporal dependencies on the specified video frames effectively via the LSTM model. It enables the extraction of subtle differences in designs revealing deepfake operation. This model ensured 92.4% accuracy for the FF++ (NT) dataset. In 2024, Sardhara et al. [55] suggest a study titled "DeepForgeryNet: a hybrid CNN–LSTM and transfer learning framework for robust image forgery and deepfake detection", which provides the DeepForgeryNet model for deepfake detection. The pre-processing is used to handle the contextual inconsistencies and artifact-level features. This model offers results with 95.12% accuracy, 94.22% recall, 94.65% precision and 94.43% F-score for the forensic dataset. Still, it has issues with optimized features.

Most of the related works did not apply many deep layers, focusing on leveraging simple models integrated with deepfake detection methods, including CNN, LSTM, GRU and ViT. The existing models have issues with noise, accuracy and computational complexity for the given deepfake video datasets. Therefore, the suggested deepfake

framework solves this issue by using transformer learning and DL models. Complete datasets and features of surveyed DL papers are illustrated in Table I, providing comparative analysis for deepfake detection using DL models to display the varied approaches and findings in the field. The suggested Deepfake detection framework focuses on these limitations and suggests strategies to overcome their drawbacks.

TABLE I. COMPARATIVE ANALYSES FOR DEEPFAKE DETECTION USING DL MODELS

References	Methods	Dataset	Accuracy	Limitations
Xia et al. (2021)	MesoNet	FF++ and Celeb-DF	91.6% and 89.9%	Couldn't deal with complex patterns, causing lower accuracy
Deng et al. (2022)	EfficientNet-V2	FF++	97.90%	Difficulties in detecting occluded face regions
Kuang et al. (2022)	EfficientNet-BiLSTM	FF++ and Celeb-DF	98.21% and 98.93%	High computational complexity
RAZA et al. (2022)	DFP	Benchmark Kaggle	94%	Higher training time and computational complexity.
Chen et al. (2022)	Convolutional LSTM	FF++, CelebDF, and DFDC	99%, 99.93%, and 92.43%	Limited performance on complex datasets
Pang et al. (2023)	MRE-Net, MIE and LIE	FF++, Celeb-DF, DFDC, and Wild Deepfake	97.76%	Longer training time
Yu et al. (2023)	MSVT	FF++, DFD, DFDC, Celeb-DF-v2, DF 1.0 Raw, and Wild Deepfake	99.73%	High memory cost
Zaho et al. (2023)	ISTVT	FF++, Celeb-DF, DFDC, FSh, and DFo	91.9%	Problem with redundant information
Lu et al. (2023)	iCapsNet-TSF	Celeb-16	94.07%	But, this model has a problem with inference time
Masud et al. (2023)	CNN-LSTM	Celeb-DF	99.24%	High computational cost
Qadir et al. (2024)	ResNet-Swish-BiLSTM	FF++(FS), FF++(F2F), FF++(NT), and DFDC	98.9%	Low precision values
Javed et al. (2024)	Hybrid DL	FF++, v1, and Celeb-DF v2	98.73%	Slow convergence speed
Alhaji et al. (2024)	ACO-PSO	DFDC	98.30%	High complexity
Omar et al. (2024)	CNN and CoAtNet	DF, F2F, FS, NT, and Celeb-DF	97.70%	Risk of overfitting
Ramadhani et al. (2024)	ViViT	Celeb-DF V2, DFDC, Deepfake TIMIT, and FF++	88.03%	Required additional parameters and memory space
Gong et al. (2024)	Swin-fake transformer	DF, F2F, FS, NT, and DFDC	93.7%	Pre-processing results are average
Liu et al. (2024)	MeST-Former	DFGC, FF++, DFD, Deeper and UADFV	76.89%	Higher inference time
Al-Adwan et al. (2024)	CNN-RNN, PSO	DFDC	94.02%	Needed additional memory space
Xu et al. (2025)	Swin-UNet	DF, F2F, FS, NT, Celeb-DF and DFDC	96.74%	Overfitting problem
Sohali et al. (2025)	CNN-LSTM, CNN-GRU, and TCN	FF++	96%	Class imbalance
Zafar et al. (2025)	FPN	FFIW 10K	91%	High processing time and memory
Usmani et al. (2025)	STKD-VViT	FakeAVCeleb	97.49%	Poor performance with quality conditions and generalizability
Xiong et al. (2025)	BMNet model	Celeb-DF and DFDC	80.20%	Low-quality and occluded videos
Birla et al. (2025)	AVENUE	CDF and DFDC	93.48%	Issues in detecting occluded faces
Long et al. (2025)	LGDF-Net	FF++ (LQ)	96.81%	Computational complexity

Pintelas et al. (2025)	QDANet	cDFDC and DFMnist	86.41%	High memory due to quantization
Katı et al. (2025)	QTL with CAiT	DFDC	90%	Error rate and scalability issues
Uddin et al. (2025)	ML-FFE-ViT	DFID	99.26%	High information loss
Jin et al. (2025)	DFD-NAS	Wild Deepfake	78.55%	Limiting accessibility
Kurniawan et al. (2025)	YOLO11	EC2-Deepfake	84.67%	Poor artistic representations
Zheng et al. (2025)	SFCF	FF++	96%	Lower sensitivity and specificity values
Sudarshana & Vamsidhar (2025)	UAM-Net	DFDC-preview	98.07%	High computational costs due to attention mechanisms
Shi et al. (2025)	FRG2D	Celeb-DF dataset	86.9%	Multiple attention mechanisms
Bahadureb & Mane (2026)	DFDN	FakeAVCeleb dataset	97.80%	High security threats
Siagian et al. (2026)	ViT	ImageNet-1k dataset	85.39%	High error rates
Adinarayana et al. (2026)	MFCC-based CNN	FaceForensics++ and DFDC	95.1%	Scalability issue
Raghavajosyula & Pawar (2026)	ViT with CNN	DFDC dataset	89.8%	High computational complexity
Tilagul et al. (2026)	CNN and LSTM	FF++(NT) dataset	92.4%	High error rates
Sardhara et al. (2026)	Hybrid CNN–LSTM and transfer learning	Forensic dataset	95.12%	Limited optimized features
Proposed HLAAWMSAN–STHGOCVT	Localized multi-scale attention, Hypergraph Convolutional Network, Convolutional Vision Transformer, and STO	DFD benchmark		Multi-stage architecture increases implementation and computational requirements

3. METHODOLOGY

Figure 1 shows the block diagram for the proposed Deepfake detection framework. The ACTCN model used Cascaded CNN (C-CNN) and RNN-based Temporal Normalization for facial region and alignment detection. The C-CNN model applies multiple convolutional layers to identify face landmarks such as eyes, nose, mouth, and jawline. The RNN-based Temporal Normalization uses an RNN network to track facial movements across multiple frames and ensures stability in face detection. This method also detects inconsistencies in head poses, eye blinks, and lip movements, which are unnatural in Deepfake. The Face Alignment and Normalization standardize face positions to a canonical form for downstream models to detect Deepfake-related distortions. The super-resolution methods are used to expose the hidden synthetic artifacts and pixel-level inconsistencies. The Multi-scale ResNet-based Feature Fusion employed multi-scale Feature Extraction (MSFE) and Feature Fusion using Residual Learning (FF-RL) methods. The MSFE applies ResNet-based convolutional layers to extract features at multiple resolutions to capture fine-grained details that are lost in low-resolution images. The FF-RL technique combined the multiple feature maps using skip connections in ResNet to preserve important facial structures and to prevent the loss of texture details during up scaling Deepfake images. The CNN-based AutoEncoder technique converts RGB images to luminance and chrominance channels (YCbCr). The CNN-based AutoEncoder learns optimal transformations to highlight color-based Deepfake artifacts and YCbCr representation detects inconsistencies in skin tone, facial lighting, and shadow alignment.

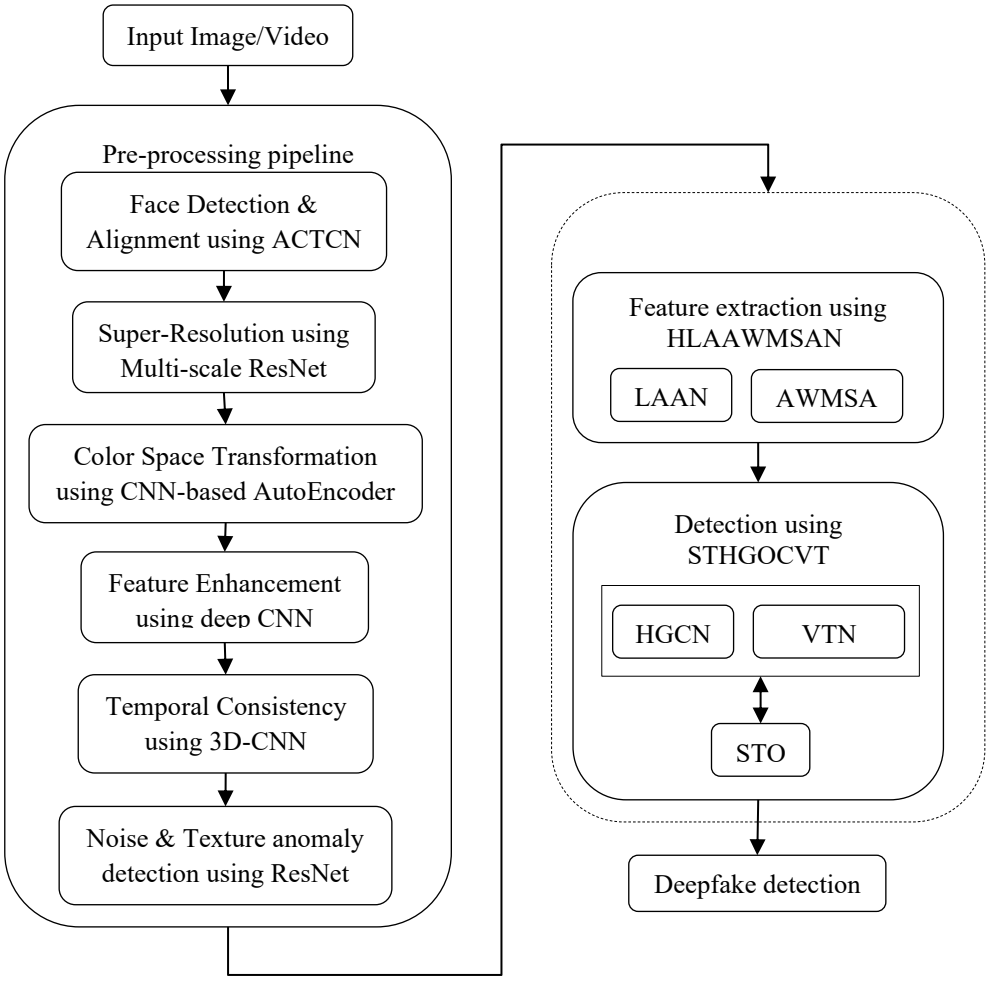


Fig. 1. Block Diagram for the Deepfake detection model

The features are enhanced by using CNN-based Laplacian Layers and the Fast Fourier Transform (FFT) technique. The Laplacian CNN Layers enhance the spatial features and detect edges, contours, unnatural textures and highlights blurred boundaries, distortions, and unnatural skin smoothing in Deepfake images. For frequency domain analysis the CNN applies FFT to extract high-frequency components that reveal Deepfake noise patterns and to identify anomalies in spectral data. The 3D-CNN model extracts spatiotemporal features from video sequences detects sudden jumps in facial misalignment between head and body movements and identifies lip- synchronization mismatches. The ResNet model extracts deep features based on noise and texture inconsistencies such as blurry patches, unrealistic skin smoothness and pixilation effects. The HLAAWMSAN-STHGOCVT framework gathers Deepfake artifacts and spatial and temporal inconsistencies. The model hyperparameters are tuned using STO, which improves convergence speed and efficiency.

3.1 Deep Learning based Preprocessing Pipeline Model for Deepfake video Detection

This pre-processing pipeline helps to expose subtle artifacts developed by the Deepfake generation technique for better classification. This process increases feature extraction for efficient identification of manipulated content. This pipeline leverages CNN with ResNet as the backbone and contains multiple pre-processing steps.

The proposed pre-processing pipeline is created to counter the adaptive and adversarial deepfake generation techniques by systematically exposing manipulation. Face localization and alignment using ACTCN restrict analysis to manipulated facial regions, preventing attackers from exploiting background and identity bias. Multi-scale resolution enhancement strengthens pixel-level artifacts that are smoothed by generative models to evade detection. Color space transformation to YCbCr isolates chrominance inconsistencies that are less perceptible to the human eye but frequently disrupted during synthesis. Frequency-domain enhancement using Laplacian and FFT operations reveals unnatural noise patterns and texture irregularities introduced by generative processes. Temporal analysis through 3D-CNN enforces frame-to-frame consistency checks, addressing temporal incoherence commonly exploited by video-based attacks. This pipeline transforms raw video inputs into artifact representations, shifting the detection task from appearance-based recognition to

forensic analysis and increases robustness against evolving and deepfake attacks. Table II summarize the purpose, configuration, input, output, and connection of each preprocessing component.

TABLE II COMPONENTS AND IMPLEMENTATION SETTINGS OF THE PREPROCESSING PIPELINE

Preprocessing Component	Main Function	Specific Settings	Input	Output
Video decoding & frame sampling	Extract representative frames from the input video	Number of sampled frames, sampling interval, supported video formats	Raw video	Sampled frame sequence
Adaptive Cascaded Temporal Convolutional Networks (ACTCN)	Face detection, alignment, and cropping	Face detector, landmark localization, alignment method	Sampled frames	Aligned facial regions
Multi-scale ResNet-based Feature Fusion	Super-resolution and artifact enhancement	ResNet backbone, feature fusion strategy, upscaling factor	Aligned face images	High-resolution face images
CNN-based Autoencoder	RGB-to-YCbCr color-space transformation	Encoder–decoder architecture, processed Y channel	High-resolution face images	Color-transformed images
Spatial & Frequency Feature Enhancement	Enhance edges, textures, and frequency artifacts	Laplacian filtering, Fast Fourier Transform (FFT), feature fusion	Color-transformed images	Enhanced spatial-frequency representation
3D Convolutional Neural Network (3D-CNN)	Temporal consistency analysis	Temporal window size, frame resolution, 3D convolution kernels	Consecutive video frames	Temporal feature representation
ResNet	Noise and texture anomaly detection	ResNet architecture, normalization, feature extraction layer	Temporally processed frames	Noise and texture feature maps

3.1.1 ACTCN for face detection and alignments

The manipulated facial regions in the video and images are detected and aligned using the Adaptive Cascaded Temporal Convolutional Networks (ACTCN). The ACTCN model integrated cascading CNN for facial region detection, temporal convolutional network (TCN) for sequence processing, adaptive attention mechanism for feature refinement and face alignment module using Geometric transformation. The Cascaded CNN extracts multi-scale spatial features from face regions. This ACTCN process detects the facial landmarks using CNN, the input face image (xf), which learns the mapping function (MF). This process is given by $MF: xf \rightarrow d(xf)$, here $d(xf)$ refers to the 2D landmark coordinates for the face [58]. The CNN model comprises three convolutional layers and three fully connected layers that are expressed as

$$MF^{(l)} = \rho\left(W^{(l)} * MF^{(l-1)} + b^{(l)}\right) \quad (1)$$

Here, $W^{(l)}\ and\ b^{(l)}$ refers to the weight and bias of the l^{th} convolutional layer (CL), $*$ represents the convolutional operations and $\rho(xf)$ represents the activation function. The response $\left(y_{i,j}\right)$ at the location (i,j) in the CL is expressed as

$$y_{i,j} = \max\left(\sum_{x=0}^{s-1}\sum_{y=0}^{s-1}\sum_{z=0}^{k-1} im_{i+x,j+y,z}W_{x,y,z} + b, 0\right) \quad (2)$$

Here, im refers to the input feature map, W, b refers to the weight and bias of the convolutional kernel and s, k represents the depth and kernel size. After initializing landmarks, each facial point is refined through a sequence of local CNN and the network errors are corrected using an adaptive sampling strategy based on Gaussian distribution [68]. The error distribution of a landmark position is modeled as

$$ed = p - p_{gt} \sim \eta(\mu, \sigma^2) \quad (3)$$

Here, pl refers to the predicted landmark position, pl_{gt} refers to the ground truth landmark position and $\eta(\mu, \sigma^2)$ refers to the Gaussian distribution of error. The refine prediction $(p_{new})$is estimated as $p_{new} = p + \Delta p$, here $\Delta p = \eta(\mu, \sigma^2)$, the values of μ, σ^2 are updated dynamically based on the error statistics of previous cascades. The temporal convolutional

network (TCN) ensures temporal stability by processing the facial feature map in sequence. The sequence of the feature maps $fm_{t-n}, \dots, fm_t, \dots, fm_{t+n}$ applies dilated causal convolutions and it is given as

$$tf_t = \sum_{i=0}^{k-1} W_T(xf) * fm_{t-dr.i} \quad (4)$$

Here, tf_t refers to the temporal feature at the time t, $W_t(xf)$ refers to the 1D temporal convolutional weight and dr represents the dilation rate. The ACTCN increases spatial and refines the facial landmarks and also captures temporal inconsistencies.

3.1.2 Multi-scale ResNet-based feature fusion approach

After the ACTCN method the detected face region resolution is enhanced by using a super-resolution (SR) technique of multi-scale ResNet-based feature fusion technique. The SR is employed to upscale images, preserve details and retain the Deepfake artifacts that are hidden in low-resolution images. This process gathers hierarchical features using multiple ResNet-based convolution layers, fuses the features at different resolutions and reconstructs the high-resolution face region using the SR network. Here, I_{LR} represents the low-resolution image and, I_{HR} represents the high-resolution output, this process is expressed as, $I_{HR} = \mathbb{F}(I_{LR}, W_{SR})$, here $\mathbb{F}$ refers to the SR function, lp_{SR} refers to the learnable parameters of the model [59]. The Adaptively Weighted Multi-Scale Attention (AW-MSA) mechanism dynamically assigns different contributions to multi-scale feature representations according to their discriminative importance. It further combines local and global channel and spatial attention to emphasize manipulation-related regions and suppress less informative features [59].. The different receptive fields from the ResNet model extract the multi-scale features. The multi-scale feature extraction using the convolution filters is defined as

$$fm_l = \rho(W_l * I_{LR} + b_l) \quad (5)$$

Here, fm_l refers to the feature map at l^{th} layer, W_l, b_l represents the weight and bias of the convolutional filters and, $\rho()$ refers to the non-linear activation function. This multi-scale CL captures fine information and global structure to improve SR quality [59]. The skip connections are used to prevent losing important details and it is expressed as

$$f_{residual} = I_{LR} + \sum_{i=1}^{N} \beta_i fm_l \quad (6)$$

Here, $f_{residual}$ represents the residual feature map, β_i represents the trainable fusion weights for each scale and Nf represents the number of feature extraction layers. The final SR output (I_{HR}) is reconstructed using a transposed convolution layer is expressed as

$$I_{HR} = W_{up} * f_{residual} + b_{up} \quad (7)$$

Here, W_{up}, b_{up} represents the upsampling weight and bias. The entire SR model with Multi-scale ResNet feature extraction, residual feature fusion, and Transposed convolution for upsampling is expressed as

$$I_{HR} = W_{up}(I_{LR} + \sum_{i=1}^{N} \beta_i \rho(W_l * I_{LR} + b_l)) + b_{up} \quad (8)$$

The SR technique enhances the image by retaining the hidden Deepfake artifacts such as blurring and edge distortions.

3.1.3 CNN-based AutoEncoder For color space transformation

After super-resolution, the colour space transformation converts RGB images into YCbCr color space using a CNN-based AutoEncoder to highlight Deepfake inconsistencies in skin tone, facial texture and colour blending. The YCbCr perceptually uniform color space that divides luminance (Y) refers to the brightness information and Chrominance (Cb, Cr) refers to the colour differences [60]. The CNN-based AutoEncoder learns optimal transformations by using encoder and decoder techniques. The encoder compresses RGB images into latent representations and the decoder reconstructs the YCbCr image from the latent space. The input RGB image (I_{RGB}) and the encoder function $\big(E(.)\big)$ is defined as

$$Z = E(I_{RGB}; W_E) \quad (9)$$

Here, Z represents the latent representation in the encoder space and W_E represents the encoder weights learned through CNN layers. The CL in the encoder detects edges, textures and colour variations and compresses input detail [60]. After passing through multiple CNN layers, the output is reshaped into YCbCr representation is given as

$$I_{YCbCr}^{pred} = \emptyset(Z) \tag{10}$$

Here, $\emptyset(Z)$ represents the learnable function that reconstructs YCbCr components. The decoder reconstructs the YCbCr output and remains perceptually consistent with the original input that is expressed as

$$I_{YCbCr}^{out} = D(Z; W_D) \tag{11}$$

Here, $D()$ refers to the decoder network and W_D represents the learnable decoder weights. The decoder applies deconvolution (upsampling) layers that are expressed as

$$fm_l' = \rho(W_l' * fm_{l-1}' + b_l') \tag{12}$$

Here, W_l', b_l' represents the weights and biases for the inverse transformation from latent space to YCbCr. The soft constraint is employed to match the YCbCr output with the expected colour distributions.

3.1.4 Spatial and Frequency Domain Feature Enhancement Using a CNN-Based Model

This method combines spatial domain analysis (pixel-level features) that identifies unnatural edges, contours and textures and frequency domain analysis (texture-based features) that exposes the compression artifacts and hidden noise patterns for feature enhancement. This method comprises a Laplacian filter in CNN for spatial feature enhancement, a fast Fourier transform (FFT) for frequency feature extraction and CNN a based fusion model for combined feature learning. This method divides into two parallel branches, namely spatial feature enhancement (SFE) using Laplacian filters in CNN and frequency feature extraction (FFE) using FFT and CNN. The Deepfake artifacts create blurred edges, distorted textures and unrealistic contours [61]. So, a Laplacian filter helps to highlight the inconsistencies by computing the second derivative of pixel intensity changes. The Laplacian operator is expressed as

$$\mathfrak{L}I(xf, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{13}$$

The CNN model employs multiple Laplacian-enhanced CL to gather spatial Deepfake inconsistencies and it is expressed as

$$F_{sp} = \rho(W_{CNN} * (\mathfrak{L}I_{YCbCr}) + b_{CNN}) \tag{14}$$

Here, b_{CNN}, W_{CNN} represents the bias and the weight of the CNN model, $\rho(xf) = \max(0, xf)$ represents the ReLU activation function. This process emphasizes fake-looking edges, texture inconsistencies, and boundary distortions due to face-swapping Deepfake [61]. The hidden high-frequency noisy patterns and compression errors are detected by employing FFT in the image. The image $I(xf, y)$ and the Fourier transform $I(\mathfrak{f}_g, \mathfrak{f}_h)$ is evaluated as

$$I(\mathfrak{f}_g, \mathfrak{f}_h) = \sum_{g=0}^{M-1} \sum_{h=0}^{N-1} I(xf, y) e^{-iu2\pi\left(\frac{\mathfrak{f}_g g}{M} + \frac{\mathfrak{f}_h h}{N}\right)} \tag{15}$$

Here, $\mathfrak{f}_g, \mathfrak{f}_h$ refers to the frequency components with g and h directions, M, N represents the Number of rows and columns and iu refers to the imaginary unit. After applying FFT, CNN processes the frequency domain representation and it is expressed as

$$F_{frequency} = \rho(W_{frqCnn} * (I_{FFT}) + b_{frqCnn}) \tag{16}$$

Here, b_{frqCnn}, W_{frqCnn} represents the bias and weight for frequency learning and I_{FFT} represents the Fourier-transformed image. The combined spatial and frequency features create a unified feature representation and it is given as

$$F_{final} = \alpha F_{sp} + \beta F_{frequency} \quad (17)$$

Here, α, β represents the learnable weight balancing spatial and feature features. The output gathers the pixel-level distortions like edges, contours and textures and frequency-level anomalies like high-frequency noise and GAN artifacts. The CNN-based enhance model loss function is given by

$$\mathbb{L} = \left\| \lambda_1 \mathbb{L}_{sp} + \lambda_2 \mathbb{L}_{frq} \right\|^2 \quad (18)$$

Here, $\mathbb{L}_{sp}$ denotes the spatial loss and $\mathbb{L}_{frq}$ denotes the frequency loss. The spatial loss is also known as the edge consistency loss and the frequency loss occurred on frequency reconstruction loss, which is formulated as

$$\mathbb{L}_{sp} = \sum \| \mathbb{L} I_{enh} - \mathbb{L} I_{GT} \|^2 \quad (19)$$

$$\mathbb{L}_{frq} = \sum_i \| \mathfrak{f}(I_{enh}) - \mathfrak{f}(I_{GT}) \|^2 \quad (20)$$

Here, I_{enh} refers to the enhanced image, $\mathfrak{f}$ denotes the frequency, and I_{GT} denotes the ground truth image. The overall equation for the spatial and frequency feature enhancement is formulated as

$$\mathfrak{f}_{final} = \alpha \rho (W_{CNN} * (\Omega I_{YCbCr}) + b_{CNN}) + \beta \, \rho \left(W_{frqCnn} * (I_{FFT}) + b_{frqCnn} \right) \quad (21)$$

This process enhances the frequency and spatial features and exposes the hidden Deepfake artifacts.

3.1.5 Temporal consistency for frame-to-frame analysis using a 3D-CNN

Temporal consistency is employed to identify the inconsistencies in the multiple video frames, which include unnatural head movements, sudden jumps in face alignment, mismatched facial expressions within frames and lip-synchronization errors in videos [٣٨]. The Deepfake models generate frame-frame synthetic videos that lack temporal coherence, so the 3D-CNN model is applied to capture the spatiotemporal relationships with multiple frames by applying convolutions in both temporal and spatial dimensions. The 3D convolution operation is expressed as

$$fm_{t,h,\acute{\omega}i,c} = \rho \left(\sum_{i=0} \sum_{j=0} \sum_{k=0} W_{i,j,k,c} . I_{t+i,h+j,\acute{\omega}i+k,c} + b_c \right) \quad (22)$$

Here, $I_{t,h,\acute{\omega}i,c}$ denotes the input video tensor with the dimensions including time, height, width and channels, $W_{i,j,k,c}$ denotes the trainable weights of the 3D convolution kernel, k_T, k_H, k_{wl} denotes temporal, height, width kernel sizes and $Fr_{t,h,\acute{\omega},c}$ denotes the output feature map at the frame [61]. The input video segment is represented as the 3D tensor and it is denoted as $V \in \mathbb{R}^{T \times H \times W \times C}$. The multiple 3D CL extracts spatiotemporal features from the input video and it is expressed as

$$I^{(l)} = \rho \left(W^{(l)} * I^{(l-1)} + b^{(l)} \right) \quad (23)$$

Here, $I^{(l-1)}$ refers to the input to the, l^{th} layer and $W^{(l)}, b^{(l)}$ represents the 3D convolution weights and biases. The temporal max-pooling is employed to reduce computational cost and it is expressed as

$$f_{pool}^{(l)} = \max_{t \in [t_1, t_n]} f^{(l)}(t) \quad (24)$$

Here, $f^{(l)}(t)$ represents the feature at time t and the maximum activation is selected by a temporal window. The flattened spatiotemporal features are passed through the FCL and it is given as $sofmax\left(W_{fc} * f_{pool} + b_{fc}\right)$, here, W_{fc}, b_{fc} represents the learnable parameters and the softmax function results in a probability score. The 3D-CNN model is trained using classification loss, temporal consistency loss and adversarial loss and it is expressed as

$$\mathbb{L}_{tot} = \lambda_1 \mathbb{L}_{clas} + \lambda_1 \mathbb{L}_{tem} + \lambda_1 \mathbb{L}_{adv} \quad (25)$$

$$\mathbb{L}_{clas} = -\sum_c y_c \log(\hat{y}_c) \tag{26}$$

$$\mathbb{L}_{tem} = \sum_{t=1}^{T-1} \|fm_t - fm_{t-1}\|^2 \tag{27}$$

$$\mathbb{L}_{adv} = -\mathbb{E}\left[logD\left(f_{pool}\right)\right] \tag{28}$$

Here, y_c refers to the ground truth class label, $\hat{y}_c$ refers to the predicted probability, fm_t refers to the extracted feature and, fm_{t-1} refers to the feature in the next frame. The temporal consistency model using 3D-CNN is expressed as

$$y = sotmax\left(W_{fc}.\max_t \rho(W_{3D} * V + b_{3D}) + b_{fc}\right) \tag{29}$$

During temporal feature extraction, consecutive video frames are organized into overlapping clips of eight frames. Each frame is resized to 64×64pixels, converted from Blue-Green-Red (BGR) to Red-Green-Blue (RGB), and normalized to the range $[0, 1]$. The resulting clip is rearranged from $(T, H, W, C)$to (C, T, H, W), and a batch dimension is added to form an input tensor of size $(B, 3, 8, 64, 64)$. This tensor is processed through three successive three-dimensional convolutional blocks with $3 \times 3 \times 3$kernels and 32, 64, and 128 output channels, respectively. Each convolutional block is followed by a Rectified Linear Unit (ReLU) activation and three-dimensional max pooling to progressively capture short-term spatial and temporal variations while reducing dimensionality. A sliding-window strategy with a stride of one frame is employed by removing the oldest frame after each prediction and adding the next incoming frame. The final spatiotemporal feature map is flattened and projected into a 512-dimensional temporal representation before the binary classification layer. These temporal features encode frame-to-frame variations in facial alignment, head movement, expression, texture, and visual mouth dynamics that may reveal temporal inconsistencies introduced by deepfake generation. This process detects frame-to-frame inconsistencies and detects unnatural head movements for improving Deepfake detection.

3.1.6 ResNet for Noise and Texture anomaly detection

After temporal consistency analysis, the noise and texture anomaly is detected using the ResNet model. This process identifies unnatural noise patterns and synthetic artifacts. The ResNet model captures fine-grained texture irregularities and detects unnatural noise patterns [62]. The input image is represented as, $I \in \mathbb{R}^{H \times W \times C}$. The ResNet extracts the deep hierarchical features from *I* using the CL that is given as

$$I^{(l)} = \rho\left(W^{(l)} * I^{(l-1)} + b^{(l)}\right) \tag{30}$$

The residual block allows the model to retain fine-grained texture details and detects unnatural smoothing and it is expressed as

$$fm_{res}^{(l)} = fm^{(l)} + W_{res} * fm^{(l-1)} \tag{31}$$

Here, W_{res} represents the residual connection weight matrix and $fm_{res}^{(l)}$ represents the enhanced feature map with texture information [62]. The Laplacian operator is employed to increase noise detection and to capture high-frequency noise patterns and it is given as

$$\mathfrak{L}I(xf, y) = \left|\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}\right| \tag{32}$$

The Deepfake textures are globally smooth but locally inconsistent; this inconsistency is detected by using multi-scale ResNet and it is multi-scale ResNet is given by

$$T_{multi} = \sum_{s=1}^{S} \alpha_s * W_s * fm^{(l)} \tag{33}$$

Here, NS refers to the number of scales, W_s represents the convolutional weights for different receptive fields, and α_s represents adaptive weights for different scales. The extracted noise and texture features are classified using an FCL. The final equation for noise and texture detection using ResNet is given as

$$y = softmax\left(W_{fc}.\sum_{s=1}^{S}\alpha_s * W_s * \left(W_{res} * fm^{(l)} + N_{res}\right) + b_{fc}\right) \quad (34)$$

This multi-scale, noise-aware ResNet approach enhances Deepfake detection performance.

3.1.7 Complexity and Scalability Analysis for pre-processing

For the face detection and alignment, the time complexity is given as $\left(O(T.nP.k^2, s)\right)$, here np refers to the number of pixels in the input image, k indicates the kernel size, T number of video frames, s indicates the depth, sl indicates the number of super-resolution scales and nL indicates the number of layers. In super-resolution, each scale processes the full image using convolutions and the fusion adds constant overhead. The super-resolution complexity is given as $\left(O(nP.k^2.sl.s)\right)$. For color space transformation the complexity is given as $\left(O(nP.k^2.s)\right)$. For feature enhancement the complexity is given as $\left(O(nP.k^2.s + n\log n)\right)$, here, $n\log n$ indicates the number of samples involved in the enhancement transformation. For temporal consistency the complexity is described as $\left(O(T.nP.k_t^3.s)\right)$, here, k_t^3 refers to the size of 3D kernel at time t. For noise and texture detection, the complexity is described as $\left(O\left((nP.sl.k^2.s)\right)\right)$. The total time complexity $\left(TC_{pp}\right)$ for the pre-processing pipeline is formulated as

$$TC_{pp} = O(T.nP.(k^2.s + k_t^3.di) + sl.nP.s + n\log n) \quad (35)$$

The space complexity evaluates memory usage from weights and activations. For the face detection and alignment, the space complexity is given as $\left(O(T.nP.k^2.s)\right)$. In super-resolution, multiple intermediate feature maps from each scale and filter weights are stored. The super-resolution complexity is given as $\left(O(sl.nP + sl.k^2.s)\right)$. For color space transformation the complexity is given as $\left(O(nP + k^2.s)\right)$. For feature enhancement the complexity is given as $\left(O(nP.s)\right)$. For temporal consistency the complexity is described as $\left(O(T.nP.s)\right)$. For noise and texture detection, the complexity is described as $\left(O\left((nP.sl.s)\right)\right)$. The total space complexity $\left(SC_{pp}\right)$ is formulated as

$$SC_{pp} = O(nP.T.s + sl.nP.k^2.di) \quad (36)$$

The dataset provides volume and diversity for building robust and scalable Deepfake detection models. The pre-processing pipeline is efficient on high-resolution video, manages longer clips without degrading detection accuracy and processes multiple video in parallel on GPU. The GPU-accelerated super-resolution allows real-time improvement for video streams. The pre-processing pipeline works with batch processing frames and it is lightweight, which is used for edge devices and mobile applications. This processing pipeline enables robust performance at scale and suitable for large scale deployment in real-world Deepfake detection systems.

3.2 The Proposed Deepfake Detection

After pre-processing pipeline, the HLAAWMSAN-STHGOCVT model is developed for extracting features and detection of Deepfake. The HLAAWMSAN model captures the features related to Deepfake and images using multi-scale attention mechanism and the STHGOCVT model identify Deepfake events by analyses using Hypergraph Convolutional Networks (HGCN) and Vision Transformers (ViT). The HLAAWMSAN model is structured into two main components including Attention mechanism Network (LAA-Net) and Adaptively Weighted Multi-Scale Feature Extraction (AW-MSA). The HLAAWMSAN model improves feature extraction by applying LAA that focuses on manipulated regions, multi-scale feature learning captures both fine and coarse Deepfake artifacts and adaptive weighting enhances important features, which surpasses irrelevant details. The LAA-Net focuses on the manipulated regions using explicit attention mechanisms. This process also uses heatmaps-based and self-consistency attention techniques to identify vulnerable pixels [63]. The LAA network includes an enhanced feature pyramid network for gathering multi-scale features without redundancy.

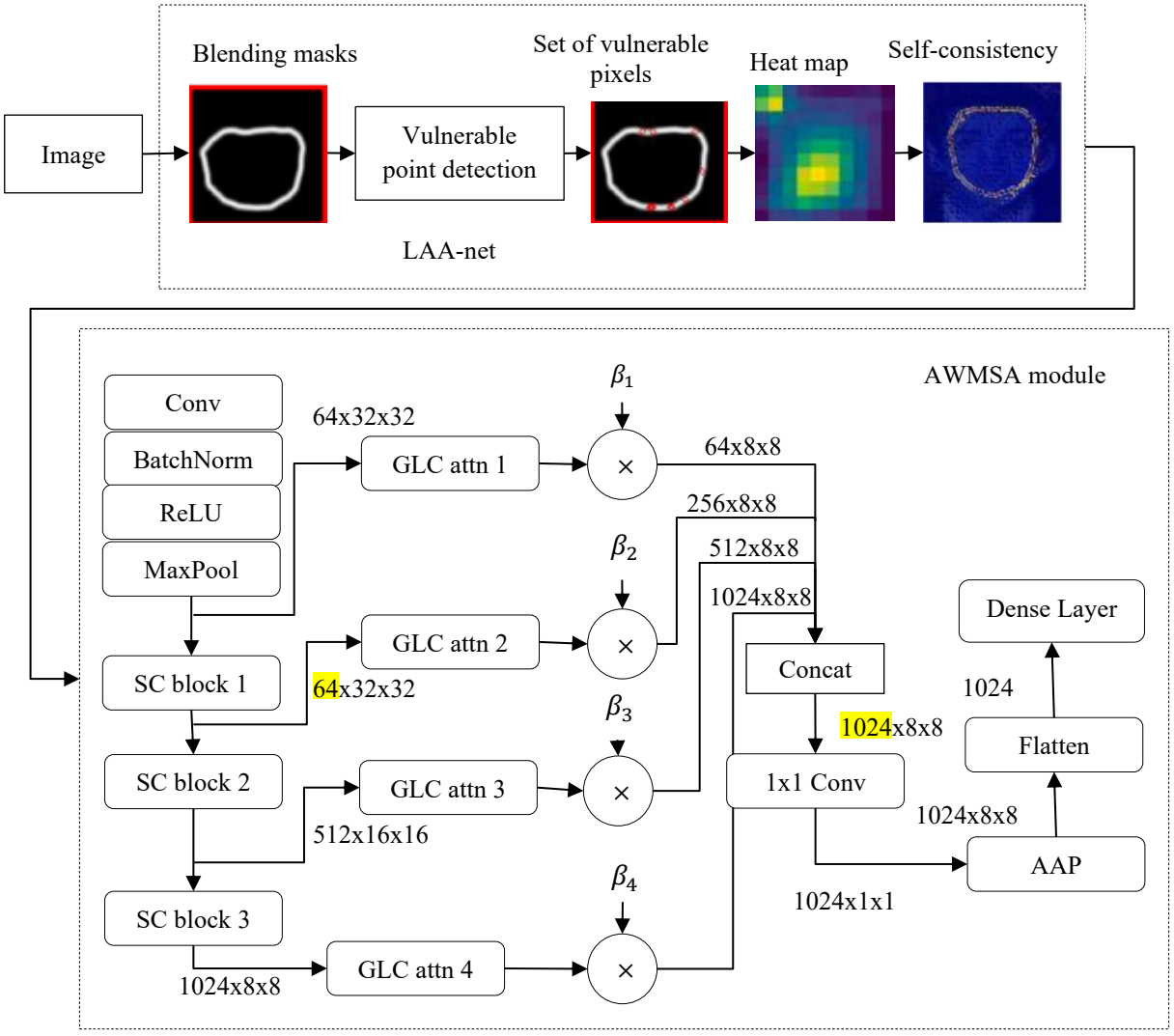


Fig. 2. HLAAWMSAN model for feature extraction

Figure 2 illustrates the HLAAWMSAN model for feature extraction. The blending masks are created for the pre-processed image. The manipulated regions are highlighted using a blending mask and it is identified using vulnerable point detection and generated using set vulnerable pixels. The heat map is developed to present the intensity and distribution of manipulated regions. The self-consistency module is performed to refine the features of manipulated areas. The adaptively weighted multi-scale attention (AWMSA) module includes convolution (Conv), batch normalization (BatchNorm), ReLU and Max pooling (MaxPool) operations with the skip connection (SC) blocks. The gathered features are passed through the global local attention module (GLC attn. 1, 2, 3 and 4) at various stages to improve Deepfake artifacts. The relevant features are emphasized using adaptive weights and the dimension sizes are increased for hierarchical learning. The processed features are fed into concatenation (concat), convolution (1x1 Conv), adaptive average pooling (AAP), flattening and dense layer for generating the deep feature vector.

The Deepfake image is created by blending a foreground (I_F) with a background (I_B) using a blending method and it is given as

$$I_M = H_M \odot I_F + (1 - H_M) \odot I_B \tag{37}$$

Here, I_B represents the background, I_M represents manipulated face image and ch_M refers to the deformed convex hull mask values between [0,1]. The blending artifacts concentrate on the boundary near the hull mask (H_M) and a blending boundary mask $(B_M)_{i,j\in[1,s]}$ is evaluated as

$$B_M = 4.H_M \odot (1 - H_M) \tag{38}$$

The vulnerable pixels $(vP_k)$are the pixel values that are impacted by blending and it is given as

$$vP_k = \underset{(B_M)_{i,j \in \|1,s\|^2}}{\operatorname{argmax}} (B_M) \tag{39}$$

The heat map branch captures vulnerable pixels and their surrounding regions affected by Deepfake artifacts. The heatmap method considers the neighborhood information to increase robustness against noise and illumination variations. This process is implemented using un-normalized Gaussian distribution to generate heatmap, adaptive standard deviation to model local inconsistencies dynamically and focal loss to optimize the heatmap prediction [63]. The heatmap (ht_m) is generated by replacing a Gaussian mask (G_{mk}) centered at each vulnerable pixel(vP_k). The standard deviation (SD_k) of the Gaussian distribution is adaptively computed based on the blending boundary width and height and it is evaluated as

$$G_{mk} = \left(g_{ij}^k\right)_{i,j \in [1,s]},\ g_{ij}^k = e^{-\frac{i^2+j^2}{2SD_k^2}} \tag{40}$$

Here, SD_k represents the adaptive standard deviation, i, j represents pixel coordinates relative to the center, g_{ij}^k represents heatmap intensity at position (i, j) and G_{mk} represents the Gaussian mask cantered at a vulnerable pixel(vP_k). To dynamically adjust SD_k based on the blending boundary, a radius is evaluated using intersection over union (IOU) with overlapping masks. The standard deviation is set as $SD_k = \frac{1}{3} vr_k$, here, vr_k is computed from the virtual object size that overlaps the blending mask. The final heatmap is obtained by superimposing all Gaussian masks (G_{mk}) is computed as

$$ht_m = \sum_{k=1}^{card(vP_k)} G_{mk} \tag{41}$$

Here, $card(vP_k)$ refers to the Cardinality of the vulnerable pixels detected in the Deepfake image. The heat map branch is optimized using focal loss, which helps handle the class imbalance between manipulated and real regions and is computed as

$$\mathbb{L}_{ht_m} = \sum_{i,j}^{D} -\left(1 - \tilde{h}_{ij}\right)^{\gamma} \log \tilde{h}_{ij} \tag{42}$$

Here, $\tilde{h}_{ij}$ represents the predicted heatmap value, γ represents the focusing parameter to adjust loss weight. The predicted heatmap value $\left(\tilde{h}_{ij}\right)$ is evaluated as

$$\tilde{h}_{ij} = \begin{cases} \tilde{h}_{ij}, & if\ h_{ij} = 1 \\ 1 - \tilde{h}_{ij}, & else \end{cases} \tag{43}$$

Here, h_{ij} refers to the ground-truth heatmap at that pixel. The self-consistency attention ensures that pixels near the vulnerable regions are internally consistent and it is evaluated as

$$SC = 1 - \left|vs_{ij}.1 - B_M\right| \tag{44}$$

Here, vs_{ij} refers to the value at the selected vulnerable point. The binary cross-entropy loss is applied to the train and it is computed as

$$\mathbb{L}_{SC} = -\sum\left(SC \log \widehat{SC} + (1 - SC) \log\left(1 - \widehat{SC}\right)\right) \tag{45}$$

The AWMSA network refines Deepfake detection by ensuring that multi-scale features are adaptively weighted based on their discriminative importance. The multi-scale features are extracted as

$$F_{multi} = \sum_{s=1}^{S_a} \alpha_s.W_s * fm \tag{46}$$

Here, fs_a refers to the number of features scales, W_s represents the convolutional weight at different receptive fields and α_s represents the adaptive learned weights. The AWMSA uses the global-local channel spatial attention (GLCS) to dynamically refine feature maps and the local channel attention (LCA) increases individual feature channels by evaluating an attention map using global average pooling (GAP) and 1×1 convolution is computed as

$$A_c^l = \rho\left(conv_{1*1}\left(GAP(fm)\right)\right) \tag{47}$$

Here, $\rho(.)$ refers is the sigmoid activation function, $GAP(fm)$ represents the GAP feature map across spatial dimensions, and $conv_{1*1}$ refers to the point-wise convolutional that refines the pooled features. The global channel attention (GCA) increases feature maps by learning feature importance globally using a softmax function, which is given as

$$A_c^g = fm \times softmax\ \rho\left(conv_{1*1}\left(GAP(fm)\right)\right) \times \rho\left(conv_{1*1,2}\left(GAP(fm)\right)\right) \tag{48}$$

$$F_c^g = fm \times A_c^g \tag{49}$$

Here, $conv_{1*1}, conv_{1*1,2}$ represents the two separate convolutions generated at different attention scores. The local spatial attention (LSA) refines feature maps at multiple scales using multi-kernel convolutions and the multi-scale features are computed as

$$fm' = conv_{1*1}(fm) \tag{50}$$

Global spatial attention (GSA) learns long-range spatial relationships using non-local filtering and it is computed as

$$A_s^g = conv_{1*1}\left(conv_{1*1}(fm) \times softmax\left(conv_{1*1}(fm) \times conv_{1*1}(fm)\right)\right) \tag{51}$$

This process uses non-local filtering to capture large-scale spatial dependencies and the softmax normalization prevents vanishing gradient issues. The global spatial attention features are computed as

$$F_s^g = F_c^g + \left(F_c^g \times A_s^g\right) \tag{52}$$

After extracting the local and global attention features, these features are weighted and fused to form the final feature representation and it is computed as

$$F_{GL} = (w \times fm) + (w_l \times F_s^l) + \left(w_g \times F_s^g\right) \tag{53}$$

Here, w, w_l, w_g refers to the learnable weights. The multi-scale attention mechanism effectively enhances Deepfake detection by focusing on local channel importance using LCA, capturing global feature dependencies using GCA, extracting spatial relationships at diverse scales using LSA, learning long-range spatial features using GSA and fusing local and global attention maps for optimal feature representation. The proposed hybrid framework employs feature-level fusion to integrate the representations extracted by the HLAAWMSAN feature extractor before classification. Multi-scale localized artifact features generated by HLAAWMSAN are concatenated into a unified feature representation and subsequently refined through a 1×1convolution, followed by a dense layer and adaptive average pooling. The fused representation is then provided as the input to the STHGOCVT module, enabling the classifier to jointly exploit localized artifact information together with global spatial-temporal relationships for video-level deepfake detection.

After extracting the features, the features are used by STHGOCVT for Deepfake detection. The STHGOCVT model integrates Hypergraph convolutional networks (HGCN), vision transformer network (ViT) and Siberian tiger optimization (STO). The HGCN model handles the complex and non-linear spatial relationships by gathering spatial dependencies, ViT captures the long-range temporal dependencies and STO optimizes hyperparameters for efficient training and inference. A Hypergraph models higher-order relationships and connects multiple nodes and it is given as $G = (V, hE)$, here V refers to the nodes and hE refers to the hyperedges [23]. Transformer-based architectures have also been employed to jointly model spatial and temporal inconsistencies in deepfake videos. For example, Kingra et al.

[61] proposed SFormer, which combines a Swin Transformer for spatial feature extraction with a sequential Transformer block for modeling temporal relationships across consecutive video frames.

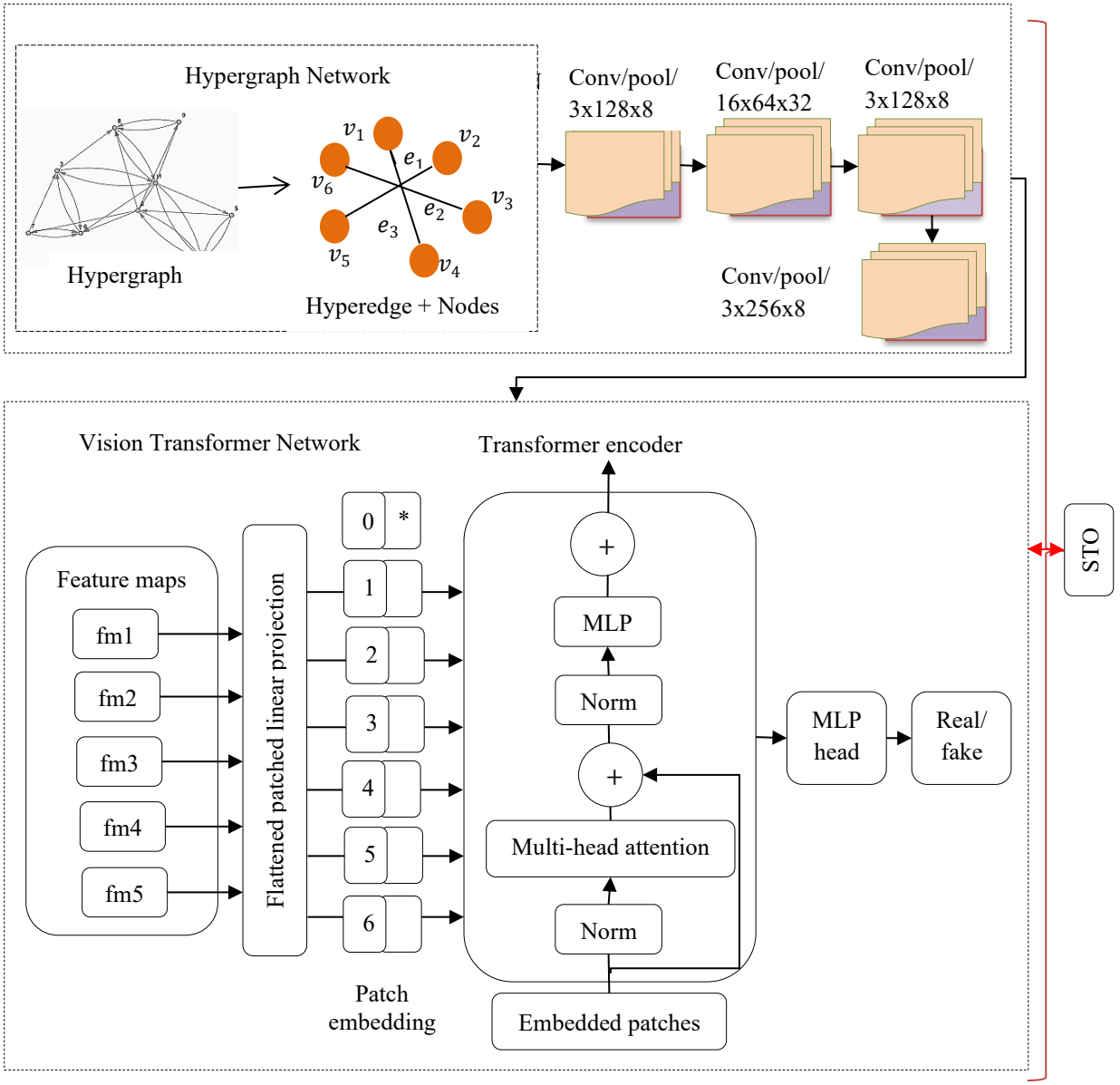


Fig. 3. Architecture of the STHGOCVT model

Figure 3 represents the Architecture of the STHGOCVT model. The Hypergraph extracts the feature points and the spatial relationships with multiple feature points. The Hypergraph relationships are processed using the convolutional layers for refining and transforming the features. The feature maps are divided into patches in the patch embedding module and the patches are processed in the transformer encoder. This transformer encoder consists of MLP and a normalization (norm) layer that refines the embedded patch and the multi-head attention increases the important features by reducing noise. The STHGOCVT model is optimized by using the STO algorithm.

A Hypergraph representation is created, where nodes (V) represents feature points gathered from HLAAWMSAN and the hyperedges (hE) gathers spatial relationships between multiple feature points and it is given as

$$H^{(l+1)} = \rho\left(Dm_v^{-1} f H^{(l)} W_g\right) \tag{54}$$

Here, $fH^{(l)}$ denotes the feature representation at the layer(l), W_g denotes the trainable weight matrix and Dm_v^{-1} denotes the degree matrix of the Hypergraph. The convolutional operation on Hypergraph with Laplacian is defined as

$$\mathfrak{L}I_H = I - Dm_v^{-\frac{1}{2}} in_m W Dm_e^{-\frac{1}{2}} in_m{}^T Dm_v^{-\frac{1}{2}} \tag{55}$$

Here, $\mathfrak{L}I_H$ represents the Hypergraph Laplacian, in_m represents the incidence matrix, and Dm_e represents the hyperedge degree matrix. The forward propagation in HGCN is expressed as

$$xm' = \rho\left(Dm_v^{-\frac{1}{2}} i_m W Dm_e^{-1} i_m{}^T Dm_v^{-\frac{1}{2}} xm\Theta\right) \tag{56}$$

Here, fxm refers to the input feature matrix from frames, and Θ denotes the trainable weight matrix. The Hypergraph allows a single edge to connect to multiple nodes for detecting multi-region inconsistencies in Deepfake. After extracting the high-order spatial dependencies the convolutional ViT is used to process frame sequences and detect Deepfake inconsistencies. This process gathers both the local features and global attention modeling that identifies the unnatural facial expressions [23]. The ViT contains patch embedding, convolution feature extraction (CFE), Multi-Head Self-Attention (MSA), and Feed-Forward Network (FFN). The patch embedding module for splitting input frames into small patches for processing. Each video frame (vFr) is split into small non-overlapping patches (Pt) and it is given as

$$X_t = W_{patch} * Fr_t + b_{patch} \tag{57}$$

Here, X_t denotes the patch embedding representation, W_{patch} represents the learnable weight embedding matrix for patches, Fr_t refers to the frame at the time step, b_{patch} denotes the bias term for embedding. The CFE uses CNN layers to preserve spatial relationships and extracts local spatial features from patches and it is formulated as

$$Fr' = conv_{3\times3}(X_t) \tag{58}$$

$$F'' = batchnorm(F') \tag{59}$$

Here, this process ensures spatial consistency before feeding into transformers. The MSA computes relationships between patches and helps to analyze the irregularities within the frames. The self-attention score is computed using the query-key-value (QKV) mechanism and it is expressed as

$$AS = softmax\left(\frac{Q_m Ke_m{}^T}{\sqrt{d_k}}\right) \tag{60}$$

Here, $Q_m = W_Q X_t$ denotes the query matrix, $Ke_m = W_K X_t$ denotes the key matrix, $Vl_m = W_V X_t$ denotes the value matrix, and di_k denotes the dimensions of key vectors. To capture long-range dependencies in video frames the weight attention output is computed as $z'_w = A.Vl_m$. The FFN refines the features before classification. The transformer encoder refines attention-based features using multi-layer perceptron (MLP) is given as

$$z''_w = ReLU(fcl_1 z'_w + b_1) \tag{61}$$

$$z_m = fcl_2 z''_w + b_2 \tag{62}$$

Here, fcl_1, fcl_2 denotes the fully connected layers, and b_1, b_2 denotes the bias term. After obtaining spatial features from HGCN and temporal features from ViT are fused adaptively with learnable weights. The fused representation is passed through a softmax classifier and it is given as

$$y = softmax\left(W_{fc} * F_{fusd} + b_{fc}\right) \tag{63}$$

Here, W_{fc} denotes the trainable weight matrix of the fully connected layer, F_{fusd} denotes the fused features and b_{fc} denotes the bias vector of the fully connected layer. The STHGOCVT model is trained using a hybrid loss function and it is given as

$$\mathbb{L}_{tot} = \lambda_1 \mathbb{L}_{clas} + \lambda_2 \mathbb{L}_{temp} + \lambda_3 \mathbb{L}_{reg} \quad (64)$$

The classification loss, Temporal Consistency Loss and Regularization Loss (L2 Penalty) are formulated as

$$\mathbb{L}_{clas} = -\sum_c y_c \log(\hat{y}_c) \quad (65)$$

$$\mathbb{L}_{tem} = \sum_{t=1}^{T-1} \|fm_t - fm_{t-1}\|^2 \quad (66)$$

$$\mathbb{L}_{reg} = \lambda \|W\|^2 \quad (67)$$

This process improves the Deepfake by gathering fine-grained local features using HGCN, modeling long-range dependencies using self-attention and detecting temporal inconsistencies with multiple frames. The STHGOCVT model hyperparameter is tuned by using the Siberian Tiger Optimization (STO) to further increase the performance of the model for detecting Deepfake detection. The STO algorithm imitates the optimization strategies (refer as ST) that explore and exploit search space for optimization problems [56]. The ST applies a technique by integrating individual and group hunting techniques. The STO applies three phases including the scouting phase, the chasing phase and the attack phase. In the initialization phase, the tigers are placed randomly in the search space and each ST represents a solution in the dimensional space and is given as

$$st_{u,v} = lb_v + r_{uv} * (ub_v - lb_v), u = 1,2, \dots t_n, j = 1,2, \dots p_h \quad (68)$$

Here, op_h denotes the dimensionality problem values, lb_v denotes the lower bound of the v^{th} variable, r_{uv} denotes the random number between the ranges $[0,1]$ and ub_v denotes the upper bound of the v^{th} variable. After initialization, the candidate solution is evaluated using the objective function $\left(f(st_u)\right)$ and the best solution is determined as

$$st_{best} = \underset{st_u}{\operatorname{argmin}} f(st_u) \quad (69)$$

The first phase of the STO algorithm includes exploration phase behavior that involves selecting the prey (optimum candidate solution) and the chasing process [56]. The selection phase makes sudden extensive changes for exploration (global search). Each ST considers multiple potential preys and the set of prey positions for the u^{th} ST is given as

$$PP_u = \{st_o | o \in \{1,2, \dots t_n\} \wedge F_o < F_u\} \cup \{st_{best}\} \quad (70)$$

Here, st_{best} denotes the current optimum solution, PP_u denotes the set of better candidate solutions (potential prey), st_o denotes the position of o^{th} ST, and F_o denotes the objective function of st_o. The value of PP_u is selected randomly and each tiger moves towards the prey position and the updated position for the u^{th} ST is evaluated as

$$st_{u,v}^{p1s1} = st_{u,v} + r_{u,v}.\left(TP_{u,v} - rn_{u,v} * st_{u,v}\right) \quad (71)$$

Here, $TP_{u,v}$ denotes the randomly selected member from the set PP_u, $st_{u,v}^{p1s1}$ denotes the updated position of the u^{th} ST in the search space and $r_{u,v}$ denotes the random number between the ranges $\{1,2\}$. If the newly calculated position is increased than the objective function, the value can be used for position updates for STO members, represented as

$$st_u = \begin{cases} st_u^{p1s1}, F_u^{p1s1} < F_u \\ st_u, \ else \end{cases} \quad (72)$$

Here, F_u^{p1s1} denotes the objective function of the u^{th} ST member. The second stage uses the chasing process to update the position fine-tune the movement and retain only the optimum positions [56]. The new position of the u^{th} ST is evaluated as

$$st_{u,v}^{p1s2} = st_{u,v} + \frac{r_{uv} * (ub_v - lb_v)}{ti}, ti = 1,2, \dots Ti \quad (73)$$

$$st_u = \begin{cases} st_u^{p1s2}, F_u^{p1s2} < F_u \\ st_u, \ else \end{cases} \quad (74)$$

Here, $st_{u,v}^{p1s2}$ denotes the updated position of the u^{th} ST after refining its attacks, $st_{u,v}$ denotes the current position before adjustment, st_u^{p1s2} denotes the new refined position based on local search and ti denotes the iteration counter.

The second phase of the STO algorithm includes ST fighting with a bear including the attacking stage and conflict stage. In the attack stage, the tiger makes a sudden movement randomly and selects the bear (better solution). In the conflict stage, the tiger refines its position through a local search near the fight area. In the selection phase of the bear, the other STO members are considered as the set of possible bears. The random bear selected from the population for attack is denoted as (st_k) that leads to positional changes and improves exploration. The new position of the u^{th} ST is updated as

$$st_{u,v}^{p2s1} = st_{u,v} + r_{u,v}.\left(st_{k,v} - st_{u,v}\right) \quad (75)$$

Here, $st_{u,v}^{p2s1}$ denotes the updated position of the u^{th} ST in the attack phase, $st_{u,v}$ denotes the current position before the attack, and $st_{k,v}$ denotes the position of the selected bear (better candidate solution). The newly calculated position value is increased than the current solution, and the value is replaced by using,

$$st_{u,v} = \begin{cases} st_u^{p2s1}, F_u^{p2s1} < F_u \\ st_u, \quad\quad else \end{cases} \quad (76)$$

In the second stage, the positions are updated based on the conflicts phase, which causes small changes to refine the position and it is given as

$$st_{u,v}^{p2s2} = st_{u,v} + \frac{r_{uv}}{ti} * (ub_v - lb_v) \quad (77)$$

Here, $st_{u,v}^{p2s2}$ denotes the new position of the u^{th} ST after refining the position using conflict. This new position is accepted if the solution is improved and it is formulated as

$$st_{u,v} = \begin{cases} st_u^{p2s2}, F_u^{p2s2} < F_u \\ st_u, \quad\quad else \end{cases} \quad (78)$$

Here, st_u^{p2s2} denotes the new refined position based on the exploitative method. The STO algorithm balances the exploration and exploitation and tunes the hyperparameters of STHGOCVT for Deepfake detection.

Algorithm 1: Simplified STO for hyperparameter tuning

- Initialize the candidate population uniformly within the predefined lower and upper bounds.
- Evaluate the fitness of all candidates and identify the global-best solution.
- For each optimization iteration:
  a. For each candidate solution:
    i. Generate a random value $r \in [0, 1]$.
    ii. If $r < 0.5$, select a random population member and generate an exploratory position by moving the current candidate toward it.
    iii. Otherwise, generate an exploitative position by moving the current candidate toward the global-best solution.
    iv. Clip the generated position to the predefined search bounds.
    v. Evaluate the new candidate and accept it only if it produces a lower fitness value.

vi. Update the global-best solution when a better candidate is obtained.
b. Record the best fitness value for convergence analysis.

- Return the global-best solution, its fitness value, and the convergence history.

The hybrid STHGOCVT model with the STO algorithm increases the model's capability to identify the fine-grained Deepfake artifacts and long-range dependencies that improve the detection performance with reduced computational cost. Hyperparameter tuning was formulated as an outer-loop minimization process using a simplified Siberian Tiger Optimization (STO)-based procedure, while Adaptive Moment Estimation (Adam) was used as the inner-loop optimizer to update the trainable network weights. Each tiger represented a seven-dimensional candidate vector comprising the learning rate, dropout rate, three stage-specific embedding dimensions, batch size, and number of training epochs. Candidate solutions were initialized uniformly within predefined lower and upper bounds. For each candidate, a new STHGOCVT model was trained on the training subset and evaluated on the validation subset. The negative validation accuracy was used as the fitness function, J(h)=−Accuracy val (h), because the implemented STO procedure minimizes the objective value. During each iteration, a candidate performed either exploration toward a randomly selected population member or exploitation toward the current global-best solution, each with a probability of 0.5. The updated solution was clipped to the permitted bounds and accepted only when it improved the fitness value. After completing the optimization process, the best candidate configuration was selected for final model training, while the test data were excluded from hyperparameter selection. Table III illustrates the optimized hyperparameter value.

TABLE III. OPTIMIZED HYPERPARAMETERS

Hyperparameter	Parameters	Optimized value
Learning Rate	1e-5 – 1e-3	1e-4
Dropout Rate	0.3 – 0.6	0.5
Hidden Size	256 – 768	512
Kernel Size	{2, 3, 5}	2, 5
Batch Size	2 – 8	4
Number of Epochs	5 – 30	10

The STO optimized learning rate, dropout ratio, hidden size, kernel size, and segment length. The STO algorithm is employed to optimize hyperparameters that influence convergence stability, generalization, and attention effectiveness.
The proposed framework performs video-level classification rather than frame-level classification. Each input video is represented by a sequence of preprocessed facial frames, from which spatial and temporal information is extracted throughout the proposed preprocessing pipeline and the STHGOCVT model. The extracted representations are then integrated to generate a single prediction for the entire video, classifying it as either Real or Fake. Consequently, all reported evaluation metrics, including Accuracy, Precision, Recall, F1-score, Matthews Correlation Coefficient (MCC), and Area Under the Curve (AUC), are computed at the video level.
As the proposed framework is a hybrid one, the reason is that convolutional neural networks and Vision Transformers give complementary feature representations. CNNs perform extremely well in identifying small-scale local artifacts such as texture mismatches, blending boundaries, and edge warps that occur during the deepfake creation process. They have a receptive field though that restricts the study of long-range spatial and temporal dependencies. On the other hand, Vision Transformers excel in capturing global contextual relationships by leveraging self-attention mechanisms and may be less sensitive to subtle manipulation artifacts in the local context. As a result, the proposed HLAAWMSAN–STHGOCVT approach utilizes localized multi-scale feature extraction and transformer-based global representation learning to boost the robustness and accuracy of the video-level deepfake detection.

3.2.1 Complexity and Scalability Analysis

The HLAAWMSAN-STHGOCVT model is developed to detect Deepfakes by combining spatial attention mechanisms, multi-scale learning, hypergraph convolution, transformers, and optimization. The complexity analysis is evaluated based on computational cost (time), memory usage (space), and scalability with increasing input size and model depth.
The framework distributes computation across lightweight, task-specific modules that operate sequentially on refined data representations. Early-stage pre-processing reduces input redundancy by focusing computation on aligned facial regions and artifact-relevant features, which lowers the effective input size for subsequent stages. The computationally intensive components such as attention and spatio-temporal modeling operate on compact, information representations. The runtime scales approximately linearly with the number of frames and facial regions processed, which makes the

framework scalable to longer videos and higher resolutions. This design enables the model to achieve strong detection performance without a proportional increase in inference cost, supporting its suitability for real-time and large-scale deepfake detection scenarios.

The time complexity analysis for feature extraction includes gathering the features from multiple receptive fields and multi-layer extraction is outlined as $\left(O(nP.k^2.s + vP_k.\sigma)\right)$ and $\left(O(sl.nP.k^2.s + nP.c)\right)$ here, vP_k refers to the number of vulnerable pixels, c indicates number of channels and σ refers to the Gaussian per pixel. The feature extraction complexity is given as $\left(O(nP.k^2.s + vP_k.\sigma + sl.nP.k^2.s + nP.c)\right)$. The total detection complexity is outlined as $\left(O(hE.s^2 + Nh.s^2 + ah.pt^2.di + pt.s^2 + \delta)\right)$, here, hE refers to the number of Hypergraph, δ refers to the optimization algorithm, di refers to the dimension, Nh refers to the number of nodes in Hypergraph, ah be the attention heads and pt be the number of patches. The total time complexity is given as

$$TC_M = O(T.nP.s + sl.nP + ah.pt^2.s + hE.di^2 + \delta) \tag{79}$$

The space complexity for collecting feature maps at multiple scales are outlined as $\left(O(nP.s + vP_k)\right)$ and $\left(O(sl.nP.c)\right)$. The space complexity for extraction is outlined as $\left(O(nP.di + vP_k + sl.nP.s)\right)$. The space complexity for detection is outlined as $\left(O(Nh.s + Nh.hE + pt^2.ah + \delta)\right)$. The total space complexity is given as

$$SC_M = O(T.nP.s + sl.nP.c + ah.pt^2 + Nh.di + Nh.hE + \delta) \tag{80}$$

The proposed model learns generalizable artifact patterns in the manipulated data. From the diverse video resolution, the features are extracted, which maintained performance with varying input quality. The model also handled video based temporal inconsistencies and enabled robust temporal modeling within video frames. The methodology supports batch processing, parallel execution and hardware acceleration in high data volume. This process retains accuracy and interpretability without increasing compute cost.

The STHGOCVT framework incorporates multiple DL components and each module serves a specific and non-redundant function within a modular pipeline, including face alignment, artifact enhancement, temporal consistency and noise detection. These components are lightweight and shallow and the pre-processing pipeline simplifies the input representation and decreases the computational burden of the STHGOCVT detection model. The complexity and scalability analyses demonstrate that the framework maintains feasible time and space complexity, which enables real-time and large-scale deployment. The proposed modular framework increases interpretability, robustness and efficiency. Table IV present the limitations and proposed solution using the HLAAWMSAN-STHGOCVT framework.

TABLE IV. PROPOSED SOLUTIONS IN IMPLEMENTING HLAAWMSAN-STHGOCVT

Challenges in related works	**Limitations**	**Solutions in HLAAWMSAN-STHGOCVT**
Shallow learning	The models have limited capacity of fails to capture subtle manipulations.	The model uses LAA-Net and AWMSA to extract localized, fine-grained, and multi-scale features from vulnerable regions.
Sensitive to noise	The models lead to high inference time and memory usage.	Pre-processing pipeline with FFT, Laplacian, YCbCr conversion, and ResNet ensures colour, texture, and spectral artefact enhancement.
Poor Generalizability	Models struggle with unseen manipulation techniques and occlusions.	HGCN model higher-order feature dependencies, capturing complex inconsistencies.
Lack of Robust Temporal Analysis	The models lack to fully capture frame-to-frame inconsistencies.	The model used 3D-CNN and ViT with MSA to model both short-term and long-range frame dependencies.
Over-reliance on Global Features	The models ignore localized inconsistencies often in Deepfakes.	Heatmap-guided attention (LAA) identifies manipulated pixel regions with Gaussian-based vulnerable point detection
Scalability Issues	Inference becomes challenging with high-dimensional video input	Modular pipeline supports parallel GPU processing, batch frame handling, and is lightweight for edge devices.
Computational complexity	Complex structures lead to slow convergence and high resource use.	STO optimization for hyperparameter tuning and pre-processed light-weight representations for efficiency
Parameter sensitive	The model performance depends on the parameters that affect the performance.	The model parameters are optimized dynamically to increase the performance.

4. RESULTS AND DISCUSSION

The HLAAWMSAN-STHGOCVT model for Deepfake detection used a dataset from Kaggle, namely the Deep Fake Detection (DFD). The implementations used Python using the PyCharm development environment on a system with an Intel Core i5 processor, Windows 11 OS with 8GB RAM, and 512GB SSD. Accuracy, Precision, Recall, F1 score, MCC, Training time (TT) and processing time (PT) are used for the analysis. Several practical constraints were encountered during experimentation. The available hardware limited the batch size and increased the training and preprocessing time. The online-video evaluation was restricted to downloadable YouTube videos with resolutions between 720p and 1024p, while compression, poor illumination, facial occlusion, and resizing to 224×224pixels may reduce the visibility of subtle manipulation artifacts. In addition, the multi-stage preprocessing pipeline and STO-based hyperparameter search increased the computational and storage requirements. Since the evaluation used one benchmark dataset, a limited number of YouTube videos, and an 80:20 split, broader cross-dataset and continuous live-stream testing remains necessary in future work.

4.1 Dataset Description

The deep fake detection dataset DFDC is curated for Deepfake detection tasks that provide a collection of videos sequences for training and computing DL models that identify manipulated media. The dataset is sourced from FaceForensics server that provides a high-quality dataset focused on face manipulation detection [57]. This dataset bridges the gap by delivering a substantial volume of video content for both original and manipulated. This dataset contains 3068 manipulated videos to supports research and development approach in video-based media forensics and manipulation detection.

Real-time videos are collected from YouTube and the video with high quality and available formats are downloaded. For the downloaded videos, the quality of the video such as resolution and video properties from the video stream is checked. The constraint for the quality of the video is given as 720p – 1024p for proper Deepfake detection. The downloaded videos are processed using the pre-processing pipeline. After processing the video, the manipulation techniques such as lip-syncing, face swapping, gender change, hair colour, skin texture, expression swap and eye colour are applied. The face frames are extracted from the video and the frames are extracted in 224x224 pixels are evaluation. Table V depicts the dataset summary.

TABLE V. DATASET SUMMARY

Attribute	**Description**
Source	FaceForensics server
Total Manipulated Videos	3068
Original Video Source	YouTube
Video Quality Constraint	720p – 1024p
Video Format	High-quality downloadable formats
Manipulation	Face swapping, lip-syncing, gender change, hair colour alteration, skin texture modification
Frame Resolution	224 x 224 pixels

Table VI represents the experimental setting used for STHGOCVT for Deepfake detection model.

TABLE VI. EXPERIMENTAL SETTINGS

	Parameters
Number of Layers	9
Learning Rate	1e-4
Drop Out	0.5
Epoch	10
Segment length	10
Hidden size	512
Kernel size	2,5
Batch size	4
Input Channels	3, 32, 64
Output Channels	32, 64, 128
Padding	2
Stride	1
Activation Function	ReLU

Figure 4 illustrates the sample images for Deepfake dataset. This figure displays a three-image sequence that visually represents the stages in a Deepfake detection workflow. In the original image frame, the individual's facial expression is

neutral, and the image maintains authentic lighting, skin tone, and natural contours. The manipulated image frame represents the Deepfake-altered version of the original image. In this image the facial expression has been slightly modified to resemble a subtle smile. The facial image textures show slight blending artifacts around the jaw and cheeks that indicates face-swapping and morphing. The proposed model identified the manipulation present in the image.

Figure 5 presents the real-time images for original, manipulated and fake videos. This figure visually represents the performance of the STHGOCVT model at various stages of Deepfake detection. In the original image the facial attributes are natural, with no signs of tampering. The image is identical to the original; no manipulation was applied and this helps evaluate false positive detection, ensuring the model correctly classify real data. In the manipulation images, the input image frame is from a clean, real video a person speaking in a neutral tone. The lips and mouth region have been digitally altered to match a different audio input and this manipulation caused slight asynchrony and unnatural mouth movements. In the last image, the face has been replaced and overlaid with another individual's face, using face-swapping techniques. In the manipulated image the lighting mismatches and blending artifacts appear around the edges. The model detects the frame as a Deepfake, which demonstrates the system's capability to detect identity-level features manipulations.

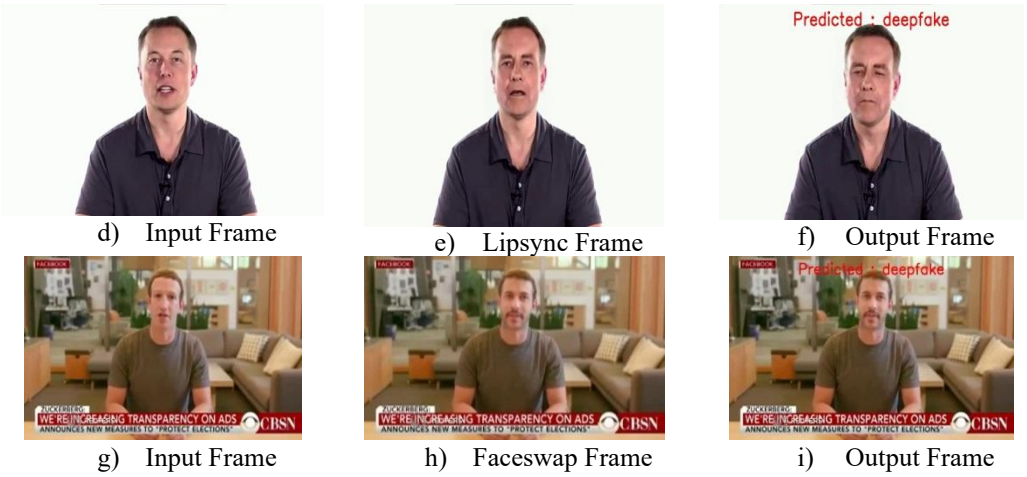
a) Original Frame b) manipulated Frame c) Output Frame

d) Original Frame e) Faceswap Frame f) Output Frame

g) Original Frame h) No change i) Output Frame

Fig. 4. Sample images for Deepfake dataset

a) Original Frame b) No change Frame c) Output Frame



d) Input Frame e) Lipsync Frame f) Output Frame

g) Input Frame h) Faceswap Frame i) Output Frame

Fig. 5. Real-time images frames for original, manipulated and fake video

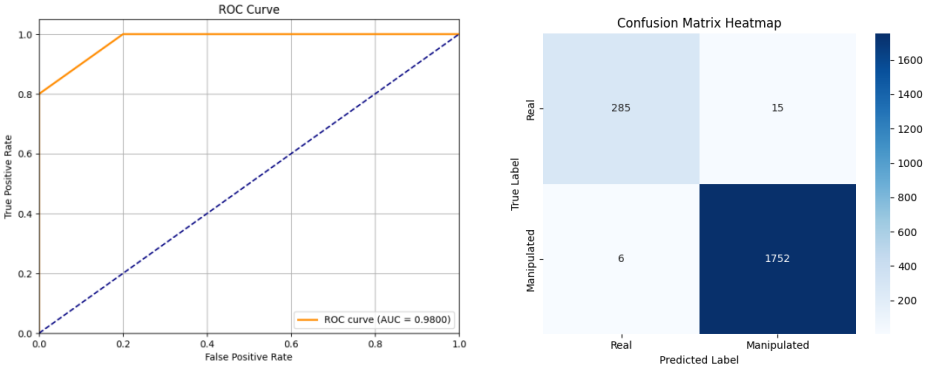


Fig. 6. ROC curves and consfusion matrix

Figure 6 provides quantitative validation of the HLAAWMSAN-STHGOCVT model classification capability. The ROC Curve demonstrates the model's ability to differentiate between real and fake frames. The ROC and confusion matrix affirm the model's balanced sensitivity and specificity. The models high values in the true classes and low misclassification rates validate the design of the model's architecture in self-consistency learning and attention refinement.

The reported Accuracy and AUC values are obtained using a fixed train–test split. This evaluation strategy was chosen to reflect realistic deployment scenarios in which a trained deepfake detection model is applied to unseen videos. The dataset is divided into training and testing sets using an 80:20 ratio, ensuring that subjects and videos in the test set are not observed during training. To reduce variability and improve result stability, model training is repeated multiple times with different random initializations, and the reported metrics correspond to the averaged performance on the same held-out test set. This protocol ensures a fair and reproducible assessment of generalization.

TABLE VII. CLASSIFICATION MODEL COMPARISON

Model	Accuracy	Precision	Recall	AUC	MCC	RMSE	MAE	TT (s)	PT (s)
HLAAWMSAN	93.85	94.20	93.45	95.74	96.48	0.3106	0.1487	135.36	485.32
STHGOCVT	95.19	93.67	94.54	96.51	95.98	0.2995	0.1342	146.11	469.54
HLAAWMSAN-STHGOCVT	97.66	95.13	96.51	97.58	97.11	0.3249	0.0921	150.54	447.69
HLAAWMSAN-STHGOCVT-STO	99.15	98.36	100	98.21	97.54	0.2486	0.0657	155.87	411.74

Table V represents the classification model comparison. The model achieved accuracy of 99.15%, outperforming other models in range of 1.49% to 5.3%, with lower processing time. The training time (TT) is increased due to the deeper layers and hybrid fusion, this increment is marginal and proportional to the performance gain. The processing time (PT) is decreased due to the STO algorithm that tuned the computational efficiency by decreasing non-informative feature

weights and by choosing optimal parameters. This process increases the deployment with faster inference time and ensures long-term lower computation.

TABLE VIII. PRE-PROCESSING PIPELINE COMPARISON

Model	Accuracy	Precision	Recall	F1 score	AUC	MCC	RMSE	MAE	TT (s)	PT(s)
HLAAWMSAN-STHGOCVT + ACTCN	93.89	91.80	90.94	91.36	95.87	97.01	0.4519	0.0991	123.89	524.78
HLAAWMSAN-STHGOCVT + ACTCN + ResNet	95.56	93.14	92.31	92.72	94.69	94.78	0.3617	0.0847	129.34	482.36
HLAAWMSAN-STHGOCVT+ ACTCN + ResNet +CNN-AE	96.64	95.23	94.16	94.69	95.14	96.41	0.3585	0.0931	135.06	469.74
HLAAWMSAN-STHGOCVT ACTCN + ResNet + CNN-AE + CNN	97.71	96.01	96.59	96.29	97.41	96.96	0.3010	0.7913	144.88	459.11
HLAAWMSAN-STHGOCVT ACTCN + ResNet + CNN-AE + CNN + 3DCNN	98.98	97.77	97.12	97.39	97.20	95.14	0.4091	0.0784	149.45	438.41
HLAAWMSAN-STHGOCVT +PP	99.15	98.36	100	99.17	98.21	97.54	0.2486	0.0657	155.87	411.74

Table VI represents pre-processing pipeline comparison. The pre-processing pipeline provides the highest accuracy of 99.15% and achieves the optimal balance of performance and efficiency. The combined model with all pre-processing steps such as ACTCN, ResNet, CNN-AE, FFT, 3DCNN produced the highest scores across all metrics. Each pre-processing module increased the computational depth in training with increased model performance. The overall processing time is decreased by pruning the redundant computation, parallel processing by 3DCNN, CNN-AE reduced dimensionality. The fully combined pipeline increases feature extraction, robustness and highlights fine-grained manipulation details across spatial, color, and temporal domains, significantly enhancing detection accuracy and minimizing error.

The pre-processing pipeline enhances artifact visibility by amplifying spatial, frequency and temporal inconsistencies, enabling the detector to operate on higher representations. The HLAAWMSAN module strengthens interpretability by using localized, adaptively weighted multi-scale attention. The HLAAWMSAN selectively emphasizes regions that are susceptible to manipulation, such as facial boundaries, eye regions, and mouth movements and suppresses background and identity-specific information. This localized attention mechanism reduces the risk of spurious correlations and helps the model focus on semantically meaningful manipulation artifacts, and increases robustness to unseen deepfake techniques.

Compared to lightweight CNN-based models such as MesoNet and EfficientNet-B0, which rely on spatial features, the proposed framework incorporates targeted artifact enhancement and temporal modeling and maintains computational feasibility through a modular design. The proposed approach distributes complexity across task-specific modules, achieving improved detection robustness without a proportional increase in inference cost. The proposed approach captures manipulation inconsistencies that lightweight CNN overlook. The framework avoids excessive inference complexity by using shallow, lightweight operations at each stage and by focusing computation only on manipulated facial regions. The proposed method achieves a balance between detection robustness and computational efficiency, offering improved generalization and maintained practical deployability compared to lightweight CNN.

The detection model effectively handles the computational overhead associated with Deepfake detection systems by combining advanced techniques in pre-processing and classification stages. The pre-processing pipeline includes various modules such as ACTCN for face alignment, multi-scale ResNet for enhancing image resolution, CNN-based autoencoder for detecting color inconsistencies, Laplacian and FFT filters for uncovering hidden artifacts, and 3D-CNNs for capturing motion irregularities. These modules are computationally intensive but these models are integrated for parallel GPU execution and batch processing, which reduces the overall time and resource usage. The modular structure also allows flexible activation of components based on the system's available resources that is adaptable for real-time applications.

In the classification stage, HLAAWMSAN uses attention mechanisms to focus on manipulated facial areas; STHGOCVT combines hypergraph convolution and ViT to learn spatial and temporal features. The model integrates STO to automatically fine-tune parameters, leading to faster training and a compact model during inference. The model achieves the lowest processing time. The model overcomes the computational challenges by combining detection modules with efficiency-focused strategies like modularization, GPU parallelism, and optimization through STO. This makes it highly suitable for large-scale deployment and real-time Deepfake detection, offering performance and speed without compromising accuracy.

TABLE IX. COMPARISON OF PROPOSED MODEL WITH EXISTING MODELS

Model	VC	Accuracy	Precision	Recall	F1 score	AUC	RMSE	MAE	PT (s)
Xception-ConvLSTM [5]	2000	92.43	93.59	92.52	93.05	93.98	0.3436	0.0116	446.51
efficientNetV2S-ViT[9]	5200	92.16	93.45	91.24	92.22	99.56	0.4326	0.0902	467.92
ViViT [16]	1123	88.03	89.74	90.74	90.23	97.25	0.2647	0.0721	431.69
Swin-fake [21]	2000	93.7	92.65	93.41	93.02	93.72	0.3784	0.1014	423.35
MeST-Former [23]	1100	72.16	70.25	71.52	70.87	82.51	0.2659	0.1209	459.36
CNN-RNN-PSO [24]	5244	94.02	94.53	93.40	94.53	95.83	0.4297	0.1410	506.29
BMNet [31]	2654	84.72	89.74	87.69	88.70	88.51	0.3345	0.1319	439.24
AVENUE [34]	4000	94.24	95.74	94.26	94.99	84.35	0.1536	0.1815	452.08
Ensemble DL [36]	400	91.14	92.17	91.45	91.80	93.77	0.2469	0.1247	512.43
QTL-CaiT [49]	25000	90	91	90	90	94	0.3639	0.1645	519.87
DiffconvNet [57]	2000	92.9	92.45	93.66	93.05	96.2	0.4746	0.1554	492.11
CoDeiT [60]	5244	86.9	88.74	90.74	89.72	95	0.3364	0.0975	486.77
HLAAWMSAN-STHGOCVT	3432	99.15	98.36	100	99.17	98.21	0.2486	0.0657	411.74

Table VIII compares the related works with detection model. The model achieves the highest accuracy (99.15%) and the lowest processing time (411.74s) among all models. The model balances precision, efficiency, and robustness, due to attention mechanism, spatial-temporal modeling and optimization with STO. The model attained higher accuracy that is increased by 4.91% to 26.99% and lower processing time decreased up to 26.17%. The detection model demonstrated higher performance over existing Deepfake detection models in both accuracy and computational efficiency.

The detection model outperforms individual components by integrating spatial attention and temporal feature modeling, ensuring improved generalization and Deepfake artifact detection. The sequence demonstrates the model's ability to accurately identify manipulated regions through localized artifacts attention and increase facial regions and details through multi-scale feature fusion. The detected Deepfake frames highlight anomalous facial features that demonstrate the effectiveness of heatmap attention and hypergraph temporal modeling in identifying subtle inconsistencies.

The STHGOCVT model attained high accuracy and it is evaluated on benchmark dataset and real-time video data that demonstrates consistent performance across different data distributions. The multi-scale feature extraction and localized adaptive attention mechanisms focus on localized artifacts than memorizing global patterns. The temporal modeling using 3D-CNN enforces frame-to-frame consistency, which increases generalization across videos. The hypergraph convolutional networks gather higher-order relationships among multiple manipulated regions, which decreases sensitivity to noise and dataset-specific patterns. The STO algorithm prevents excessive parameters tuning by guiding the model toward globally optimal hyperparameter configuration. The model demonstrated stable performance in varying data distributions. These factors mitigate overfitting and increase generalization to unseen Deepfake scenarios.

The proposed framework focuses on forensic inconsistencies that persist across synthesis methods, including localized texture irregularities, frequency-domain noise, chrominance distortions, and temporal incoherence. The pre-processing pipeline explicitly exposes these artifacts before detection, reducing dependence on surface-level. The localized multi-scale attention mechanism emphasizes vulnerable facial regions and suppresses identity and background bias, mitigating overfitting to known manipulation styles. Spatio-temporal modeling using hypergraph convolution and ViT enforces consistency constraints across frames, enabling the detection of subtle anomalies that emerging deepfake generators fail to maintain. This detection framework shift from appearance-based classification to forensic anomaly analysis, which increases resilience against evolving and previously unseen deepfake attacks.

The model provides implementation details, including network architecture descriptions, pre-processing steps, hyperparameter settings, and optimization strategies, which enable independent reproduction of the proposed framework. The source code and trained models are available. All critical components required for re-implementation are explicitly documented, and the authors are committed to sharing additional implementation clarifications upon reasonable academic request.

A limitation of this study is the absence of comprehensive cross-dataset and ablation evaluations. Future work will use a curated dataset from three public deepfake datasets to assess cross-dataset generalization and the contribution of individual model components.

future work will extend the proposed framework through cross-dataset evaluation on additional benchmark datasets and testing against unseen manipulation techniques, compression, blur, poor illumination, and facial occlusion. The model will also be optimized to reduce memory usage and inference latency for continuous live-stream and edge-device deployment. Furthermore, integrating audio-visual information and explainable attention maps may improve the robustness, interpretability, and practical reliability of the detection system.

5.CONCLUSION

This paper addresses the limitations by proposing a, efficient, and robust Deepfake detection framework with a pre-processing pipeline. The proposed pre-processing pipeline improved the input data quality by applying multiple enhancement stages. These combined processes extract and expose Deepfake artifacts such as unnatural edges, inconsistent lighting, and motion irregularities at multiple spatial, temporal, and frequency levels, generating high-quality feature-rich inputs. The HLAAWMSAN module leverages attention mechanism and adaptively weighted multi-scale attention mechanisms highlighted manipulated regions while suppressing irrelevant information. The STHGOCVT module integrates HGCN to model complex spatial relationships and ViT to capture long-range temporal inconsistencies across video frames. The detection framework is optimized using the STO algorithm that increased convergence speed, reduces parameter sensitivity, and enhances detection performance. The model demonstrated highest accuracy of 99.15% with 100% recall, confirming robust detection without false negatives and lower processing time. This research contributes a hybrid framework that enhanced detection and reduced computational complexity with reliable deployment of Deepfake detection in critical domains such as media verification and cybersecurity.

The proposed framework is suited for real-world deployment in cybersecurity-sensitive applications. In social media monitoring, the model can assist platforms in automatically identifying and flagging manipulated videos before large-scale dissemination, helping to mitigate misinformation and reputational harm. In digital forensic analysis, the spatio-temporal design supports post-event verification of video authenticity, providing investigators with reliable forensic cues for evidence assessment. The modular and scalable architecture enables integration into content moderation pipelines and surveillance systems. These deployments highlight the practical relevance of the proposed approach in addressing emerging threats posed by malicious synthetic media.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

No funding or financial support was received to carry out the research presented in this paper.

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Dr. Salwani Binti Abdullah and Dr Kahlid Shakir , for their continuous guidance, valuable feedback, and support throughout the completion of this work.

References

[1] M. Westerlund, “The emergence of Deepfake technology: A review. Technology innovation management review,” vol. 9, pp. 11, 2019. doi: 10.22215/timreview/1282.

[2] J. W. Seow, M. K. Lim, R. C. Phan, and J. K. Liu, “A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities,” Neurocomputing, vol. 513, pp. 351-371, 2022. https://doi.org/10.1016/j.neucom.2022.09.135

[3] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, “Deepfake detection: A systematic literature review,” IEEE access, vol. 10, pp. 25494-25513, 2022. https://doi.org/10.1109/ACCESS.2022.3154404

[4] A. Heidari, N. Jafari Navimipour, H. Dag, and M. Unal, “Deepfake detection using deep learning methods: A systematic and comprehensive review,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 14, no. 2, pp. e1520, 2024. https://doi.org/10.1002/widm.1520

[5] T. Zhang, “Deepfake generation and detection, a survey,” Multimedia Tools and Applications, vol. 81, no. 5, pp. 6259-6276, 2022. https://doi.org/10.1007/s11042-021-11733-y

[6] Z. Xia, T. Qiao, M. Xu, X. Wu, L. Han, and Y. Chen, “Deepfake Video Detection Based on MesoNet with Preprocessing Module,” Symmetry, vol. 14, no. 5, Art. no. 939, 2022, doi: 10.3390/sym14050939.

[7] L. Deng, H. Suo, and D. Li, “Deepfake video detection based on EfficientNet-V2 network,” Computational Intelligence and Neuroscience, vol. 2022, Art. no. 3441549, 2022, doi: 10.1155/2022/3441549.

[8] L. Kuang, Y. Wang, T. Hang, B. Chen, and G. Zhao, “A dual-branch neural network for DeepFake video detection by detecting spatial and temporal inconsistencies,” *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42591–42606, 2022, doi: 10.1007/s11042-021-11539.

[9] A. Raza, K. Munir, and M. Almutairi, “A novel deep learning approach for deepfake image detection,” *Applied Sciences*, vol. 12, no. 19, Art. no. 9820, 2022, doi: 10.3390/app12199820.

[10] B. Chen, T. Li, and W. Ding, “Detecting deepfake videos based on spatiotemporal attention and convolutional LSTM,” *Information Sciences*, vol. 601, pp. 58–70, 2022, doi: 10.1016/j.ins.2022.04.014.

[11] G. Pang, B. Zhang, Z. Teng, Z. Qi, and J. Fan, “MRE-Net: Multi-Rate Excitation Network for Deepfake Video Detection,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 8, pp. 3663–3676, 2023, doi: 10.1109/TCSVT.2023.3239607.

[12] Yu, Y., Ni, R., Zhao, Y., Yang, S., Xia, F., Jiang, N., & Zhao, G. (2023). MSVT: Multiple spatiotemporal views transformer for DeepFake video detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(9), 4462-4471.doi: 10.1109/TCSVT.2023.3281448.

[13] Zhao, C., Wang, C., Hu, G., Chen, H., Liu, C., & Tang, J. (2023). ISTVT: Interpretable spatial-temporal video transformer for deepfake detection. *IEEE Transactions on Information Forensics and Security*, *18*, 1335-1348.doi: 10.1109/TIFS.2023.3239223.

[14] Deng, L., Wang, J., & Liu, Z. (2023). Cascaded network based on EfficientNet and transformer for deepfake video detection. *Neural Processing Letters*, *55*(6), 7057-7076.doi: 10.1007/s11063-023-11249-6.

[15] Lu, T., Bao, Y., & Li, L. (2023). Deepfake Video Detection Based on Improved CapsNet and Temporal-Spatial Features. Computers, Materials and Continua, 2023, 75 (1): 715-740. doi: 10.32604/cmc.2023.034963**.**

[16] Masud, U., Sadiq, M., Masood, S., Ahmad, M., & Abd El-Latif, A. A. (2023). LW-DeepFakeNet: a lightweight time-distributed CNN-LSTM network for real-time DeepFake video detection. *Signal, Image and Video Processing*, *17*(8), 4029-4037.doi: 10.1007/s11760-023-02633-9.

[17] Qadir, A., Mahum, R., El-Meligy, M. A., Ragab, A. E., AlSalman, A., & Awais, M. (2024). An efficient deepfake video detection using robust deep learning. *Heliyon*, *10*(5). doi: 10.1016/j.heliyon.2024.e25757.

[18] Javed, M., Zhang, Z., Dahri, F. H., & Laghari, A. A. (2024). Real-time deepfake video detection using eye movement analysis with a hybrid deep learning approach. *Electronics*, *13*(15), 2947.doi: 10.3390/electronics13152947.

[19] Alhaji, H. S., Celik, Y., & Goel, S. (2024). An approach to deepfake video detection based on ACO-PSO features and deep learning. *Electronics*, *13*(12), 2398.doi: 10.3390/electronics13122398.

[20] Omar, K., Sakr, R. H., & Alrahmawy, M. F. (2024). An ensemble of CNNs with self-attention mechanism for DeepFake video detection. *Neural Computing and Applications*, *36*(6), 2749-2765.doi: 10.1007/s00521-023-09196-3.

[21] Ramadhani, K. N., Munir, R., & Utama, N. P. (2024). Improving video vision transformer for deepfake video detection using facial landmark, depthwise separable convolution and self-attention. *IEEE Access*, *12*, 8932-8939.doi: 10.1109/ACCESS.2024.3352890.

[22] Cunha, L., Zhang, L., Sowan, B., Lim, C. P., & Kong, Y. (2024). Video deepfake detection using Particle Swarm Optimization improved deep neural networks. *Neural Computing and Applications*, *36*(15), 8417-8453.doi: 10.1007/s00521-024-09536-x.

[23] Kaddar, B., Fezza, S. A., Akhtar, Z., Hamidouche, W., Hadid, A., & Serra-Sagristà, J. (2024). Deepfake detection using spatiotemporal transformer. *ACM Transactions on Multimedia Computing, Communications and Applications*, *20*(11), 1-21.DOI: 10.1145/3643030.

[24] Tipper, S., Atlam, H. F., & Lallie, H. S. (2024). An investigation into the utilisation of CNN with LSTM for video deepfake detection. *Applied Sciences*, *14*(21), 9754.DOI: 10.3390/app14219754.

[25] Boongasame, L., Boonpluk, J., Soponmanee, S., Muangprathub, J., & Thammarak, K. (2024). Design and Implement Deepfake Video Detection Using VGG-16 and Long Short-Term Memory. *Applied Computational Intelligence and Soft Computing*, *2024*(1), 8729440.DOI: 10.1155/2024/8729440.

[26] Gong, L. Y., Li, X. J., & Chong, P. H. J. (2024). Swin-fake: A consistency learning transformer-based deepfake video detector. *Electronics*, *13*(15), 3045.DOI: 10.3390/electronics13153045.

[27] Liu, B., Liu, B., Ding, M., & Zhu, T. (2024). MeST-Former: Motion-enhanced spatiotemporal transformer for generalizable deepfake detection. *Neurocomputing*, *610*, 128588.DOI: 10.1016/j.neucom.2024.128588.

[28] Al-Adwan, A., Alazzam, H., Al-Anbaki, N., & Alduweib, E. (2024). Detection of deepfake media using a hybrid CNN–RNN model and particle swarm optimization (PSO) algorithm. *Computers*, *13*(4), 99. DOI: 10.3390/computers13040099.

[29] Xu, J., Liu, X., Lin, W., Shang, W., & Wang, Y. (2025). Localization and detection of deepfake videos based on self-blending method. *Scientific Reports*, *15*(1), 3927. DOI: 10.1038/s41598-025-88523-1.

[30] Sohail, S., Sajjad, S. M., Zafar, A., Iqbal, Z., Muhammad, Z., & Kazim, M. (2025). Deepfake image forensics for privacy protection and authenticity using deep learning. *Information*, *16*(4), 270. DOI: 10.3390/info16040270.

[31] Zafar, F., Khan, T. A., Akbar, S., Ubaid, M. T., Javaid, S., & Kadir, K. A. (2025). A hybrid deep learning framework for deepfake detection using temporal and spatial features. *IEEE Access*, *13*, 79560-79570. DOI: 10.1109/ACCESS.2025.3566008.

[32] Agrawal, D. R., & Haneef, F. (2025). Eye blinking feature processing using convolutional generative adversarial network for deepfake video detection. *Transactions on Emerging Telecommunications Technologies*, *36*(3), e70083. DOI: 10.1002/ett.70083.

[33] Siddiqui, F., Yang, J., Xiao, S., & Fahad, M. (2025). Enhanced deepfake detection with DenseNet and Cross-ViT. *Expert Systems with Applications*, *267*, 126150. DOI: 10.1016/j.eswa.2024.126150.

[34] Usmani, S., Kumar, S., & Sadhya, D. (2025). Spatio-temporal knowledge distilled video vision transformer (STKD-VViT) for multimodal deepfake detection. *Neurocomputing*, *620*, 129256. DOI: 10.1016/j.neucom.2024.129256.

[35] Xiong, D., Wen, Z., Zhang, C., Ren, D., & Li, W. (2025). BMNet: Enhancing deepfake detection through bilstm and multi-head self-attention mechanism. *IEEE Access*, *13*, 21547-21556. DOI: 10.1109/ACCESS.2025.3533653.

[36] Birla, L., Saikia, T., & Gupta, P. (2025). AVENUE: A novel deepfake detection method based on temporal convolutional network and rPPG information. *ACM Transactions on Intelligent Systems and Technology*, *16*(1), 1-16. DOI: 10.1145/3702232.

[37] Long, M., Liu, Z., Zhang, L. B., & Peng, F. (2025). LGDF-Net: Local and global feature-based dual-branch fusion networks for deepfake detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *35*(6), 5489-5500. DOI: 10.1109/TCSVT.2025.3530402.

[38] Pintelas, E., Livieris, I. E., & Pintelas, P. E. (2025). Quantization-based 3D-CNNs through circular gradual unfreezing for DeepFake detection. *IEEE Transactions on Artificial Intelligence*, *6*(12), 3351-3363.DOI: 10.1109/TAI.2025.3564903.

[39] Sagar, N. K., & Arukonda, S. (2025). A Novel CNN-LSTM Approach for Robust Deepfake Detection. *Procedia Computer Science*, *258*, 1844-1855. DOI: 10.1016/j.procs.2025.04.436.

[40] Katı, B. E., Küçüksille, E. U., & Sarıman, G. (2025). Enhancing deepfake detection through quantum transfer learning and class-attention vision transformer architecture. *Applied Sciences*, *15*(2), 525. DOI: 10.3390/app15020525.

[41] Uddin, M., Fu, Z., & Zhang, X. (2025). Deepfake face detection via multi-level discrete wavelet transform and vision transformer. *The Visual Computer*, *41*(10), 7049-7061. DOI: 10.1007/s00371-024-03791-8.

[42] Jin, X., Yu, W., Chen, D. W., & Shi, W. (2025). DFD-NAS: General deepfake detection via efficient neural architecture search. *Neurocomputing*, *619*, 129129. DOI: 10.1016/j.neucom.2024.129129.

[43] Kurniawan, W., Kurniasih, A., & Ghani, M. A. (2025). Real or deepfake face detection in images and video data using YOLOv11 algorithm. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, *4*(2), 1514-1521. DOI: 10.59934/jaiea.v4i2.939.

[44] Zheng, J., Zhou, Y., Zhang, N., Hu, X., Xu, K., Gao, D., & Tang, Z. (2025). A spatio-frequency cross fusion model for deepfake detection and segmentation. *Neurocomputing*, *628*, 129683. DOI: 10.1016/j.neucom.2025.129683.

[45] Pintelas, E., & Livieris, I. E. (2025). Convolutional neural network framework for deepfake detection: A diffusion-based approach. *Computer Vision and Image Understanding*, *257*, 104375. DOI: 10.1016/j.cviu.2025.104375.

[46] Sudarshana, K., & Vamsidhar, Y. (2025). UAM-Net: Robust Deepfake Detection Through Hybrid Attention Into Scalable Convolutional Network. *Expert Systems*, *42*(3), e70009. DOI: 10.1111/exsy.70009.

[47] Mashetty, H., Erukulla, N., Belidhe, S., Jella, N., Reddy Pishati, V., & Enesheti, B. K. (2025, January). Deep Fake Detection with Hybrid Activation Function Enabled Adaptive Milvus Optimization-Based Deep Convolutional Neural Network. In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)* (pp. 1159-1166). IEEE. DOI: 10.1109/ICMCSI64620.2025.10883193.

[48] Shi, Z., Liu, W., & Chen, H. (2025). Face reconstruction-based generalized deepfake detection model with residual outlook attention. *ACM Transactions on Multimedia Computing, Communications and Applications*, *21*(4), 1-19. DOI: 10.1145/3686162.

[49] Zakkam, J., Jayaraman, U., Sahayam, S., & Rattani, A. (2024, December). Codeit: Contrastive data-efficient transformers for deepfake detection. In *International Conference on Pattern Recognition* (pp. 62-77). Cham: Springer Nature Switzerland. DOI: 10.1007/978-3-031-78125-4_5.

[50] Bahadure, S., & Mane, V. (2026). Deepfake detection using deep convolutional neural network and long short-term memory. *Multimedia Tools and Applications*, *85*(3), 259. DOI: 10.1007/s11042-026-21375-7.

[51] Siagian, O. L., Ebenhaizer, R., Wicaksono, P., & Izdihar, Z. N. (2026). Improving Vision Transformer for Deepfake Detection. *Journal of Image and Graphics*, *14*(1). DOI: 10.18178/joig.14.1.76-83.

[52] Adinarayana, M. R., Harini, B., Yeswanth, S., Pranavanth, V. D., & Rao, K. B. (2026). Multi-model Deepfake Detection using Vision Transformers (ViT) and Hybrid Deep Learning Techniques. *International Journal of Data Science and IoT Management System*, *5*(2), 98-105. DOI: 10.64751/ijdim.2026.v5.n2.pp98-105.

[53] Raghavajosyula, V., & Pawar, D. (2026). Hybrid deep learning architecture for comprehensive deepfake video facial forgery detection and segmentation. *Multimedia Tools and Applications*, *85*(1), 2. DOI: 10.1007/s11042-026-21320-8.

[54] Tilagul, A., Mohan, D. N., PN, K. K., Lahari, K. N., Sinchana, R., & Prakruthi, K. M. (2026, April). DeepFake Face Detection Using Machine Learning: A Hybrid CNN-LSTM Approach. In *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)* (pp. 1-5). IEEE. DOI: 10.1109/ICKECS70176.2026.11528004.

[55] Sardhara, A., Vekariya, V., Pathak, A. R., & Dash, S. (2026). DeepForgeryNet: a hybrid CNN–LSTM and transfer learning framework for robust image forgery and deepfake detection. *Frontiers in Artificial Intelligence*, *9*, 1810912. DOI: 10.3389/frai.2026.1810912.

[56] P. Trojovský, M. Dehghani, and P. Hanuš, Siberian tiger optimization: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. IEEE Access, vol. 10, pp. 132396-132431, 2022. https://doi.org/10.1109/ACCESS.2022.3229964

[57] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The Deepfake detection challenge (dfdc) dataset. arXiv preprint arXiv:2006.07397, 2020. https://doi.org/10.48550/arXiv.2006.07397

[58] H. Zhang, H. Huang and H. Han, "Cascade Temporal Convolutional Network for Multitask Learning," 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 2023, pp. 261-269, doi: 10.1109/ICAIBD57115.2023.10206379.

[59] A. Yadav, and D. K. Vishwakarma, "AW-MSA: Adaptively weighted multi-scale attentional features for Deepfake detection," Engineering Applications of Artificial Intelligence, vol. 127, pp. 107443, 2024 https://doi.org/10.1016/j.engappai.2023.107443

[60] B. Shah, M. K. Guragai, and A. Verma, "Image colorization using AutoEncoder," in *Proc. 13th IOE Graduate Conf.*, vol. 13, Institute of Engineering, Tribhuvan University, Nepal, Apr. 2023, pp. 307–310.

[61] S. Kingra, N. Aggarwal, and N. Kaur, "SFormer: An end-to-end spatio-temporal transformer architecture for Deepfake detection," Forensic Science International: Digital Investigation, vol. 51, pp. 301817, 2024. https://doi.org/10.1016/j.fsidi.2024.301817

[62] D. Nguyen, N. Mejri, I. P. Singh, P. Kuleshova, M. Astrid, A. Kacem, and D. Aouada, "Laa-net: Localized artifact attention network for quality-agnostic and generalizable Deepfake detection," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 17395-17405, 2024. https://doi.org/10.48550/arXiv.2401.13856.