Deepfake Video Detection Using Generative Convolutional Vision Transformer

Deressa Wodajo
Jimma University
derssa.wodajo@ju.edu.et

Solomon Atnafu
Addis Ababa University
solomon.atnafu@aau.edu.et

Zahid Akhtar
State University of New York Polytechnic Institute, USA
akhtarz@sunypoly.edu

Abstract

Deepfakes have raised significant concerns due to their potential to spread false information and compromise digital media integrity. In this work, we propose a Generative Convolutional Vision Transformer (GenConViT) for deepfake video detection. Our model combines ConvNeXt and Swin Transformer models for feature extraction, and it utilizes Autoencoder and Variational Autoencoder to learn from the latent data distribution. By learning from the visual artifacts and latent data distribution, GenConViT achieves improved performance in detecting a wide range of deepfake videos. The model is trained and evaluated on DFDC, FF++, DeepfakeTIMIT, and Celeb-DF (v2) datasets, achieving high classification accuracy, F1 scores, and AUC values. The proposed GenConViT model demonstrates robust performance in deepfake video detection, with an average accuracy of 95.8% and an AUC value of 99.3% across the tested datasets. Our proposed model addresses the challenge of generalizability in deepfake detection by leveraging visual and latent features and providing an effective solution for identifying a wide range of fake videos while preserving media integrity. The code for GenConViT is available here.

Keywords: Deep Learning, Deepfakes, Deepfakes Detection, Vision Transformer, Generative models

1. Introduction

Deepfakes are hyper-realistic manipulated media generated using various advanced Deep Learning (DL) techniques. Deepfake videos are produced by superimposing one person's facial features onto another person's face through techniques such as face replacement, facial reenactment, face editing, and complete face synthesis [46, 63], among others. Recently, deepfakes have become a public concern due to their potential for misuse and the spread of false information [10, 24, 72]. Various Deepfake creation methods are readily available for use, and anyone interested can use them to easily modify existing media, presenting a false representation of reality where individuals appear to be speaking or performing actions that never actually took place. Additionally, the manipulation of the videos is done on a frame-by-frame basis, making deepfakes seem more realistic and believable. Consequently, deepfake creation techniques have been used to manipulate images and videos, causing celebrities and politicians to be depicted as saying or doing things that are untrue.

The applications of Deepfake videos can range from creative uses such as creating realistic special effects or replacing an actor in the film and entertainment industry, to potentially harmful attacks, such as using deepfakes to create misleading videos for criminal purposes [51, 35]. In the realm of politics, deepfakes have been utilized to create fake political videos, war propaganda, and videos with the intention of influencing elections. This has led to widespread concern about the potential for malicious actors to use deepfakes to spread false information, erode the credibility of political candidates, or even pose a threat to national security. The dissemination of misleading and false information through deepfakes can have real-life consequences, making it imperative to address the need for accurate and reliable deepfake video detection techniques.

Deepfake video detection attempts to detect if a given video has been tampered with or manipulated. The challenge of detecting deepfakes has inspired researchers and technology companies to develop various deep-learning methods to identify tampered videos [65, 33, 10, 46, 43]. Currently, deepfake detection methods often rely heavily on visual features [1, 70, 64, 47]. The visual features-based detection can be on a frame-per-frame basis [21] or temporal relationship between differences between frames through time [17, 28]. However, these methods mostly encounter challenges in detecting deepfakes that differ from the train-

ing data, thus, failing sufficient detection. In recent years, the rapid progress in advanced generative models like Generative Adversarial Networks (GAN) [15, 27, 48], Variational Autoencoders (VAE) [31, 32], and Diffusion Models (DM) [11] has made it even more challenging to detect deepfakes based on visual artifacts alone since the deepfakes leave small to no trace of visual clues that help to distinguish them from the images from the real world. Several authors have proposed alternative detection methods, such as utilizing biological signals [8], geometric features [69], frequency information [66], spatial features with temporal information [73, 30] and generative adversarial network fingerprints [71], as potential solutions to the difficulties of detecting deepfakes. Another approach to spotting deepfake videos involves examining the consistency of pixels or groups of similar pixels [18].

Current Deepfake video detection methods have a competitive result in identifying manipulated videos, but they often fail to generalize in more diverse videos, particularly in environments with different facial poses, light angles, and movements. Based on our observation of current detection methods, we propose a novel architecture called the Generative Convolutional Vision Transformer (GenConViT), aimed at detecting a diverse set of fake videos. Our proposed architecture leverages both visual artifacts and latent data distribution of the data in its detection process. GenConViT has two main components: the generative part and the feature extraction part.

The generative part utilizes two different models (i.e., an Autoencoder (AE) [3] and VAE), to learn the latent data distribution of the training data. The feature extraction component employs ConvNext [45] and the Swin Transformer [44] to extract relevant visual features. Our proposed model addresses the limitations of current detection methods by learning both the visual artifacts and the latent data distribution, making it capable of detecting a wider range of fake videos. Extensive experimental results demonstrate that GenConViT has competitive results compared to the other models proposed for deepfake detection.

In this paper, we propose the following contributions:

- GenConViT: We propose Generative Convolution Vision Transformer for deepfake video detection. We use AE and VAE to generate an image, extract the visual features of the original and generated image using ConvNext and the Swin Transformer, and then use the extracted features to detect whether the video is a deepfake video.
- By training our model on a large dataset and detecting both visual and latent features, our proposed method aims to fill in the gap of generalizability of deepfake detection, compared to previously proposed models.

The reminder of this article is structured as follows. An overview of the existing works is presented in Section 2. The proposed Generative Convolutional Vision Transformer (GenConViT) deepfake detection framework and datasets are detailed in Section 3. Experiments are discussed in Section 4. The conclusions are drawn in Section 5.

2. Related Work

In recent years, deepfakes have become more and more realistic as well as becoming harder to detect. In this section, we will discuss the various deep learning methods used to create and detect deepfakes.

2.1. Deepfake Generation

Deepfake videos are created by advanced Deep Learning (DL) methods, such as GANs, VAE, and DMs. Deepfake was first introduced in 2017 on the social media platform Reddit by a user called Deepfakes to showcase a DL technique he had developed to generate Deepfake videos [51]. The videos and the accompanying code he released garnered significant attention, and soon people started to explore other ways to create a hyper-realistic video. Some widely used DL techniques to create deepfakes include face synthesis, face reenactment, face replacement, identity swapping, attribute manipulation, and expression swap, to name a few. GANs are a type of GM that consists of two Neural Networks (NNs), a generator G, which takes in random noise and produces synthetic data, and the discriminator D, which determines whether the input sample is real or fake. Another NN used to create Deepfakes is VAE, which consists of an Encoder and a Decoder NNs. The Encoder encodes the input image into a lower dimensional latent space, and the decoder reconstructs an image from the latent space. Convolutional Neural Networks (CNN) [52, 19] are also used for face synthesis by learning the mapping between a face image and its attributes, such as facial expression, age, and gender.

Face synthesis [25] is a method of synthesizing desired non-existent face images based on a given input image. In Face synthesis, generator G generates a face image, and Discriminator D discriminates whether the sample is real or fake. Face reenactment [61, 22, 58] is a face synthesis method that transfers source face attributes to a target face while preserving the appearance and identity of the target face's features. Face2Face [61] is a real-time face reenactment technique that animates the facial expressions of the target video by using a source actor and generates manipulated output video. Some other examples of face synthesis methods include frontal view synthesis, changing the facial pose from an input image, altering facial attributes, aging the face to create diverse and realistic results. Face swapping [36, 37, 75] is the process of replacing a person's face

in an image or video with the face of another person to creates a non-existent realistic face.

Several different types of GANs have been proposed for face synthesis, including Progressive GANs [26], Wasserstein GANs [2], and Style-Based GANs [27]. Progressive GANs allow for the generation of high-resolution images by gradually increasing the resolution of the generated images throughout the training process. Wasserstein GANs improve the stability of the GAN training process by using a different loss function. Style-Based GANs allow for the control of certain style aspects of the generated faces, such as facial expression and hair style. Conditional GANs [48], also known as supervised GANs, use labeled data to generate facial semantics. Paired image-to-image translation GANs [23, 53] are a type of conditional GAN that translate an input image from one domain to another, given input-output pairs of images as training data. Pix2Pix is one example of a paired image-to-image translation GAN.

2.2. Deepfake Detection

A key question in the deepfake detection pipeline is determining whether a given video is fake or real. With the advancement of deep learning methods that can create hyper-realist images, deepfake detection has increasingly become a challenging task. To this effect, various authors have proposed deepfake detection techniques that use different approaches, including visual features, biological signals, and frequency information, to name a few[16].

Several deepfake detection methods rely on extracting visual features from manipulated videos. MesoNet [1] is a deepfake detection methods that uses CNNs to extract mesoscopic properties and identify deepfakes created with techniques like Deepfake [49] and Face2Face [61]. Nguyen et al. [50] proposed a model that incorporates a combination of VGG-19 and capsule networks to learn complex hierarchical representations for detecting various types of forgery, including FaceSwap [36], Facial Reenactment [61], replay attacks, and AI-generated videos. Yang et al. [70] proposed a model to compare 3D head poses estimated from all facial landmarks. The method considers that splicing synthesized face regions into original images can introduce errors in landmark locations. The landmark location errors can be detected by comparing the head poses estimated from the facial landmarks. Li and Lyu [40] proposed a CNN model for deepfake detection by identifying face-warping artifacts. Their method leverages the limitations of deepfake algorithms that generate face images of lower resolutions, which results in distinctive warping artifacts when the generated images are transformed to match the original faces in the deepfake creation pipeline. By comparing the Deepfake face region with surrounding pixels, resolution inconsistencies caused by face warping are identified. Li et al. [38] proposed a technique called Face X-ray that detects the blending boundaries of images and reveals whether an input face image has been manipulated by blending two images from different sources. Sun et al. [59] proposed a virtual-anchor-based approach to robustly extract the facial trajectory, capturing displacement information. They constructed a network utilizing dual-stream spatial-temporal graph attention and a gated recurrent unit backbone to expose manipulated videos. The proposed method achieves competitive results on the FaceForensics++ dataset, demonstrating its effectiveness in detecting manipulated videos.

In addition to visual features, researchers have explored the use of biological signals for deepfake detection. Y. Li [39] proposed a model that combines CNN and recursive neural network (LRCN) to detect deepfake videos by tracking eye blinking with previous temporal knowledge. Chintha et al. [7] proposed a modified XceptionNet architecture, which incorporates visual frames, edge maps, and dense optical flow maps alongside RGB channel data to target low-level features. The architecture isolates deepfakes at the instance and video levels, making the technique effective in detecting deepfakes. Zhao et al. [74] proposed a pair-wise self-consistency learning with an inconsistency image generator to train a ConvNet [45] that extracts local source features and measures their self-consistency to identify Deepfakes.

D. Kim and K. Kim [30] proposed a facial forensic framework that uses pixel-level color features in the edge region of an image and a 3D-CNN classification model to interpret the extracted color features spatially and temporally for generalized and robust face manipulation detection. Sabir et al. [55] proposed a Recurrent convolutional model, whereas [68] proposed a Convolutional Vision Transformer (CViT) model, Y. Heo [21] introduced an improved Vision Transformer model with vector-concatenated CNN feature and patch-based positioning while [20] used Vision Transformer with distillation and [9] combined EfficientNet and Vision Transformers (ViT) to detect deepfakes.

3. The Proposed Generative Convolutional Vision Transformer (GenConViT) Deepfake Detection Framework

In this section, we present the dataset we used, the preprocessing techniques, and our proposed Generative Convolutional Vision Transformer (GenConViT) for Deepfake video detection. The proposed model consists of three main steps: 1) video preprocessing, 2) feature extraction and reconstruction, and 3) video classification.

3.1. Datasets

In our work, we utilized five datasets (i.e., DFDC [13, 12], TrustedMedia (TM) [6], DeepfakeTIMIT (TIMIT) [34, 56], Celeb-DF (v2) [42, 41], and FaceForensics++

(FF++) [54]) to train, validate, and test our model. DFDC and FF++ are well-known benchmark datasets for deepfake detection, while TM is a relatively new dataset that provides a diverse range of deepfake manipulation techniques, primarily of Asian descent.

The DFDC dataset is the largest publicly available dataset and contains over $100,000$ high resolutions of real and fake videos. The dataset is created using $3,426$ volunteers, and the videos are captured in various natural settings, different angles, and lighting conditions. The dataset is created using eight deepfake creation techniques.

The FF++ dataset comprises $1,000$ original videos collected from YouTube, which have been manipulated using four automated face manipulation methods: Deepfakes, Face2Face, FaceSwap, and NeuralTextures [60]. The dataset includes a $c23$ and a $c40$ compression scheme and various video resolutions. The TM dataset consists of $4,380$ fake and $2,563$ real videos, with multiple video and audio manipulation techniques. The TM dataset is used only in the training phase. The Celeb-DF (v2) dataset consists of 890 real videos and $5,639$ videos deepfake videos. We used all datasets to train, validate, and test our model.

3.2. Video Preprocessing

The preprocessing component in Deep Learning (DL) plays a critical role in preparing raw datasets for training, validating, and testing DL models. The proposed model focuses on the face region, which is crucial in Deepfake generation and synthesis mechanisms. We preprocess the videos using a series of image processing operations. These operations include the following steps: (1) Extracting the face region from each videos using OpenCV, face_recognition [14], and BlazeFace [4] face recognition deep learning libraries; (2) Resizing the input image to a 224×224 RGB format, where the dimensions of the input image are $H \times W \times C$ with $H = 224$ representing the height, $W = 224$ representing the width, and $C = 3$ representing the RGB channels; and (3) verifying extracted face region images quality manually.

We extracted approximately 30 frames per video from each dataset to ensure diversity in our training data. To mitigate the ratio between fake and real videos in DFDC and TM datasets, we extracted a higher number of frames from the real videos. The DFDC dataset has a ratio of $6 : 1$ for fake to real videos, and TM has $2 : 1$ ratio. After the face regions were extracted, we manually review them. As noted in [12], deepfake videos may contain pristine frames within them and that the face region may not always be accurately detected by the deep learning frameworks used to extract them. To address this issue, we manually review the images and exclude images that did not contain a face or were deemed to be real image within the fake class, resulting in a total of $1,004,810$ images for training. This approach allowed us to curate a fake class dataset comprising only relevant and potentially manipulated face images. To train, validate, and test our model, the images were divided into a ratio of $80 : 15 : 5$, respectively. To evaluate our model's performance on videos, we held out $3,972$ videos from both DFDC, DeepfakeTIMIT, Celeb-DF (v2) , and FF++ datasets for testing. All the facial images are saved to a 224×224 RGB format.

3.3. Generative Convolutional Vision Transformer (GenConViT)

The Generative Convolutional Vision Transformer (GenConViT) model generates latent spaces of video frames and extracts visual clues and hidden patterns within them to determine if a video is real or fake. The proposed GenConViT model, as shown in Fig. 1, has two independently trained networks and four main modules: an Autoencoder (AE), a Variational Autoencoder (VAE), a ConvNeXt layer, and a Swin Transformer. The first network (A part in Fig. 1) includes an AE, a ConvNeXt layer, and a Swin Transformer, while the second network (B part in Fig. 1) includes a VAE, a ConvNeXt layer, and a Swin Transformer. The first network uses an AE to generate Latent Feature (LF) space of an input images to maximize the class prediction probability, indicating the likelihood that a given input is deepfake. The second network uses a VAE to reconstruct images to maximize class prediction probability and minimize the loss distance between the sample image and the reconstructed image. Both AE and VAE models extract LFs from input video frames, which capture hidden patterns and correlations present in the learned deepfake visual artifacts. The ConvNeXt and Swin transformer models form a hybrid model ConvNeXt-Swin. The ConvNeXt model acts as the backbone of the hybrid model, using a CNN to extract features from the input frames. The Swin Transformer, with its hierarchical feature representation and attention mechanism, further extracts the global and local features of the input images. The GenConViT two networks have each two ConvNeXt-Swin models, and both take in a 224×224 RGB image and an LF of AE (I_A) or VAE (I_B). The use of the ConvNeXt-Swin hybrid model enables the learning of relationships among the extracted LFs by the AE and VAE.

3.3.1 Autoencoder and Variational Autoencoder

An AE and a VAE are NNs that consist of two networks: an Encoder and a Decoder. The Encoder of AE maps an input image $X \in \mathbb{R}^{H \times W \times C}$ to a latent space $Z \in \mathbb{R}^{H' \times W' \times \mathrm{K}}$, where K is the number of channels (features) in the output and H' and W' are the height and width of the output feature map, respectively, and its decoder maps the latent space $Z \in \mathbb{R}^{H' \times W' \times \mathrm{K}}$ to an output image $X' \in \mathbb{R}^{H \times W \times C}$. The

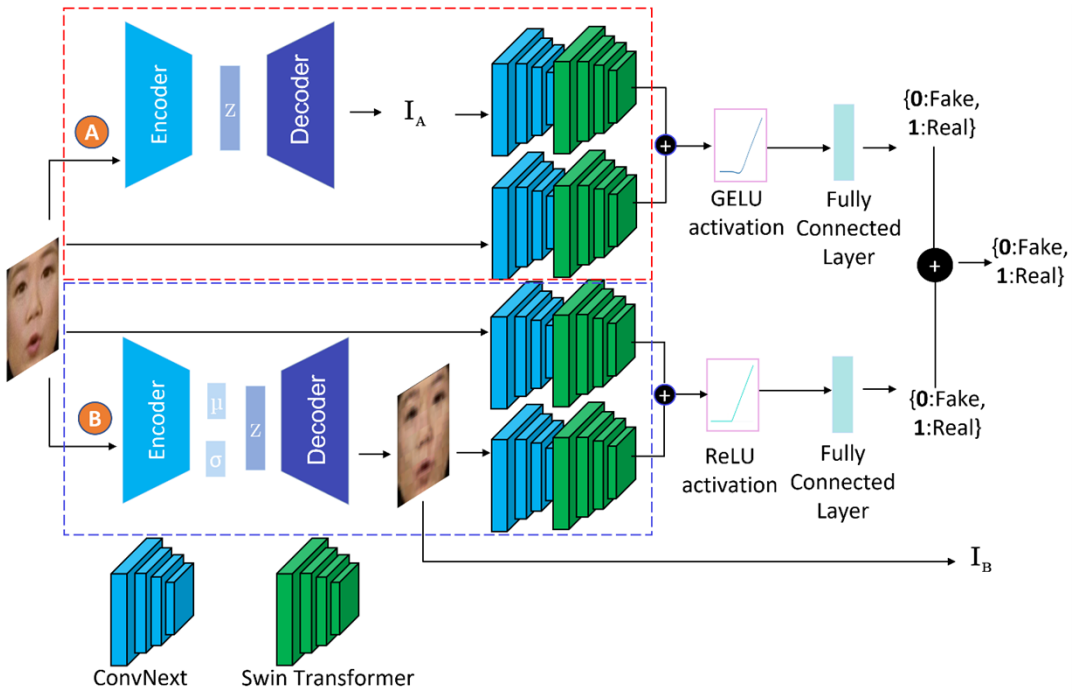


Figure 1: The Proposed Generative Convolutional Vision Transformer (GenConViT) Deepfake Detection Framework.

Encoder of VAE maps an input image X to a probability distribution over a latent space $Z' \in \mathbb{R}^K, Z' \sim \mathcal{N}(\mu, \sigma^2)$, and μ and σ^2 are the mean and variance of the learned distribution, respectively. The Encoder of the AE is composed of five convolutional layers with width starting from 3 up to 256, with kernels of size 3×3 and a stride of 2. Each convolutional layer is followed by ReLU non-linearity and Max-pooling of kernel size 2×2 and stride 2. The output of the Encoder is a $256 \times 7 \times 7$ down-sampled LF. The Decoder is composed of five transposed convolutional layers with width starting from 256 up to 3, with kernels of size 2×2 and stride of 2. Each transposed convolutional layer is followed by ReLU non-linearity. The output of the Decoder, I_A, is a reconstructed feature space of the input image with dimensions $H \times W \times C$. In this case, I_A has dimensions $224 \times 224 \times 3$. The detailed configuration is shown in Table 1.

The goal of VAE is to learn a meaningful latent representation of the input image and reconstruct the input image by performing random sampling of the latent space while minimizing the reconstruction loss. The Encoder of the VAE is composed of four convolutional layers with width starting from 3 up to 128, with kernels of size 3×3 and a stride of 2. Each convolutional layer is followed by Batch Normalization (BN) and LeakyReLU non-linearity. The output of the Encoder is a 1-dimensional vector of length 12544 representing the latent distributions. The Decoder is composed of four transposed convolutional layers with width starting from 256 up to 3, with kernels of size 2×2 and stride of 2. Each transposed convolutional layer is followed by LeakyReLU non-linearity. The output of the Decoder, I_B, is a reconstructed feature space of the input image with dimensions $H/2 \times W/2 \times C$. In this case, I_B has dimensions $112 \times 112 \times 3$. The detailed configuration is shown in Table 2. The choice of the convolutional layers for both the AE and VAE is due to the compute power and memory we had, model accuracy, extensive experiment, and training time during the training of our model.

Table 1: GenConViT model Autoencoder configuration.

Network	**AE Configuration**						
Encoder	**Conv-**					**Kernel**	**Stride**
	3-16	16-32	32-64	64-128	128-256	3	1
Decoder	**ConvTranspose-**						
	256-128	128-64	64-32	32-16	16-3	2	1

Table 2: GenConViT model Variational Autoencoder configuration.

Network	**VAE Configuration**					
Encoder		**Conv-**			**Kernel**	**Stride**
	3-16	16-32	32-64	64-128	3	2
Decoder	**ConvTranspose-**					
	256-64	64-32	32-16	16-3	2	2

3.3.2 ConvNeXt-Swin Hybrid

The ConvNeXt-Swin Transformer architecture is a hybrid CNN-Transformer model that combines the strengths of ConvNeXt and Swin Transformer architectures for deepfake detection task. The ConvNeXt model is a CNN architecture that has shown impressive performance in image recognition tasks, by extracting high-level features from images through a series of convolutional layers. The Swin Transformer is a transformer-based model that uses a self-attention mechanism to extract local and global features.

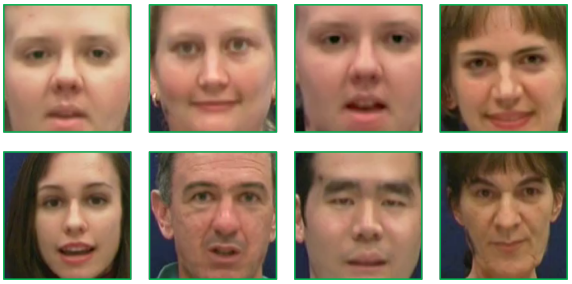
The GenConViT model leverages the strengths of both architectures by using ConvNeXt as the backbone for feature extraction and the Swin Transformer for feature processing. In our proposed method, the ConvNeXt architecture extracts high-level features from images, which are then passed through a HybridEmbed module to embed the features into a compact and informative vector. The resulting vector is then passed to the Swin Transformer model. The ConvNeXt backbone consists of multiple convolutional layers that extract high-level features from input images and the LFs from AE or VAE. We use pre-trained ConvNeXt and Swin Transformer models, which are trained on an ImageNet dataset.

After extracting learnable features by the ConvNeXt backbone, we pass the feature maps through a HybridEmbed module. The HybridEmbed module is designed to extract feature maps from the ConvNeXt, flatten them, and project them to an embedding dimension of 768. It consists of a 1×1 convolutional layer, which takes the feature maps from the backbone and reduces their channel dimension to the desired embedding dimension. The resulting feature maps are flattened and transposed to obtain a sequence of feature vectors, which are then further processed by the Swin Transformer.

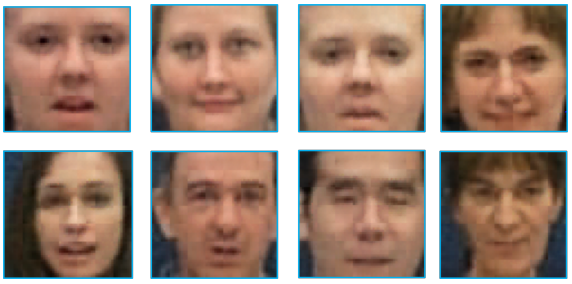
The GenConViT's network A consists of two Hybrid ConvNeXt-Swin models that take in a LF of size $224 \times 224 \times 3$ generated by the AE (I_A) and an input image of the same size. The models output a feature space of size $1,000$, which is then concatenated. Since the model has only two classes (real or fake), a linear mapping layer of size 2 is used. Network B has the same configuration as A, but it uses VAE and outputs both class prediction probability and the reconstructed image of size $112 \times 112 \times 3$.

4. Experiments

We conducted extensive experiments on various configurations of AE and VAE, as well as different variants of CNN and Transformer models. Our findings suggest that a hybrid architecture using ConvNeXt and Swin Transformer performs well. Due to our limited resources and large training dataset, we implemented the "tiny" model versions of both architectures.



(a) Input Sample Images



(b) Reconstructed Images by Network B

Figure 2: Generated Images (I_B) from Input Samples (a) using Network B (b)

4.1. Implementation Details

The network A and B were trained to classify real and fake videos while B was also trained to the reconstruct images, as shown in Fig. 2. Therefore, Network A is trained using Cross Entropy loss while Network B is trained using Cross Entropy and MSE loss. We used timm [67] library to load the class definitions and the weight of the pretrained ConvNext and Swin Transformer. We used two model variants, namely `convnext_tiny` and `swin_tiny_patch4_window7_224`, both trained on ImageNet-1k.

Both network, A and B, were trained using Adam optimizer with a learning rate of 0.0001 and weight decay of 0.0001. The Albumentation [5] library was used for data augmentation, and the following augmentation techniques were used with a strong augmentation rate of 90%: RandomRotate, Transpose, HorizontalFlip, VerticalFlip, GaussNoise, ShiftScaleRotate, CLAHE, Sharpen, IAAEmboss, RandomBrightnessContrast, and HueSaturationValue. The training data was normalized. The batch size for network A is set to 32 and for network B, it was set to 16. Both networks were trained on the DFDC, FF++, and TM datasets for 30 epochs. The training size of our deepfake data is as follows: train: $826,756$, valid: $130,948$, and test: $47,106$.

4.2. Experimental Results and Discussion

In this section, we present the experimental results and discuss the performance of our proposed GenConViT

model.

To assess GenConViT's performance, we used multiple evaluation metrics, including classification accuracy, F1 score, Receiver Operating Characteristic (ROC) curve, and Area Under the Curve (AUC) value. We evaluated the model using four different datasets: DFDC, FF++, DeepfakeTIMIT, and Celeb-DF (v2). The model was trained with $1,004,810$ images. Ninety percent of the images were augmented online. This resulted in a total of $1,909,139$ images. The model was tested on $3,972$ videos across these datasets. To predict our model's performance, we extracted 15 frames from each video. We averaged the result of both networks for the final prediction. The results demonstrate that GenConViT delivers strong performance across various datasets. Table 3 shows our model's accuracy on the training, validation, and testing dataset.

Table 3: GenConViT model accuracy on the training, validation, and testing dataset

Dataset	GenConViT(A)	GenConViT(B)	GenConViT
Train	99%	99%	-
Validation	97%	97%	-
Test	99.99%	99.99%	99.99%

We evaluated GenConViT classification accuracy for each of the dataset. Table 4 summarizes their results.

Table 4: GenConViT classification accuracy for each dataset.

Dataset	Accuracy (%)
DFDC	98.5
TIMIT	98.28
FF++	97
Celeb-DF (v2)	90.94

We evaluated GenConViT classification accuracy for each Real and False classes within each dataset. Table 5 provides the breakdown of these results.

Table 5: GenConViT classification accuracy for each Real and Fake class in each dataset.

Dataset	REAL (%)	FAKE (%)
DFDC	98.7	98.45
FF++	95.58	98.5
TIMIT	-	98.28
Celeb-DF (v2)	83	98.8

We evaluated the GenConViT performance by testing the classification accuracy for each Real and Fake class of the two variants of the network, GenConViT(A) and GenConViT(B) within each dataset. The results are summarized on Table 6.

Table 6: GenConViT classification accuracy for each Real and Fake class of the two Network's A and B in each dataset.

Dataset	GenConViT(A)	GenConViT(B)
DFDC	97.5%	98.45%
FF++	95.57%	96.8%
TIMIT	97.65%	97.81%
Celeb-DF (v2)	85.42%	83.97%

Table 7 summarizes the classification accuracy for each Real and Fake classes of the two networks': GenConViT(A) and GenConViT(B) within each dataset:

Table 7: GenConViT classification accuracy for Real and Fake classes in each dataset.

Dataset	GenConViT(A)		GenConViT(B)	
	REAL	FAKE	REAL	FAKE
DFDC	98.7%	97.2%	98.7%	95.52%
FF++	94.12%	95.56%	95.58%	98.02%
TIMIT	-	97.5%	-	97.8%
Celeb-DF (v2)	70.22%	93.38%	55%	99.38%

We compared our results with other state-of the-art results on DFDC dataset, as summarized in Table 8. Previous approaches achieved accuracy values ranging from 91.5% to 98.24%. Notable models included Khan [28] (91.69%), and Thing [62] (92.02%), Selim [57] (97.2%). Some models excelled in additional metrics, such as the STDT [73] model with 97.44% accuracy, 99.1% AUC, and 98.48% F1-score. In comparison to previous approaches, our proposed model demonstrated excellent performance, achieving an accuracy of 98.5%, an AUC of 99.9%, and an F1-score of 99.1%.

Table 9 summarizes our model's accuracy and AUC on the FF++ dataset compared to with other state-of-the-art models.

Table 10 summarizes our model's accuracy on the three subsets of FF++ dataset compared to with other models.

Table 11 presents the AUC values for each dataset, showcasing the model's ability to discriminate between Real and Fake classes. To further investigate GenConViT performance, we also examined the ROC curve. Figure 3 presents

Table 8: GenConViT model accuracy and AUC on the DFDC dataset compared to with other state-of-the-art models.

Model	Accuracy	AUC	F1-Score
Image+Video Fusion [28]	91.69%	-	-
Selim EfficientNet B7 [57]	97.20%	90.60%	-
CViT [68]	91.50%	91%	-
Thing [62]	92.02%	97.61%	-
Random cut-out [29]	98.24%	-	-
STDT [73]	97.44%	99.10%	98.48%
ViT with distillation [20]	-	97.80%	91.90%
Heo et. al [21]	-	97.80%	91.90%
Coccomini et. al [9]	-	95.10%	88.00%
Ours	**98.50%**	**99.90%**	**99.10%**

Table 9: GenConViT model accuracy and AUC on the FF++ dataset compared to with other state-of-the-art models.

Model	Accuracy	AUC	F1-Score
Li et. al [38]	97.73%	98.52%	-
Image+Video Fusion [28]	99.52%	99.64%	**99.28**
Random cut-out [29]	97.00%	99.28%	-
Ours	97%	**99.60%**	97.1

Table 10: GenConViT model accuracy on the three subsets of FF++ dataset compared to with other state-of-the-art models.

Model	Deepfakes	Face2Face	NeuralTextures
Li et. al [38]	99.17%	97.73%	-
Coccomini [9]	87%	-	69%
Random cut-out [29]	98.57%	98.57%	90.71%
Ours	92.27%	**98%**	**97%**

the resulting ROC curve.

Table 11: GenConViT AUC values for each dataset.

Dataset	AUC (%)
DFDC	99.9
FF++	99.6
Celeb-DF (v2)	98.1

The AUC and F1 score of GenConViT(A) and GenConViT(B) are shown in Table 12, which provides a summary of the results, allowing for a comparison between the two networks. Both networks have relatively similar results, with GenConViT(B) is slightly outperforming GenConViT(A).

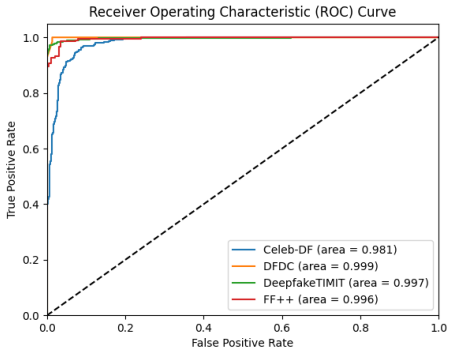


Figure 3: ROC curve illustrating the model's discrimination ability between real and fake classes.

Table 12: AUC values for GenConViT(A) and GenConViT(B) in each dataset.

Dataset	GenConViT(A)	GenConViT(B)
DFDC	99.9	99.9
FF++	99.1	99.6
Celeb-DF (v2)	97.8	94.1

Table 13: F1 scores for GenConViT(A) and GenConViT(B) in each dataset.

Dataset	GenConViT(A)	GenConViT(B)
DFDC	0.984	0.984
FF++	0.949	0.968
Celeb-DF (v2)	0.952	0.890

Additionally, our model achieved F1 scores of 99.1%, 97.1%, 95.5%, and 91.6% for the DFDC, FF++, DeepfakeTIMIT, and Celeb-DF (v2) datasets, respectively. Overall, the proposed GenConViT model has an average accuracy of 95.8% and an AUC value of 99.3% across the tested datasets. These results highlight our model's robust performance in detecting deepfake videos, demonstrating its potential for practical applications in the field.

5. Conclusion

In this work, we proposed a Generative Convolutional Vision Transformer (GenConViT) that extracts visual artifacts and latent data distributions to detect deepfake videos. GenConViT combines ConvNext and the Swin transformer models to learn from local and global image features of a video, as well as AE and VAE to learn from internal data representation. Our approach aims to fill the gap in the generalizability of deepfake detection and provides an effective solution for identifying a wide range of fake videos while preserving media integrity. Through extensive experiments on a diverse dataset, including DFDC, FF++, DeepfakeTIMIT, and Celeb-DF (v2), our GenConViT model demonstrated an improved and robust performance with high classification accuracy, F1 scores, and AUC values. Overall, our proposed GenConViT model offers a promising approach for accurate and reliable deepfake video detection.

Acknowledgement

This research was funded by Addis Ababa University Research Grant for the Adaptive Problem-Solving Research. Reference number RD/PY-183/2021. Grant number AR/048/2021.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. *MesoNet: a Compact Facial Video Forgery Detection Network*. in 2018 IEEE International Workshop on Information Forensics and Security (WIFS, 2018.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. 2017.

[3] D. Bank, N. Koenigstein, and R. Giryes. *Autoencoders*. arXiv, 2021.

[4] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus. 2019.

[5] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11, February 2020.

[6] W. Chen, S. L. B. Chua, S. Winkler, and S.-K. Ng. Trusted media challenge dataset and user study. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, 2022.

[7] A. Chintha, A. Rao, S. Sohrawardi, K. Bhatt, M. Wright, and R. Ptucha. Leveraging edges and optical flow on faces for deepfake detection. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*. Houston, 2020.

[8] U. A. Ciftci and I. Demir. Fakecatcher: Detection of synthetic portrait videos using biological signals. 2020.

[9] D. Coccomini, N. Messina, C. Gennaro, and F. Falchi. Combining efficientnet and vision transformers for video deepfake detection. 1323:219–229, 2022.

[10] D. Dagar and D. K. Vishwakarma. A literature review and perspectives in deepfakes: generation, detection, and applications. *International Journal of Multimedia Information Retrieval*, 11:219–289, September 2022.

[11] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis, arxiv. 2021.

[12] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (dfdc) dataset, arxiv. 2020.

[13] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer. The deepfake detection challenge (dfdc) preview dataset, arxiv. 2019.

[14] A. Geitgey. *The world's simplest facial recognition api for Python and the command line*. Github.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, arxiv. 2014.

[16] Diego Gragnaniello, Francesco Marra, and Luisa Verdoliva. *Detection of AI-Generated Synthetic Faces*, pages 191–212. Springer International Publishing, Cham, 2022.

[17] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, F. Huang, and L. Ma. Spatiotemporal inconsistency learning for deepfake video detection, arxiv. 2021.

[18] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization, arxiv. 2022.

[19] R. Haridas and R. L. Jyothi. Convolutional neural networks: A comprehensive survey. *International Journal of Applied Engineering Research*, 14, February 2019.

[20] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim. Deepfake detection scheme based on vision transformer and distillation, arxiv. 2021.

[21] Y.-J. Heo, W.-H. Yeo, and B.-G. Kim. Deepfake detection algorithm based on improved vision transformer. *Applied Intelligence*, 53:7512–7527, April 2023.

[22] G.-S. Hsu, C.-H. Tsai, and H.-Y. Wu. Dual-generator face reenactment. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 2022.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. 2018.

[24] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, and Y. Liu. Countering malicious deepfakes: Survey, battleground, and horizon. *International Journal of Computer Vision*, 130:1678–1734, July 2022.

[25] A. Kammoun, R. Slama, H. Tabia, T. Ouni, and M. Abid. Generative adversarial networks for face generation: A survey. *ACM Comput. Surv.*, 55, December 2022.

[26] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. 2018.

[27] T. Karras, S. Laine, and A T. Aila. A style-based generator architecture for generative adversarial networks. 2019.

[28] S. A. Khan and H. Dai. Video transformer for deepfake detection with incremental learning. In Virtual Event China, editor, *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[29] S. A. Khan and D.-T. Dang-Nguyen. Hybrid transformer network for deepfake detection, arxiv. 2022.

[30] D.-K. Kim and K. Kim. Generalized facial manipulation detection with edge region feature extraction. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, 2022.

[31] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. 2022.

[32] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12:307–392, 2019.

[33] S. Kingra, N. Aggarwal, and N. Kaur. Emergence of deepfakes and video tampering detection approaches: A survey. *Multimedia Tools and Applications*, 82:10165–10209, March 2023.

[34] P. Korshunov and S. Marcel. *DeepFakes: a New Threat to Face Recognition?* Assessment and Detection, 2018.

[35] Pavel Korshunov and Sébastien Marcel. *The Threat of Deepfakes to Computer and Human Visions*, pages 97–115. Springer International Publishing, Cham, 2022.

[36] I. Korshunova, W. Shi, J. Dambre, and L. Theis. Fast face-swap using convolutional neural networks. 2017.

[37] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. 2020.

[38] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection, arxiv. 2020.

[39] Y. Li, M.-C. Chang, and S. Lyu. *In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking*. in 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, Hong Kong, 2018.

[40] Y. Li and S. Lyu. Exposing deepfake videos by detecting face warping artifacts.

[41] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. *Toward the Creation and Obstruction of DeepFakes*, pages 71–96. Springer International Publishing, Cham, 2022.

[42] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, 2020.

[43] P. Liu, Y. Lin, Y. He, Y. Wei, L. Zhen, J. T. Zhou, R. S. M. Goh, and J. Liu. Automated deepfake detection, arxiv. 2021.

[44] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, arxiv. 2021.

[45] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and A S. Xie. Convnet for the 2020s, arxiv. 2022.

[46] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, 53:3974–4026, February 2023.

[47] F. Matern, C. Riess, and M. Stamminger. *Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations*. in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), Waikoloa Village, HI, USA, 2019.

[48] M. Mirza and S. Osindero. Conditional generative adversarial nets, arxiv. 2014.

[49] H. H. Nguyen, N.-D. T. Tieu, H.-Q. Nguyen-Son, V. Nozick, J. Yamagishi, and I. Echizen. Modular convolutional neural network for discriminating between computer-generated images and photographic images. In *13th International Conference on Availability*, Hamburg, 2018. Reliability and Security (ARES 2018.

[50] H. H. Nguyen, J. Yamagishi, and I. Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos, arxiv. 2018.

[51] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223, October 2022.

[52] K. O'Shea and R. Nash. An introduction to convolutional neural networks, arxiv. 2015.

[53] Y. Pang, J. Lin, T. Qin, and Z. Chen. Image-to-image translation: Methods and applications. 2021.

[54] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images, arxiv. 2019.

[55] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. Recurrent convolutional strategies for face manipulation detection in videos.

[56] C. Sanderson and B. C. Lovell. *Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference*. in Advances in Biometrics, Berlin, 2009.

[57] S. Seferbekov. A prize winning solution for dfdc challenge. *0*, 6, June.

[58] J. Sun, Q. Deng, Q. Li, M. Sun, M. Ren, and Z. Sun. Anyface: Free-style text-to-face synthesis and manipulation. 2022.

[59] Y. Sun, Z. Zhang, I. Echizen, H. H. Nguyen, C. Qiu, and L. Sun. Face forgery detection based on facial region displacement trajectory series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2023.

[60] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. 2019.

[61] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos.

[62] V. L. L. Thing. Deepfake detection with deep learning: Convolutional neural networks versus transformers. 2023.

[63] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. An Introduction to Digital Face Manipulation. In Christian Rathgeb, Ruben Tolosana, Ruben Vera-Rodriguez, and Christoph Busch, editors, *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*, pages 3–26. Springer International Publishing, Cham, 2022.

[64] A. Trabelsi, M. M. Pic, and J.-L. Dugelay. Improving deepfake detection by mixing top solutions of the dfdc. *in*, 2022:30, 2022.

[65] P. N. Vasist and S. Krishnan. *Engaging with deepfakes: a meta-synthesis from the perspective of social shaping of technology theory*. Internet Research, 2022.

[66] V. Wesselkamp, K. Rieck, D. Arp, and E. Quiring. Misleading deep-fake detection with gan fingerprints, arxiv. 2022.

[67] R. Wightman and GitHub PyTorch Image Models. 2019.

[68] D. Wodajo and S. Atnafu. Deepfake video detection using convolutional vision transformer, arxiv. 2021.

[69] Z. Yan, P. Sun, Y. Lang, S. Du, S. Zhang, W. Wang, and L. Liu. Multimodal graph learning for deepfake detection, arxiv. 2023.

[70] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In Icassp Ieee, editor, *International Conference on Acoustics, Speech and Signal*, pages 2019–2019. Processing (ICASSP, 2019.

[71] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. 2022.

[72] G. P. Zachary. Digital manipulation and the future of electoral democracy in the u.s. *IEEE Transactions on Technology and Society*, 1:104–112, June 2020.

[73] D. Zhang, F. Lin, Y. Hua, P. Wang, D. Zeng, and S. Ge. Deepfake video detection with spatiotemporal dropout transformer, arxiv. 2022.

[74] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia. Learning self-consistency for deepfake detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, 2021.

[75] Y. Zhu, Q. Li, J. Wang, C. Xu, and Z. Sun. One shot face swapping on megapixels. 2022.