Deepfake Video Detection Using Convolutional Vision Transformer

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

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

Abstract

The rapid advancement of deep learning models that can generate and synthesis hyper-realistic videos known as Deepfakes and their ease of access have raised concern on possible malicious intent use. Deep learning techniques can now generate faces, swap faces between two subjects in a video, alter facial expressions, change gender, and alter facial features, to list a few. These powerful video manipulation methods have potential use in many fields. However, they also pose a looming threat to everyone if used for harmful purposes such as identity theft, phishing, and scam. In this work, we propose a Convolutional Vision Transformer for the detection of Deepfakes. The Convolutional Vision Transformer has two components: Convolutional Neural Network (CNN) and Vision Transformer (ViT). The CNN extracts learnable features while the ViT takes in the learned features as input and categorizes them using an attention mechanism. We trained our model on the DeepFake Detection Challenge Dataset (DFDC) and have achieved 91.5 percent accuracy, an AUC value of 0.91, and a loss value of 0.32. Our contribution is that we have added a CNN module to the ViT architecture and have achieved a competitive result on the DFDC dataset.

1. Introduction

Technologies for altering images, videos, and audios are developing rapidly [12, 62]. Techniques and technical expertise to create and manipulate digital content are also easily accessible. Currently, it is possible to seamlessly generate hyper-realistic digital images [28] with a little resource and an easy how-to-do instructions available online [30, 9]. Deepfake is a technique which aims to replace the face of a targeted person by the face of someone else in a video [1]. It is created by splicing synthesized face region into the original image [62]. The term can also mean to represent the final output of a hype-realistic video created. Deepfakes can be used for creation of hyper-realistic Computer Generated Imagery (CGI), Virtual Reality (VR) [7], Augmented Reality (AR), Education, Animation, Arts, and Cinema [13]. However, since Deepfakes are deceptive in nature, they can also be used for malicious purposes.

Since the Deepfake phenomenon, various authors have proposed different mechanisms to differentiate real videos from fake ones. As pointed by [10], even though each proposed mechanism has its strength, current detection methods lack generalizability. The authors noted that current existing models focus on the Deepfake creation tools to tackle by studying their supposed behaviors. For instance, Yuezun *et al*. [33] and TackHyun *et al*. [25] used inconsistencies in eye blinking to detect Deepfakes. However, using the work of Konstantinos *et al*. [58] and Hai *et al*. [46], it is now possible to mimic eye blinking. The authors in [58] presented a system that generates videos of talking heads with natural facial expressions such as eye blinking. The authors in [46] proposed a model that can generate facial expression from a portrait. Their system can synthesis a still picture to express emotions, including a hallucination of eye-blinking motions.

We base our work on two weaknesses of Deepfake detection methods pointed out by [10, 11]: data preprocessing, and generality. Polychronis *et al*. [11] noted that current Deepfake detection systems focus mostly on presenting their proposed architecture, and give less emphasis on data preprocessing and its impact on the final detection model. The authors stressed the importance of data preprocessing for Deepfake detections. Joshual *et al*. [10] focused on the generality of facial forgery detection and found that most proposed systems lacked generality. The authors defined generality as reliably detecting multiple spoofing techniques and reliably spoofing unseen detection techniques.

Umur *et al*. [13] proposed a generalized Deepfake detector called FakeCatcher using biological signals (internal representations of image generators and synthesizers). They used a simple Convolutional Neural Network (CNN) classifier with only three layers. The authors used 3000 videos for training and testing. However, they didn't specify in detail how they preprocessed their data. From [31, 52, 21], it is evident that very deep CNNs have superior performance than shallow CNNs in image classification tasks. Hence, there is still room for another generalized Deepfake detector that has extensive data preprocessing pipeline and also is trained on a very deep Neural Network model to catch as many Deepfake artifacts as possible.

Therefore, we propose a generalized Convolutional Vision Transformer (CViT) architecture to detect Deepfake videos using Convolutional Neural Networks and the Transformer architecture. We call our approach generalized for three main reasons. 1) Our proposed model can learn local and global image features using the CNN and the Transformer architecture by using the attention mechanism of the Transformer [6]. 2) We give equal emphasis on our data preprocessing during training and classification. 3) We propose to train our model on a diverse set of face images using the largest dataset currently available to detect Deepfakes created in different settings, environments, and orientations.

2. Related Work

With the rapid advancement of the CNNs [4, 20], Generative Adversarial Networks (GANs) [18], and its variants [22], it is now possible to create hyper-realistic images [32], videos [61] and audio signals [53, 15] that are much harder to detect and distinguish from real untampered audiovisuals. The ability to create a seemingly real sound, images, and videos have caused a steer from various concerned stakeholders to deter such developments not to be used by adversaries for malicious purposes [12]. To this effect, there is currently an urge in the research community to come with Deepfake detection mechanisms.

2.1. Deep Learning Techniques for Deepfake Video Generation

Deepfake is generated and synthesized by deep generative models such GANs and Autoencoders (AEs) [18, 37]. Deepfake is created by swapping between two identities of subjects in an image or video [56]. Deepfake can also be created by using different techniques such as face swap [43], puppet-master [53], lip-sync [49, 47], face-reenactment [14], synthetic image or video generation, and speech synthesis [48]. Supervised [45, 24, 51], and unsupervised image-to-image translation [19] and video-to-video translation [59, 35] can be used to create highly realistic Deepfakes.

The first Deepfake technique is the FakeAPP [42] which used two AE network. An AE is a Feedforward Neural Network (FFNN) with an encoder-decoder architecture that is trained to reconstruct its input data [60]. FakeApp's encoder extracts the latent face features, and its decoder reconstructs the face images. The two AE networks share the same encoder to swap between the source and target faces, and different decoders for training.

Most of the Deepfake creation mechanisms focus on the face region in which face swapping and pixel-wise editing are commonly used [28]. In the face swap, the face of a source image is swapped on the face of a target image. In puppet-master, the person creating the video controls the person in the video. In lip-sync, the source person controls the mouse movement in the target video, and in face reenactment, facial features are manipulated [56]. The Deepfake creation mechanisms commonly use feature map representations of a source image and target image. Some of the feature map representations are the Facial Action Coding System (FACS), image segmentation, facial landmarks, and facial boundaries [37]. FACS is a taxonomy of human facial expression that defines 32 atomic facial muscle actions named Action Units (AU) and 14 Action Descriptors (AD) for miscellaneous actions. Facial land marks are a set of defined positions on the face, such as eye, nose, and mouth positions [36].

2.1.1 Face Synthesis

Image synthesis deals with generating unseen images from sample training examples [23]. Face image synthesis techniques are used in face aging, face frontalization, and pose guided generation. GANs are used mainly in face synthesis. GANs are generative models that are designed to create generative models of data from samples [3, 18]. GANs contain two adversarial networks, a generative model G , and discriminative model D . The generator and the discriminator act as adversaries with respect to each other to produce real-like samples [22]. The generator's goal is to capture the data distribution. The goal of the discriminator is to determine whether a sample is from the model distribution or the data distribution [18]. Face frontalization GANs change the face orientation in an image. Pose guided face image generation maps the pose of an input image to another image. GAN architecture, such as StyleGAN [26] and FSGAN [43], synthesize highly realistic-looking images.

2.1.2 Face Swap

Face swap or identity swap is a GAN based method that creates realistic Deepfake videos. The face swap process inserts the face of a source image in a target image of which the subject has never appeared [56]. It is most popularly used to insert famous actors in a variety of movie clips [2]. Face swaps can be synthesized using GANs and traditional CV techniques such as FaceSwap (an application for swapping faces) and ZAO (a Chines mobile application that swaps anyone's face onto any video clips) [56]. Face Swapping GAN (FSGAN) [43], and Region-Separative GAN (RSGAN) [39] are used for face swapping, face reenactment, attribute editing, and face part synthesis. Deepfake FaceSwap uses two AEs with a shared encoder that reconstructs training images of the source and target faces [56]. The processes involve a face detector that crops and aligns the face using facial landmark information [38]. A trained encoder and decoder of the source face swap the features of the source image to the target face. The autoencoder output is then blended with the rest of the image using Poisson editing [38].

Facial expression (face reenactment) swap alters one's facial expression or transforms facial expressions among persons. Expression reenactment turns an identity into a puppet [37]. Using facial expression swap, one can transfer the expression of a person to another one [27]. Various facial reenactments have proposed through the years. CycleGAN is proposed by Jun-Yan *et al.* [63] for facial reenactment between two video sources without any pair of training examples. Face2Face manipulates the facial expression of a source image and projects onto another target face in real-time [54]. Face2Face creates a dense reconstruction between the source image and the target image that is used for the synthesis of the face images under different light settings [38].

2.2. Deep Learning Techniques for Deepfake Video Detection

Deepfake detection methods fall into three categories [34, 37]. Methods in the first category focus on the physical or psychological behavior of the videos, such as tracking eye blinking or head pose movement. The second category focus on GAN fingerprint and biological signals found in images, such as blood flow that can be detected in an image. The third category focus on visual artifacts. Methods that focus on visual artifacts are data-driven, and require a large amount of data for training. Our proposed model falls into the third category. In this section, we will discuss various architectures designed and developed to detect visual artifacts of Deepfakes.

Darius *et al.* [1] proposed a CNN model called MesoNet network to automatically detect hyper-realistic forged videos created using Deepfake [40] and Face2Face [54]. The authors used two network architectures (Meso-4 and MesoInception-4) that focus on the mesoscopic properties of an image. Yuezun and Siwei [34] proposed a CNN architecture that takes advantage of the image transform (i.e., scaling, rotation and shearing) inconsistencies created during the creation of Deepfakes. Their approach targets the artifacts in affine face warping as the distinctive feature to distinguish real and fake images. Their method compares the Deepfake face region with that of the neighboring pixels to spot resolution inconsistencies that occur during face warping.

Huy *et al.* [41] proposed a novel deep learning approach to detect forged images and videos. The authors focused on replay attacks, face swapping, facial reenactments and fully computer generated image spoofing. Daniel Mas Montserrat *et al.* [38] proposed a system that extracts visual and temporal features from faces present in a video. Their method combines a CNN and RNN architecture to detect Deepfake videos.

Md. Shohel Rana and Andrew H. Sung [50] proposed a DeepfakeStack, an ensemble method (A stack of different DL models) for Deepfake detection. The ensemble is composed of XceptionNet, InceptionV3, InceptionResNetV2, MobileNet, ResNet101, DenseNet121, and DenseNet169 open source DL models. Junyaup Kim *et al.* [29] proposed a classifier that distinguishes target individuals from a set of similar people using ShallowNet, VGG-16, and Xception pre-trained DL models. The main objective of their system is to evaluate the classification performance of the three DL models.

3. Convolutional Vision Transformer

In this section, we present our approach to detect Deepfake videos. The Deepfake video detection model consists of two components: the preprocessing component and the detection component. The preprocessing component consists of the face extraction and data augmentation. The detection components consist of the training component, the validation component, and the testing component. The training and validation components contain a Convolutional Vision Transformer (CViT). The CViT has a feature learning component that learns the features of input images and a ViT architecture that determines whether a specific video is fake or real. The testing component applies the CViT learning model on input images to detect Deepfakes. Our proposed model is shown in Figure 1.

3.1. Preprocessing

The preprocessing component's function is to prepare the raw dataset for training, validating, and testing our CViT model. The preprocessing component has two subcomponents: the face extraction, and the data augmentation component. The face extraction component is responsible for extracting face images from a video in a *224* x *224* RGB format. Figure 2 and Figure 3 shows a sample of the extracted faces.

3.2. Detection

The Deepfake detection process consists of three subcomponents: the training, the validation, and the testing components. The training component is the principal part of the proposed model. It is where the learning occurs. DL models require a significant time to design and fine-tune to fit a particular problem domain into its model. In our case, the foremost consideration is to search for an optimal CViT model that learns the features of Deepfake videos. For this, we need to search for the right parameters appropriate for training our dataset. The validation component is similar to that of the training component. The validation component is a process that fine-tunes our model. It is used to

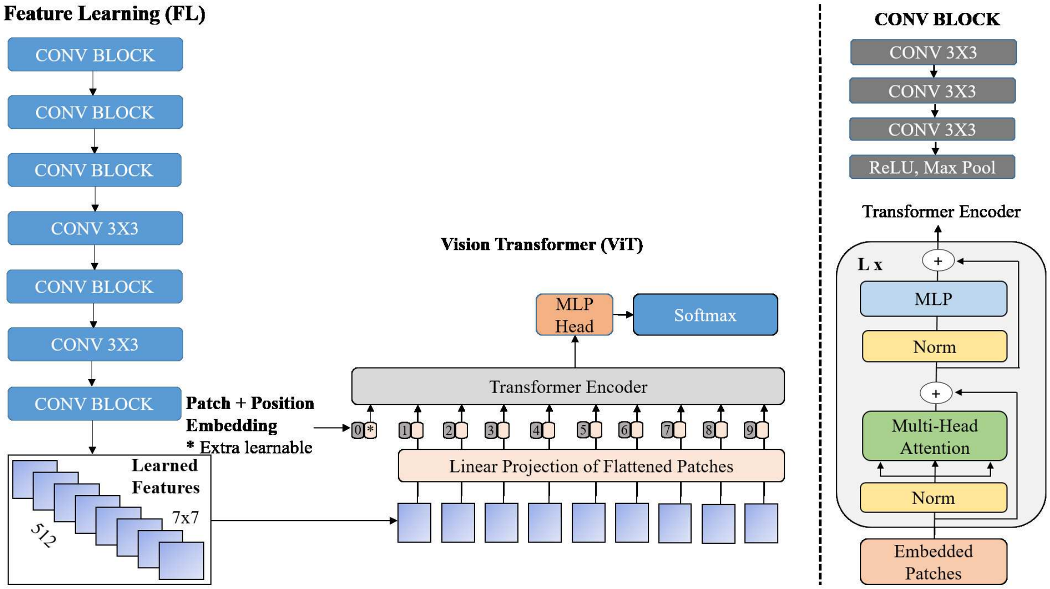

Figure 1. Convolutional Vision Transformer.

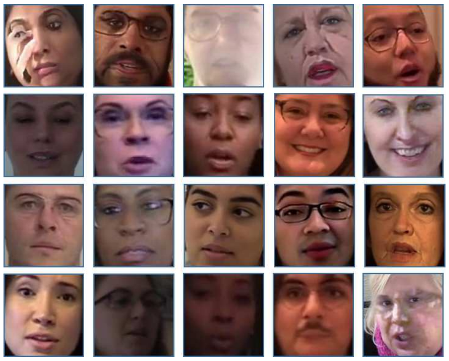

Figure 2. Sample extracted fake face images.

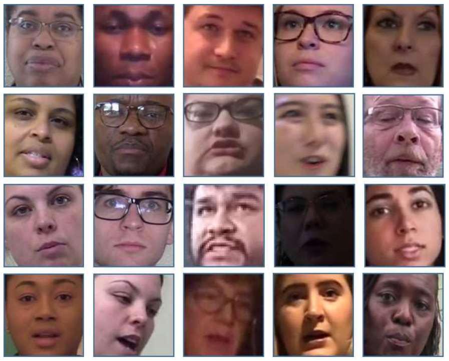

Figure 3. Sample extracted real face images.

evaluate our CViT model and helps the CViT model to update its internal state. It helps us to track our CViT model's training progress and its Deepfake detection accuracy. The testing component is where we classify and determine the class of the faces extracted in a specific video. Thus, this sub-component addresses our research objectives.

The proposed CViT model consists of two components: Feature Learning (FL) and the ViT. The FL extracts learnable features from the face images. The ViT takes in the FL as input and turns them into a sequence of image pixels for the final detection process.

The Feature Learning (FL) component is a stack of convolutional operations. The FL component follows the structure of VGG architecture [52]. The FL component differs from the VGG model in that it doesn't have the fully connected layer as in the VGG architecture, and its purpose is not for classification but to extract face image features for the ViT component. Hence, the FL component is a CNN without the fully connected layer.

The FL component has 17 convolutional layers, with a kernel of 3 x 3. The convolutional layers extract the low level feature of the face images. All convolutional layers have a stride and padding of 1. Batch normalization to normalize the output features and the ReLU activation function for non-linearity are applied in all of the layers. The Batch normalization function normalizes change in the distribution of the previous layers [41], as the change in between the layers will affect the learning process of the CNN architecture. A five max-pooling of a 2 x 2-pixel window with stride equal to 2 is also used. The max-pooling operation reduces dimension of image size by half. After each max-pooling operation, the width of the convolutional layer (channel) is doubled by a factor of 2, with the first layer having 32 channels and the last layer 512.

The FL component has three consecutive convolutional operations at each layer, except for the last two layers, which have four convolutional operations. We call those three convolutional layers as CONV Block for simplicity. Each convolutional computation is followed by batch normalization and the ReLU nonlinearity. The FL component has 10.8 million learnable parameters. The FL takes in an image of size 224 x 224 x 3, which is then convolved at each convolutional operation. The FL internal state can be represented as (C, H, W) tensor, where C is the channel, H is the height, and W is the width. The final output of the FL is a 512 x 7 x 7 spatially correlated low level feature of the input images, which are then fed to the ViT architecture.

Our Vision Transformer (ViT) component is identical to the ViT architecture described in [16]. Vision Transformer (ViT) is a transformer model based on the work of [57]. The transformer and its variants (e.g., GPT-3 [44]) are predominantly used for NLP tasks. ViT extends the application of the transformer from the NLP problem domain to a CV problem domain. The ViT uses the same components as the original transformer model with slight modification of the input signal. The FL component and the ViT component makes up our Convolutional Vision Transformer (CViT) model. We named our model CViT since the model is based on both a stack of convolutional operation and the ViT architecture.

The input to the ViT component is a feature map of the face images. The feature maps are split into seven patches and are then embedded into a 1 x 1024 linear sequence. The embedded patches are then added to the position embedding to retain the positional information of the image feature maps. The position embedding has a 2 x 1024 dimension.

The ViT component takes in the position embedding and the patch embedding and passes them to the Transformer. The ViT Transformer uses only an encoder, unlike the original Transformer. The ViT encoder consists of MSA and MLP blocks. The MLP block is an FFN. The Norm normalizes the internal layer of the transformer. The Transformer has 8 attention heads. The MLP head has two linear layers and the ReLU nonlinearity. The MLP head task is equivalent to the fully connected layer of a typical CNN architecture. The first layer has 2048 channels, and the last layer has two channels that represent the class of Fake or Real face image. The CViT model has a total of 20 weighted layers and 38.6 million learnable parameters. Softmax is applied on the MLP head output to squash the weight values between 0 and 1 for the final detection purpose.

4. Experiments

In this section, we present the tools and experimental setup we used to design and develop the prototype to implement the model. We will present the results acquired from the implementation of the model and give an interpretation of the experimental results.

4.1. Dataset

DL models learn from data. As such, careful dataset preparation is crucial for their learning quality and prediction accuracy. BlazeFace neural face detector [5], MTCNN [55] and face_recognition [17] DL libraries are used to extract the faces. Both BlazeFace and face_recognition are fast at processing a large number of images. The three DL libraries are used together for added accuracy of face detection. The face images are stored in a JPEG file format with 224 x 224 image resolution. A 90 percent compression ratio is also applied. We prepared our datasets in a train, validation, and test sets. We used 162,174 images classified into 112,378 for training, 24,898 for validation and 24,898 for testing with $70{:}15{:}15$ ratios, respectively. Each real and fake class has the same number of images in all sets.

We used Albumentations for data augmentation. Albumentations is a python data augmentation library which has a large class of image transformations. Ninety percent of the face images were augmented, making our total dataset to be 308,130 facial images.

4.2. Evaluation

The CViT model is trained using the binary cross-entropy loss function. A mini-batch of 32 images are normalized using mean of $[0.485,\ 0.456,\ 0.406]$ and standard deviation of $[0.229,\ 0.224,\ 0.225]$. The normalized face images are then augmented before being fed into the CViT model at each training iterations. Adam optimizer with a learning rate of $0.1e\text{-}3$ and weight decay of $0.1e\text{-}6$ is used for optimization. The model is trained for a total of 50 epochs. The learning rate decreases by a factor of 0.1 at each step size of 15.

The classification process takes in 30 facial images and passes it to our trained model. To determine the classification accuracy of our model, we used a log loss function. A log loss described in Equation 1 classifies the network into a probability distribution from 0 to 1, where $0 > y < 0.5$ represents the real class, and $0.5 \geq y < 1$ represents the fake class. We chose a log loss classification metric because it highly penalizes random guesses and confident false predictions.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + log(1 - y_i) log(1 - \hat{y}_i)] \tag{1}$$

Another metric we used to measure our model capacity is the ROC and AUC metrics [8]. The ROC is used to visualize a classifier to select the classification threshold. AUC is an area covered by the ROC curve. AUC measures the accuracy of a classifier.

We present our result using accuracy, AUC score, and loss value. We tested the model on 400 unseen DFDC videos and achieved 91.5 percent accuracy, an AUC value of 0.91, and a loss value of 0.32. The loss value indicates how far our models' prediction is from the actual target value. For Deepfake detection, we used 30 face images from each video. The amount of frame number we use affects the chance of Deepfake detection. However, accuracy might not always be the right measure to detect Deepfakes as we might encounter all real facial images from a fake video (fake videos might contain real frames).

We compared our result with other Deepfake detection models, as shown in Table 1, 2, and 3. From Table 1, 2, and 3, we can see that our model performed well on the DFDC, UADFV, and FaceForensics++ dataset. However, our model performed poorly on the FaceForensics++ FaceShifter dataset. The reason for this is because visual artifacts are hard to learn, and our proposed model likely didn't learn those artifacts well.

Dataset	Accuracy
FaceForensics++ FaceSwap	69%
FaceForensics++ DeepFakeDetection	91%
FaceForensics++ Deepfake	93%
FaceForensics++ FaceShifter	46%
FaceForensics++ NeuralTextures	60%

Table 1. CViT model prediction accuracy on FaceForensics++ dataset

Method	Validation	Test
CNN and RNN-GRU [38] [47]	92.61%	91.88%
CViT	87.25	91.5

Table 2. Accuracy of our model and other Deepfake detection models on the DFDC dataset

Method	Validation	FaceSwap	Face2Face
MesoNet	84.3%	96%	92%
MesoInception	82.4%	98%	93.33%
CViT	93.75	69%	69.39%

Table 3. AUC performance of our model and other Deepfake detection models on UADFV dataset. * FaceForensics++

4.3. Effects of Data Processing During Classification

A major potential problem that affects our model accuracy is the inherent problems that are in the face detection DL libraries (MTCNN, BlazeFace, and face_recognition). Figure 4, Figure 5 and Figure 6 show images that were misclassified by the DL libraries. The figures summarize our preliminary data preprocessing test on 200 videos selected randomly from 10 folders. We chose our test set video in all settings we can found in the DFDC dataset: indoor, outdoor, dark room, bright room, subject sited, subject standing, speaking to side, speaking in front, a subject moving while speaking, gender, skin color, one person video, two people video, a subject close to the camera, and subject away from the camera. For the preliminary test, we extracted every frame of the videos and found the 637 nonface region.

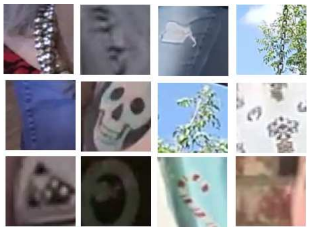

Figure 4. face_recognition non face region detection.

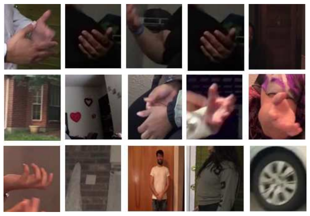

Figure 5. BlazeFace non face region detection.

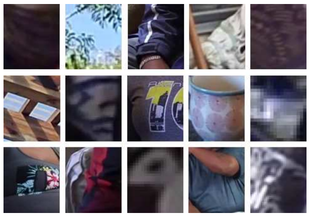

Figure 6. MTCNN non face region detection.

We tested our model to check how its accuracy is affected without any attempt to remove these images, and our models' accuracy dropped to 69.5 percent, and the loss value increased to 0.4.

To minimize non face regions and prevent wrong predictions, we used the three DL libraries and picked the best performing library for our model, as shown in Table 4. As a solution, we used face_recognition as a "filter" for the face images detected by BlazeFace. We chose face_recognition because, in our investigation, it rejects more false-positive than the other two models. We used face_recognition for final Deepfake detection.

Dataset	Blazeface	f_rec **	MTCNN
DFDC	83.40%	**91.50%**	90.25%
FaceSwap	56%	**69%**	63%
FaceShifter	40%	**46%**	44%
NeuralTextures	57%	**60%**	**60%**
DeepFakeDetection	82%	**91%**	79.59
Deepfake	87%	**93%**	81.63%
Face2Face	54%	61%	**69.39%**
UADF	74.50%	**93.75%**	88.16%

Table 4. DL libraries comparison on Deepfake detection accuracy. ** face_recognition

5. Conclusion

Deepfakes open new possibilities in digital media, VR, robotics, education, and many other fields. On another spectrum, they are technologies that can cause havoc and distrust to the general public. In light of this, we have designed and developed a generalized model for Deepfake video detection using CNNs and Transformer, which we named Convolutional Vison Transformer. We called our model a generalized model for three reasons. 1) Our first reason arises from the combined learning capacity of CNNs and Transformer. CNNs are strong at learning local features, while Transformers can learn from local and global feature maps. This combined capacity enables our model to correlate every pixel of an image and understand the relationship between nonlocal features. 2) We gave equal emphasis on our data preprocessing during training and classification. 3) We used the largest and most diverse dataset for Deepfake detection.

The CViT model was trained on a diverse collection of facial images that were extracted from the DFDC dataset. The model was tested on 400 DFDC videos and has achieved an accuracy of 91.5 percent. Still, our model has a lot of room for improvement. In the future, we intend to expand on our current work by adding other datasets released for Deepfake research to make it more diverse, accurate, and robust.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. MesoNet: a Compact Facial Video Forgery Detection Network. pages 1–7, 2018.

[2] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting World Leaders Against Deep Fakes. In *CVPR Workshops*, 2019.

[3] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, Switzerland, 2020.

[4] Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3):292, 2019.

[5] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs. *arXiv preprint arXiv:1907.05047v2*, 2019.

[6] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3285–3294, 2019.

[7] Avishek Joey Bose and Parham Aarabi. Virtual Fakes: DeepFakes for Virtual Reality. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–1. IEEE, 2019.

[8] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[9] John Brandon. Terrifying high-tech porn: Creepy 'deepfake' videos are on the rise, 2018. Available at https://www.foxnews.com/tech/terrifying-high-tech-porn-creepy-deepfake-videos-are-on-the-rise.

[10] Joshua Brockschmidt, Jiacheng Shang, and Jie Wu. On the Generality of Facial Forgery Detection. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, pages 43–47. IEEE, 2019.

[11] Polychronis Charitidis, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Ioannis Kompatsiaris. Investigating the Impact of Pre-processing and Prediction Aggregation on the DeepFake Detection Task. *arXiv preprint arXiv:2006.07084v1*, 2020.

[12] Bobby Chesney and Danielle Citron. Deep Fakes A Looming Challenge for Privacy, Democracy, and National Security, 2019. Available at https://ssrn.com/abstract=3213954.

[13] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals. *arXiv preprint arXiv:1901.02212v2*, 2019.

[14] Sourabh Dhere, Suresh B. Rathod, Sanket Aarankalle, Yash Lad, and Megh Gandhi. A Review on Face Reenactment Techniques. In *2020 International Conference on Industry 4.0 Technology (I4Tech)*, pages 191–194, Pune, India, 2020. IEEE.

[15] Chris Donahue, Julian J. McAuley, and Miller S. Puckette. Adversarial Audio Synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, New York, NY, USA, 2019. OpenReview.net.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929v1*, 2020.

[17] Adam Geitgey. The world's simplest facial recognition api for Python and the command line. Available at `https://github.com/ageitgey/face_recognition`.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairy, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[19] Arushi Handa, Prerna Garg, and Vijay Khare. Masked Neural Style Transfer using Convolutional Neural Networks. In *2018 International Conference on Recent Innovations in Electrical, Electronics Communication Engineering (ICRIEECE)*, pages 2099–2104, 2018.

[20] Rahul Haridas and Jyothi R L. Convolutional Neural Networks: A Comprehensive Survey. *International Journal of Applied Engineering Research (IJAER)*, 14(03):780–789, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[22] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How Generative Adversarial Networks and Their Variants Work: An Overview. volume 52, New York, NY, USA, 2019. Association for Computing Machinery.

[23] He Huang, Phillip S. Yu, and Changhu Wang. An Introduction to Image Synthesis with Generative Adversarial Nets. *arXiv preprints arXiv:1803.04469v1*, 2018.

[24] Xun Huang, Ming-Yu Liu, Serge Belongie, and Ming-Yu Liu. Multimodal Unsupervised Image-to-Image Translation. In *Computer Vision – ECCV 2018*, pages 179–196, Cham, 2018. Springer International Publishing.

[25] TackHyun Jung, SangWon Kim, and KeeCheon Kim. DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. *IEEE Access*, 8:83144–83154, 2020.

[26] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprints arXiv:1812.04948*, 2018.

[27] Hasam Khalid and Simon S. Woo. OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2794–2803, 2020.

[28] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, and Pankaj Wasnik. Fake face detection methods: Can they be generalized? In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2018.

[29] Junyaup Kim, Siho Han, and Simon S. Woo. Classifying Genuine Face images from Disguised Face Images. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 6248–6250, 2019.

[30] Pavel Korshunov and Sebastien Marcel. DeepFakes: a New Threat to Face Recognition? Assessment and Detection. *arXiv preprints arXiv:1812.08685*, 2018.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, 60(6):84–90, 2017.

[32] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv preprint arXiv:1609.04802v5*, 2017.

[33] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking. *arXiv preprint arXiv:1806.02877v2*, 2018.

[34] Yuezun Li and Siwei Lyu. Exposing DeepFake Videos By Detecting Face Warping Artifacts. *arXiv preprint arXiv:1811.00656v3*, 2019.

[35] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-Consistent Video-to-Video Synthesis. In *Computer Vision – ECCV 2020*, pages 359–378, Cham, 2020. Springer International Publishing.

[36] Brais Martinez, Michel F. Valstar, Bihan Jiang, and Maja Pantic. Automatic Analysis of Facial Actions: A Survey. *IEEE Transactions on Affective Computing*, 10(3):325–347, 2019.

[37] Yisroel Mirsky and Wenke Lee. The Creation and Detection of Deepfakes: A Survey. *ACM Comput. Surv.*, 54(1), 2021.

[38] Daniel Mas Montserrat, Hanxiang Hao, S. K. Yarlagadda, Sriram Baireddy, Ruiting Shao, Janos Horvath, Emily Bartusiak, Justin Yang, David Guera, Fengqing Zhu, and Edward J. Delp. Deepfakes Detection with Automatic Face Weighting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2851–2859, 2020.

[39] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. RSGAN: Face Swapping and Editing Using Face and Hair Representation in Latent Spaces. In *ACM SIGGRAPH 2018 Posters*, SIGGRAPH '18, New York, NY, USA, 2018. Association for Computing Machinery.

[40] Huy H. Nguyen, Ngoc-Dung T. Tieu, Hoang-Quoc Nguyen-Son, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Modular Convolutional Neural Network for Discriminating between Computer-Generated Images and Photographic Images. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, New York, NY, USA, 2018. Association for Computing Machinery.

[41] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311, 2019.

[42] Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep Learning for Deepfakes Creation and Detection. *arXiv preprint arXiv:1909.11573v1*, 2019.

[43] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7183–7192. IEEE, 2019.

[44] OpenAI. OpenAI API, 2020. Available at `https://openai.com/blog/openai-api`.

[45] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis With Spatially-Adaptive

Normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341. IEEE, 2019.

[46] Hai X. Pham, Yuting Wang, and Vladimir Pavlovic. Generative Adversarial Talking Head: Bringing Portraits to Life with a Weakly Supervised Neural Network. *arXiv preprint arXiv:1803.07716*, 2018.

[47] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C V Jawahar. *A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild*, page 484–492. Association for Computing Machinery, New York, NY, USA, 2020.

[48] Mike Price and Matt Price. Playing Offense and Defense with Deepfakes, 2019. Available at lhttps://www.blackhat.com/us-19/briefings/schedule/playing-offense-and-defense-with-deepfakes-14661.

[49] Prajwal K R, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and C V Jawahar. Towards Automatic Face-to-Face Translation. In *the 27th ACM International Conference on Multimedia (MM '19)*, page 1428–1436, New York, NY, USA, 2019. Association for Computing Machinery.

[50] Md. Shohel Rana and Andrew H. Sung. DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection. In *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 70–75, 2020.

[51] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. COCO-FUNIT: Few-Shot Unsupervised Image Translation with a Content Conditioned Style Encoder. In *Computer Vision – ECCV 2020*, pages 382–398, Cham, 2020. Springer International Publishing.

[52] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[53] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: Learning Lip Sync from Audio. *ACM Trans. Graph.*, 36(4):780–789, 2017.

[54] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. *Commun. ACM*, 62(1):96–104, 2018.

[55] Timesler. Pretrained Pytorch face detection (MTCNN) and recognition (InceptionResnet) models. Available at `https://github.com/timesler/facenet-pytorch`.

[56] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. *Inf. Fusion*, 64:131–148, 2020.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010. Curran Associates Inc., 2017.

[58] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic Speech-Driven Facial Animation with GANs. *International Journal of Computer Vision*, 128:1398–1413, 2020.

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1152–1164, Red Hook, NY, USA, 2018. Curran Associates Inc.

[60] M. Arif Wani, Farooq Ahmad Bhat, Saduf Afzal, and Asif Iqbal Khan. *Advances in Deep Learning*, volume 57 of *Studies in Big Data*. Springer Nature, Singapore, 2020.

[61] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-Shot Adversarial Learning of Realistic Neural Talking Head Models. *arXiv preprint arXiv:1905.08233v2*, 2019.

[62] Lilei Zheng, Ying Zhang, and Vrizlynn L.L. Thing. A survey on image tampering and its detection in real-world photos. *Elsevier*, 58:380–399, 2018.

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.