Improved Deepfake Video Detection Using Convolutional Vision Transformer

Deressa Wodajo Deressa*, Peter Lambert**, Glenn Van Wallendael**, Solomon Atnafu†, Hannes Mareen**
**Ghent University – imec, IDLab, Department of Electronics and Information Systems, Gent, Belgium
firstname.lastname@ugent.be, https://media.idlab.ugent.be
*deressawodajo.deressa@ugent.be
†Addis Ababa University, solomon.atnafu@aau.edu.et

***Abstract*—Deepfakes are hyper-realistic videos in which the faces are replaced, swapped, or forged using deep-learning models. This potent media manipulation techniques hold promise for applications across various domains. Yet, they also present a significant risk when employed for malicious intents like identity fraud, phishing, spreading false information, and executing scams. In this work, we propose a novel and improved Deepfake video detector that uses a Convolutional Vision Transformer (CViT2), which builds on the concepts of our previous work (CViT). The CViT architecture consists of two components: a Convolutional Neural Network that extracts learnable features, and a Vision Transformer that categorizes these learned features using an attention mechanism. We trained and evaluted our model on 5 datasets, namely Deepfake Detection Challenge Dataset (DFDC), FaceForensics++ (FF++), Celeb-DF v2, DeepfakeTIMIT, and TrustedMedia. On the test sets unseen during training, we achieved an accuracy of 95%, 94.8%, 98.3% and 76.7% on the DFDC, FF++, Celeb-DF v2, and TIMIT datasets, respectively. In conclusion, our proposed Deepfake detector can be used in the battle against misinformation and other forensic use cases.**

***Index Terms*—Deepfake Video Detection, Vision Transformer, Convolutional Neural Network, Misinformation Detection, Multimedia Forensics**

I. Introduction

In Deepfake videos, the face of a targeted person is manipulated, replaced by the face of someone else (1). It is created by splicing a synthesized face into the original video, using Deep Learning (DL) (2). The term Deepfake is also used to represent the final output of generated hype-realistic videos.

Deepfakes can be used in a wide range of areas such as Education, Arts, Cinema, Computer Generated Imagery, Virtual and Augmented Reality (3; 4). However, since Deepfake videos are difficult to distinguish from real videos, they are increasingly being used for malicious purposes. Hence, distinguishing between authentic and altered videos is crucial to prevent and reduce the danger associated with Deepfakes.

Although various DL models have been proposed to differentiate real videos from Deepfake ones, they still struggle to accurately detect Deepfakes. Reliably detecting multiple Deepfake techniques, including those that are unseen, is defined as generality or generalization. This is considered one of the main challenges in Deepfake or forgery detection (5; 6; 7).

Current Deepfake detection systems focus mostly on presenting their architecture, whereas they give less emphasis on data preprocessing and its impact on the final detection model (8). In practice, data preprocessing during classification is key in accurate Deepfake detection. Hence, we believe that this aspect should be discussed and analyzed more in depth.

This paper proposes an Improved Convolutional Vision Transformer (CViT2) architecture to detect Deepfake videos. Our architecture consists of Convolutional Neural Networks (CNNs) and the Transformer architecture. We present our data preprocessing pipeline in depth, and train our network on multiple datasets in order to provide more generality.

Our contributions can be summarized as follows:

1) Efficient Feature Extraction: CViT2 combines CNNs and Transformers with attention mechanisms to learn local and global image features with significant detail, crucial for Deepfake detection. This enables CViT2 to capture subtle details across various image regions.
2) During classification, we incorporate data preprocessing methods, such as cropping, and padding. This demonstrates the importance of data preprocessing during classification.
3) To boost generality, we trained on a dataset exceeding 1 million images extracted from five distinct sources. Our dataset encompasses Deepfakes from various environment settings, light conditions, face orientations, and skin color for robust Deepfake detection.
4) To facilitate further research and collaboration, we have open-sourced our model and code. [1]

Note that this work is an improved version (CViT2) of our previous work (CViT) (9), which was trained on 135,000 images from the DFDC dataset. In CViT2, we use 1 million face images extracted from five diverse datasets (DFDC (10), TrustedMedia (TM) (11), DeepfakeTIMIT (TIMIT) (12; 13), Celeb-DF v2 (14), and FaceForensics++ (FF++) (15)) and achieved approx. 3% accuracy gain without modifying the model architecture.

This research was supported by Jimma University, Addis Ababa University, the Research Foundation – Flanders (FWO under project grant G0A2523N), the Flemish government (COM-PRESS project, within the relanceplan Vlaamse Veerkracht), IDLab (Ghent University – imec), Flanders Innovation and Entrep-reneurship (VLAIO), and the European Union.

[1] Open-source model and code available at https://github.com/erprogs/CViT

II. Related Work

This section presents related Deepfake creation and detection methods.

Majority of Deepfake generation techniques concentrate on the facial area, where face swapping and pixel-level editing are frequently employed (16). For example, in the faceswap method, the face of a source image is swapped on the face of a target image (17). In puppet-master, the person creating the video controls the person in the video (18). In lip-sync, the facial movements of the source person dictate the mouse movements in the target video (19; 20), and in face reenactment, facial features are manipulated (21; 22). Deepfake generation methods typically utilize feature map representations from both source and target images. Among these representations are the Facail Action Coding System (FCAS), image segmentation, facial landmarks, and facial boundaries. (23). FACS categorizes human facial expression, outlining 32 fundamental facial muscle movements called Action Units (AU) and 14 Action Descriptors (AD) for various actions. Facial landmarks refer to specific points on the face, like the locations of the eyes, nose, and mouth. (24).

Deepfake detection techniques can be classified into three types (23; 25). Methods in the first category concentrate on the physical or psychological behaviours of the videos, like monitoring eye blinking or head movements. The second category targets the GAN fingerprints present in images.. The third category centers on visual artifacts, employing data-driven approaches that requires extensive datasets for training. Our proposed model belongs to this category. In this section, we explore various proposed models to detect visual anomalies in Deepfakes.

Current existing Deepfake detection models focus on flaws in the Deepfake creation process, such as inconsistencies in eye blinking (26; 27). However, due to the rapid advancement of Deepfake technology, these flaws are fixed fairly quickly in new Deepfake creation models. For example, Vougioukas et al. (28) presented a system that generates videos of talking heads with natural facial expressions such as eye blinking. Similarly, Pham et al. (29) proposed a model that can generate facial expression from a portrait. Their system can transform a still image to display emotions, incorporating an illusion of eye-blinking movements. As such, these detection methods struggle with new Deepfake generation models.

Darius et al. (1) proposed a CNN model called MesoNet to automatically detect hyper-realistic forged videos created using Deepfake (30) and Face2Face (31). The authors used two network architectures, Meso-4 and MesoInception-4, which target the mesoscopic attributes of an image. Their approach achieved a 98% success rate in detecting Deepfakes and a 95% accuracy in identifying Face2Face reenactments. However, MesoNet operates on a shallow CNN framework, which offers lower capacity relative to deeper CNNs, and it was trained on a limited dataset..

Yuezun and Siwei (25) proposed a CNN architecture that detects inconsistencies in image transformations (such as scaling, rotation, and shearing) generated during the creation of Deepfakes. Their method focuses on artifacts from affine face warping as the key element to differentiate between genuine and fake images. The approach involves comparing adjacent pixels to identify resolution discrepancies arising from face warping. However, current Deepfake generation techniques such as NeuralTexture can directly transfer textures of a source image to a target image without having resolution inconsistency problems.

Montserrat et al. (32) proposed a system that extracts visual and temporal features from faces in a video. Their method integrates a CNN with a Recurrent Neural Network (RNN) architecture and introduced an Automatic Frame Weighting mechanism to address common detection challenges. Their method assigns weights to video frames using logit metrics from a pre-trained EfficientNet-b5 model, improving accuracy by considering both visual and temporal features. This approach, tested on the DFDC dataset, achieved a 92.61% accuracy rate. Despite its advancements, the AFW mechanism does not fully solve the inaccuracies inherent in face detection algorithms used to detect Deepfake videos.

Kim et al. (33) proposed a model that differentiates target individuals from a group of similar individuals using ShallowNet, VGG-16, and Xception pre-trained DL models. The goal of their model is to assess the classification capabilities of these three DL models. The authors conducted tests with three different DL pre-trained models, each demonstrating a low performance rate with 62% accuracy. This indicates that current image classifiers struggle to effectively classify and detect Deepfakes.

Ciftci et al. (4) proposed a generalized Deepfake detector named FakeCatcher that utilizes biological signals. They implemented a shallow Convolutional Neural Network (CNN) consisting of just three layers. Although their method is effective in detecting Deepfakes, it has been shown that (very) deep CNNs outperform shallow CNNs in image classification tasks. (34; 35).

In general, the ongoing progress in the development of Deepfake creation tools has made the detection of Deepfake videos increasingly difficult. Hence, there is still an urgent need for robust Deepfake detectors that have extensive data processing pipelines and are trained on very large datasets to catch as many Deepfake artifacts as possible.

III. Proposed Method

In this section, we present our approach to detect Deepfake videos. First, we present the datasets in Section III-A. Then, we describe how we preprocessed these datasets to fix their issues, in Section III-B. Finally, the actual Deepfake detection component is discussed in Section III-D.

A. Datasets

In our work, we employed five distinct datasets for training and validation (DFDC, FF++, Celeb-v2, TIMIT, and Trusted-Media), of which an unseen subset of the first four were used for testing.

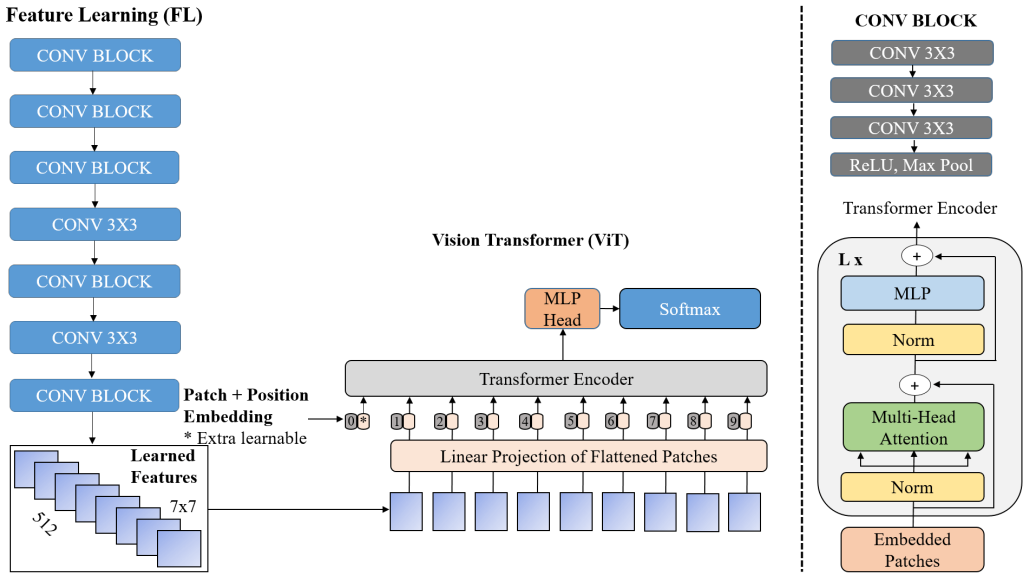


Fig. 1: Our proposed CViT2 model.

The DFDC dataset contains over 100,000 real and fake videos, compiled using 3,426 volunteers. These videos were recorded in a variety of natural settings, from different angles, and under diverse lighting conditions. The dataset was produced using eight different Deepfake generation methods.

The FF++ dataset contains 1,000 original videos sourced from YouTube, which have undergone alterations through four different automated face manipulation techniques: Deepfakes, Face2Face, FaceSwap, and NeuralTextures. It features compression in two schemes, c23 and c40, alongside a range of video resolutions.

The Celeb-DF v2 dataset includes 890 real and 5,639 Deepfake videos. All of these datasets were employed in the training, validation, and testing phases of our model.

The DeepfakeTIMIT dataset comprises 620 videos, derived by swapping faces in the VidTIMIT database using GAN techniques. Two resolutions were used: low-quality (64×64) and high-quality (128×128), resulting in two sets of 320 videos per resolution.

The TrustedMedia dataset comprises of 4,380 fabricated and 2,563 authentic videos, employing a variety of video and audio manipulation methods and is only used during the training phase.

B. Data Preprocessing

The preprocessing step is a series of image processing steps that helps us prepare and refine our dataset, focusing on the facial region. We used various preprocessing techniques for preparing the datasets.

First, in the DFDC and TM datasets, each real video was used to create one or more Deepfake videos at ratios of 6:1 and 2:1 frames, respectively. To address this imbalance, we increased the number of frames extracted from the real videos and reduced the number of frames derived from the fake videos. Second, not all of the frames in the Deepfake videos are fake. For instance, the DFDC dataset contains unmanipulated frames within its Deepfake class. Those frames needed to be removed from the training dataset because they would affect the learning processes of our proposed model. Therefore, we did a manual inspection to verify extracted face region images quality and its class, and deleted those we thought did not meet our requirement (e.g, too dark images, anomaly face detection, non-face regions, "real" face regions within a given Deepfake video, to name a few).

Finally, we apply face extraction. The face extraction component extracts face images from a video in a 224×224 RGB format. We experimented with the OpenCV, BlazeFace, and face_recognition libraries, and selected face_recognition (more evaluation details can be found in Section IV-C). We extracted a total of 1,004,810 images from four Deepfake datasets: DFDC (686,858; 68.36%), FaceForensics (176,814; 17.59%), TM (100,168; 9.97%), Celeb-DF (40,970; 4.07%).

C. Ethical Considerations

The Deepfake datasets used are all open-source and have been released for research under fair use conditions. Furthermore, we have been granted the required access by each dataset's publishers.

D. Deepfake Video Detection using CViT2

The proposed Convolutional Vision Transformer model consists of two components: Feature Learning (FL) and the Visual Transformer (ViT), as shown in Fig. 1. First, the FL extracts learnable features from the face images. Then, the ViT takes these learned features as input and transforms them into a sequence of feature embeddings for the final detection process.

1) Feature Learning Component: The FL component's task is to transform input face images into features. Our FL component is a CNN that is based on the VGG architecture

(36). Our FL component differs from the VGG model in that it does not have a fully connected layer, but rather a higher-dimension matrix of features. Additionally, its purpose is not for image classification, but rather to extract face image features that the ViT component can subsequently use.

The FL component consists of 17 convolutional layers, each with a kernel size of 3×3. The convolutional layers are responsible for extracting the low-level features of the face images. All layers have a stride and padding set to 1. Batch normalization, to normalize changes in the distribution from preceding layers, and the ReLU activation function, to introduce non-linearity, are applied across all layers. A max-pooling of a 2×2-pixel window with stride equal to 2 is used at each 5 layers. The max-pooling operation halves the dimension of the image size. After each max-pooling operation, the width of the convolutional layer (channel) is doubled by a factor of 2, starting with 32 channels in the first layer and 512 channels in the final layer.

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 a 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 $224 \times 224 \times 3$, which is then processed through 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 are spatially correlated low-level features with a resolution of $512 \times 7 \times 7$, which are then fed to the ViT architecture.

2) Vision Transformer Component: The ViT's task is to transform low-level features into a decision whether the face image is real or Deepfake.

While the 1-Dimensional (1D) transformer and its variants (e.g., GPT-3 (37)) are predominantly used for Natural Language Processing (NLP) tasks, ViT extends their application to a 2-Dimensional (2D) computer-vision problem. The 2D ViT uses the same components as the original 1D transformer model, albeit with slight modifications of the input signal. The architecture of our Vision Transformer (ViT) component is identical to the ViT architecture described by Dosovitskiy et al. (38), which uses the attention mechanism (39).

The input to the ViT component is the output of the FL component, i.e., a feature map of the face images. The feature maps are then embedded into a 2×1024 linear sequence. The embedded patches are added to the position embedding 2×1024 to retain the positional information of the image feature maps.

The ViT component takes in the concatenated patch and position embedding, and passes them to the Transformer Encoder. The ViT Transformer uses only an encoder, unlike the original Transformer which has both an encoder and a decoder. The ViT encoder consists of Multi-head Self Attention (MSA) and (Multi-Layer Perceptron) 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 90.3 million learnable parameters. Softmax is applied on the MLP head output to map the weight values between 0 and 1 for the final detection purpose.

In summary, the FL component and the ViT component make up our proposed Convolutional Vision Transformer (CViT2) model. We named our model CViT since the model is based on both a stack of convolutional operations and the ViT architecture. CViT is able to transform an image internally to learnable features, and subsequently into a real/fake detection result. When combined with our data preprocessing pipeline, this results in an effective Deepfake video detector.

IV. Evaluation

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

A. Experimental Setup

For face extraction, we experimented with the BlazeFace neural face detector (40), MTCNN (41) and face_recognition (42) libraries. We evaluate these libraries in Section IV-C, and selected face_recognition in the final proposed system. The extracted face images are stored in a JPEG file format with 224×224 image resolution, with a quality factor (QF) of 90.

We used 12 augmentation techniques during training, each with varying percentage (10-50%) depending on the experimentally observed impact of the technique on the models accuracy. The augmentations we used in training are: RandomBrightnessContrast (20%), HorizontalFlip (50%), VerticalFlip (50%), IAAPerspective (20%), RandomRotate90 (20%), ShiftScaleRotate(20%), Transpose (20%), GaussianNoise (20%), CLAHE (20%), Sharpen (20%), Emboss (20%), and HueSaturationValue (20%). We used the Albumentations library for online data augmentation. To identify the optimal level of data augmentation during training, we experimented with 0 to 90% data augmentation. We observed that data augmentation ranging from 60 to 90 percent was effective for our dataset, and therefore, we opted for 90 percent augmentation level.

Using the five datasets mentioned in Section III-A, we created a total of 1,004,810 images classified into 826,756 for training, 130,948 for validation and 47,106 for testing, i.e., with an approximate 80:15:5 ratio. We balanced the datasets, such that there are an equal number of real and fake images in each set.

We tested the model on 2,669 videos prepared from the test set: Celeb-DF v2 (518), FF+ (407), TIMIT (1,344), DFDC (400). For Deepfake detection, we used 15 face images from each video.

Dataset	Accuracy (%)	AUC
DFDC	95	0.95
FF++ (all)	94.84	0.96
FF++ Face2Face	98.00	–
FF++ Deepfake	91.26	–
FF++ NeuralTextures	86.00	–
Celeb-DF v2	98.25	0.99
TIMIT	76.72	–

TABLE I: Accuracy and AUC results of our CViT2 model on all datasets.

Method	Accuracy (%)
CNN and RNN-GRU (32)	91.88
CViT (previous preprint)	91.5
CViT2 (proposed)	**95**

TABLE II: Comparison of accuracy with other Deepfake detection models on the DFDC dataset.

The CViT2 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 CViT2 model at each training iteration. For optimization, we used the Adam optimizer with a learning rate of $0.1e$-3 and weight decay of $0.1e$-6. The model is trained for a total of 50 epochs. The learning rate decreases by a factor of 0.1 at each epoch step size of 15.

As evaluation measure, we use both the Accuracy and Area Under receiver operating Curve (AUC) metrics (43). The Receiver Operating Curve (ROC) is used to visualize the trade-off between false-positive and false-negative detections of a classifier. The AUC is the area covered by the ROC curve. A higher value signifies higher performance.

We compare our proposed CViT2 method with our previous preprint work (CViT) (9), as well as with the work of Montserrat et al. that use a CNN and RNN-GRU (32). We only compare on the DFDC dataset, because those methods only reported results on that dataset.

B. Experimental Results

Table I shows the accuracy and AUC results on all evaluated datasets. From these tables, we can see that our proposed CViT2 performs very well, achieving accuracy values between 86 and 98%, for FF++ (NeuralTextures), DFDC, and Celeb-DF v2. In contrast, it shows relatively lower perfomance on the TIMIT dataset, with an accuracy of 77%.

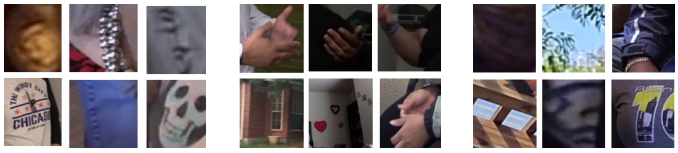
In Table II, we compare CViT2 to other Deepfake detection methods, only on the DFDC dataset. We observe that CViT2 performs approximately 3% better than CViT, and CNN and RNN-GRU (32).

C. Effects of Face Extraction Methods

An issue that can potentially affect our model's accuracy arises from inherent challenges within the deep learning libraries used for face detection. Fig. 2a, Fig. 2b, and Fig. 2c show images that were misclassified by three different DL libraries (MTCNN, BlazeFace, and face_recognition, respectively). Preliminary data preprocessing tests were conducted on 200 randomly selected videos from our test set, covering wide range of conditions, including different settings, lighting, subject positions, activities, demographic variations, and group sizes. For the preliminary test, we extracted every frame of the videos and discovered through inspection that the face extraction DL libraries mistakenly identified hundreds of non-face region as faces.

Dataset	Accuracy (%)		
	face_recognition (proposed)	Blazeface	MTCNN
DFDC	**95**	70.25	90.25
FF++ (all)	**94.84**	57.00	56
FF++ NeuralTextures	**86**	56.00	60
FF++ Deepfake	**91.25**	68.93	81.63
FF++ Face2Face	**98**	64.00	69.39
Celeb-DF v2	**98.25**	60.12	68.73
TIMIT	**76.7**	66.25	77.18

TABLE III: Accuracy when using the three different face extraction methods in our preprocessing pipeline. For our final model, we selected face_recognition as it performs significantly better.



(a) face_recognition (b) BlazeFace (c) MTCNN

Fig. 2: Examples of non-face regions that were detected as faces by the three evaluated face extraction libraries.

We tested our model to assess how its accuracy is affected when no attempt is made to remove these mistakenly identified images. Table III shows the accuracy using each of the three evaluated face extraction methods. From the table, we observe that face_recognition is better than Blazeface and MTCNN in face detection and based on our experiment, face_recognition has higher accuracy of detecting faces than both of the face detectors. Hence, we used face_recognition for our final Deepfake detection model.

V. Conclusion

We have designed and developed a model for Deepfake video detection using CNNs and the Transformer architecture, which we named Convolutional Vison Transformer. CNNs excel at learning local features, while Transformers can learn from local and global feature maps. This combined capacity enables our model to analyze every pixel of an image and understand the relationship between non-local features.

The CViT2 model was trained on a diverse set of facial images that were extracted from five different datasets. The model was tested on 2,669 videos. For example, we achieved an accuracy of 95% on the DFDC dataset. In the future, we intend to expand on our current work by training on more datasets, in order to create a more generalizable model that is diverse, accurate, and robust. In practice, the current proposed model can already be used for Deepfake detection, and as

such tackle core societal problems such as misinformation and fraud.

REFERENCES

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," in *Proc. IEEE WIFS 2018*, 2018, pp. 1–7.

[2] L. Zheng, Y. Zhang, and V. L. Thing, "A survey on image tampering and its detection in real-world photos," *Elsevier*, vol. 58, pp. 380–399, 2018.

[3] A. J. Bose and P. Aarabi, "Virtual Fakes: DeepFakes for Virtual Reality," in *Proc. IEEE MMSP 2019*. IEEE, 2019, pp. 1–1.

[4] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals," *arXiv preprint arXiv:1901.02212v2*, 2019.

[5] K. Yao, J. Wang, B. Diao, and C. Li, "Towards understanding the generalization of deepfake detectors from a game-theoretical view." IEEE, 2023.

[6] H. Mareen, L. De Neve, P. Lambert, and G. Van Wallendael, "Harmonizing image forgery detection & localization: Fusion of complementary approaches," *Journal of Imaging*, vol. 10, no. 1, 2024.

[7] J. Brockschmidt, J. Shang, and J. Wu, "On the Generality of Facial Forgery Detection," in *Proc. IEEE MASSW 2019*. IEEE, 2019, pp. 43–47.

[8] P. Charitidis, G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris, "Investigating the Impact of Preprocessing and Prediction Aggregation on the DeepFake Detection Task," *arXiv preprint arXiv:2006.07084v1*, 2020.

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

[10] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton-Ferrer, "The deepfake detection challenge dataset," *CoRR*, vol. abs/2006.07397, 2020. [Online]. Available: https://arxiv.org/abs/2006.07397

[11] W. Chen, B. Chua, and S. Winkler, "AI singapore trusted media challenge dataset," *CoRR*, vol. abs/2201.04788, 2022. [Online]. Available: https://arxiv.org/abs/2201.04788

[12] P. Korshunov and S. Marcel, "DeepFakes: a New Threat to Face Recognition? Assessment and Detection," *arXiv preprints arXiv:1812.08685*, 2018.

[13] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics*, M. Tistarelli and M. S. Nixon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 199–208.

[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," *CoRR*, vol. abs/1909.12962, 2019. [Online]. Available: http://arxiv.org/abs/1909.12962

[15] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *CoRR*, vol. abs/1803.09179, 2018. [Online]. Available: http://arxiv.org/abs/1803.09179

[16] A. Khodabakhsh, R. Ramachandra, K. Raja, and P. Wasnik, "Fake face detection methods: Can they be generalized?" in *Proc. BIOSIG 2018*. IEEE, 2018, pp. 1–6.

[17] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject Agnostic Face Swapping and Reenactment," in *Proc. IEEE/CVF ICCV 2019*. IEEE, 2019, pp. 7183–7192.

[18] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning Lip Sync from Audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 780–789, 2017.

[19] K. R. Prajwal, R. Mukhopadhyay, J. Philip, A. Jha, V. P. Namboodiri, and C. V. Jawahar, "Towards Automatic Face-to-Face Translation," in *Proc. 27th ACM Int. Conf. on Multimedia (MM '19)*. New York, NY, USA: Association for Computing Machinery, 2019, p. 1428–1436.

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

[21] S. D. et al., "A Review on Face Reenactment Techniques," in *Proc. 2020 Int. Conf. on Industry 4.0 Technology (I4Tech)*. Pune, India: IEEE, 2020, pp. 191–194.

[22] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection," *Inf. Fusion*, vol. 64, pp. 131–148, 2020.

[23] Y. Mirsky and W. Lee, "The Creation and Detection of Deepfakes: A Survey," *ACM Comput. Surv.*, vol. 54, no. 1, 2021.

[24] B. Martinez, M. F. Valstar, B. Jiang, and M. Pantic, "Automatic Analysis of Facial Actions: A Survey," *IEEE Transactions on Affective Computing*, vol. 10, no. 3, pp. 325–347, 2019.

[25] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," *arXiv preprint arXiv:1811.00656v3*, 2019.

[26] K. Vougioukas, S. Petridis, and M. Pantic, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking," *arXiv preprint arXiv:1806.02877v2*, 2018.

[27] P. Charitidis, G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," *IEEE Access*, vol. 8, pp. 83 144–83 154, 2020.

[28] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic Speech-Driven Facial Animation with GANs," *Int. J. Comput. Vis.*, vol. 128, p. 1398–1413, 2020.

[29] H. X. Pham, Y. Wang, and V. Pavlovic, "Generative Adversarial Talking Head: Bringing Portraits to Life with a Weakly Supervised Neural Network," *arXiv preprint arXiv:1803.07716*, 2018.

[30] H. H. Nguyen, N.-D. T. Tieu, H.-Q. Nguyen-Son, V. Nozick, J. Yamagishi, and I. Echizen, "Modular Convo-

lutional Neural Network for Discriminating between Computer-Generated Images and Photographic Images," in *Proc. 13th Int. Conf. on Availability, Reliability and Security*. New York, NY, USA: Commun. ACM, 2018.

[31] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," *Commun. ACM*, vol. 62, no. 1, p. 96–104, 2018.

[32] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horvath, E. Bartusiak, J. Yang, D. Guera, F. Zhu, and E. J. Delp, "Deepfakes Detection with Automatic Face Weighting," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2851–2859.

[33] J. Kim, S. Han, and S. S. Woo, "Classifying Genuine Face images from Disguised Face Images," in *Proc. 2019 IEEE Int. Conf. on Big Data (Big Data)*, 2019, pp. 6248–6250.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, 2017.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. CVPR*. IEEE, 2016, pp. 770–778.

[36] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. of ICLR 2015, Conference Track*, Y. Bengio and Y. LeCun, Eds., 2015.

[37] OpenAI, "OpenAI API," 2020, available at https://openai.com/blog/openai-api.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929v1*, 2020.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Curran Associates Inc., 2017, p. 6000–6010.

[40] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs," *arXiv preprint arXiv:1907.05047v2*, 2019.

[41] Timesler, "Pretrained Pytorch face detection (MTCNN) and recognition (InceptionResnet) models," available at https://github.com/timesler/facenet-pytorch.

[42] A. Geitgey, "The world's simplest facial recognition api for Python and the command line," available at https://github.com/ageitgey/face_recognition.

[43] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.