A Novel Machine Learning based Method for Deepfake Video Detection in Social Media

Alakananda Mitra
Dept. of Computer Science and Engineering
University of North Texas, USA.
Email: AlakanandaMitra@my.unt.edu

Saraju P. Mohanty
Dept. of Computer Science and Engineering
University of North Texas, USA.
Email: saraju.mohanty@unt.edu

Peter Corcoran
School of Engineering and Informatics
National University of Ireland, Galway, Ireland.
Email: peter.corcoran@nuigalway.ie

Elias Kougianos
Dept. of Electrical Engineering
University of North Texas, USA.
Email: elias.kougianos@unt.edu

***Abstract*—With the advent of deepfake videos, video forgery has become a serious threat. Videos in social media are the most common and serious targets. There are some existing works for detecting deepfake videos but very few attempts have been made for videos in social media. This paper presents a neural network based method to detect fake videos. A model, consisting of a convolutional neural network (CNN) and a classifier network is proposed. Three different structures, XceptionNet, InceptionV3 and Resnet50 have been considered as the CNN modules and a comparative study has been made. Xception Net has been chosen in the proposed model and paired with the proposed classifier for classification. We used the FaceForensics++ dataset to reach the best model. Our model integrated in the algorithm detects compressed videos in social media.**

***Index Terms*—Deepfake, Deep Learning, Depthwise Separable Convolution, Convolutional Neural Network (CNN), Transfer Learning, Social Media, Compressed Video.**

I. Introduction

Artificial intelligence, especially machine learning, manipulates images and videos in such a way that they are often visually indistinguishable from real ones. Among deep learning-based video falsification techniques, deepfake is a serious contender. The term ‘deepfake’ originates from the words ‘deep learning’ and ‘fake’. Use of deep learning networks (DNN) has made the process of creating convincing fake images and videos increasingly easier and faster. In social media, when images or videos are uploaded, they get compressed and resized. Compression causes losses. So, to detect deepfake social media videos we need techniques which will be applicable to highly compressed videos. In this paper we propose a novel method to detect deepfake videos in social media.

The rest of this paper is organized as follows: Section II presents the motivation for our work. Section III focuses on the novel contributions of this paper. Section IV is a review of related works in this field. Our detailed work for deepfake detection is described in Section VI. Section VII presents the theoretical perspective. Section VIII discusses experiments and results, while Section IX states the conclusion and directions for future works.

II. Deepfake is a Social and Economical Issue

In the last two decades face forgery in multimedia has increased enormously. Among the reported works, an image based approach [1] to generate a video in 1997, face replacement of an actor without changing the expression [2] and real time expression transfer [3] in 2015, are important. In 2017, a Reddit user named Deepfake, created some fake videos using deep learning networks. Use of convolution auto encoders [4] and generative adversarial networks [5] made this forgery so sophisticated that the synthesized videos are often visually indistinguishable from real ones. Smartphone applications to manipulate images, like FaceApp, are easily available to anybody.

This disruptive technological change distorts the truth. Many are intended to be funny, but others are not. They could be a threat to national security, democracy, and an individuals identity. People have started to lose faith in the news or images/videos brought to them by media.

III. Novel Contributions of the Current Paper

In this paper we propose a DNN based framework and an algorithm to detect deepfake videos in social media. A system level overview of the network is shown in Fig.1.

A. The Problem and Challenges Addressed in the Current Paper

The problem addressed in this paper lies in the very origin of how a deepfake video is created. It is hard to distinguish between a real and a deepfake video in social media. Our goal is to model a framework which can detect those deepfake videos in social media generated mostly by an autoencoder. The challenge is threefold: (1) detecting deepfake videos, (2) creating a model applicable to compressed video, and (3) creating a lighter version of it. Our paper addresses the first two problems together.

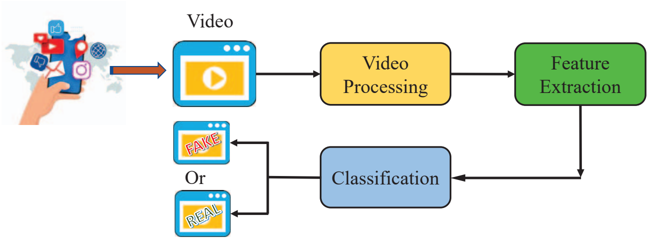


Fig. 1. System Level Overview of the Proposed Network.

B. The Solution Proposed in the Current Paper

We assume a video is manipulated when at least one frame of the video is forged. This assumption is the foundation of our proposed algorithm. Our main contributions are as follows:

- An algorithm of low complexity to detect deepfake videos.
- A network consisting of two modules - (1) a convolutional neural network for frame feature extraction, and (2) a proposed classifier network for detecting deepfake videos. To choose the best CNN module we made a comparative study of Xception, InceptionV3 and Resnet50. Table I gives a comparative study on different networks.
- For training we primarily used the uncompressed and compressed deepfake and original videos at different compression levels of the Face Forensics ++ data set [6]. We trained our network with an intermediate compression level to have a general model which can be applied to other compressed videos.

TABLE I
A Comparative Perspective with Different CNN Models

Model	Size (MB)	Parameters (Millions)	Top-1 Accuracy (%)	Top-5 Accuracy (%)
GoogLeNet	40	7	71.8	90.7
AlexNet	217	60	57.1	84.7
VGG16	528	138.83	71.3	90.1
VGG19	549	143.67	71.3	90.0
InceptionV3	92	23.8	77.9	93.7
ResNet50	98	25.6	74.9	92.1
Xception	88	22.9	79.0	94.5

C. The Novelty of the Solution Proposed

In this paper, we propose a novel algorithm with a deep neural model to detect social media deepfake videos with high accuracy. We also consider a small size model as feature extractor so that we can extend our model to edge devices as future work.

IV. Related Prior Works

Most of the solutions proposed for video forensics are for easy manipulations such as copy-move manipulation [7], dropped or duplicated frames [8], or varying interpolation [9]. But the use of auto-encoders or generative adversarial networks made image/video forgery sophisticated. Existing works are presented in Table II.

Among deep learning solutions, some are temporal feature based and some are based on visual artifacts. In visual-artifact based works, videos are processed frame-by-frame. Each frame contains different features. These features are first extracted and then used as input to a deep learning classifier as CNN models can detect these artifacts. The classifiers are ResNet152 [10], VGG16, Inception V3, DenseNet etc. Certain works are associated with detection techniques based on eye blinking rate [11], noting the difference between head pose [12] of an original video and fake video, and detecting the artifacts of eyes, teeth and face [13]. A general capsule network based method has been proposed to detect manipulated images and videos [14]. A VGG-19 [15] network has been used for latent feature extraction along with a capsule network to detect different spoofs, replay attack etc. Two inception modules along with two classic convolution layers followed by maxpooling layers have been explored [16]. A combined network of CNN and LSTM architectures has been explored in [17]. A DenseNet structure combined with RNN has been used [18]. A blockchain based approach to detect forged videos is proposed in [19]. From Table II, it is evident that not much work has been done for compressed video which is predominantly used in social media. A triplet structure has been used to detect highly compressed videos [20].

V. Why are Deepfakes Hard to Detect?

There are two main ways to create Deepfake videos - by autoencoders and by generative adversial networks (GANs). Our method addresses deepfake videos generated by autoencoders, in which the creation of deep fake video consists of three steps - extraction, training and creation.

- In the extraction process all frames from video clips are extracted and faces are identified and aligned.
- The training stage is shown in Fig. 2(a). During training, common features for both image sets are created.
- The creation of a deepfake video frame is shown in Fig. 2(b).

VI. The Proposed Novel Method for Deepfake Detection

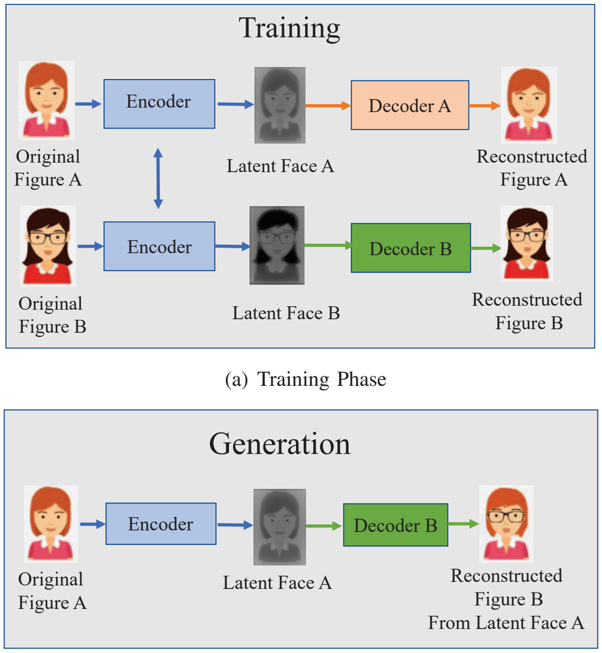
From Table I it is evident that GoogLeNet, ResNet50, InceptionV3 and Xception Net are smaller size than the other models. We preliminary chose three architectures except GoogLeNet as the feature extractor of our model as those models have better accuracy than GoogLeNet. We then compared the results and finally propose a final model. The end-to-end framework of our proposed method is shown in Fig. 3.

The framework consists of (1) CNN module (2) a classifier network. Three different CNN modules are used initially to get the best feature extractor for compressed video. Finally Xception Net has been used as our model feature extractor.

- *Data processing:* A dataset of videos each 4 sec long are clipped from the original and manipulated videos.Then frames are extracted from each compressed video with no decompression. We then detected the faces and cropped

TABLE II
A COMPARATIVE PERSPECTIVE WITH EXISTING WORKS ON DEEPFAKE VIDEO DETECTION.

Works	**DataSet**	**Model Features**	**Remarks**
Sabir et al. [18]	FaceForensics++	Use spatio-temporal features of video streams. Bidirectional RNN + DenseNet/ResNet50.	Not applicable to long video clips. Not trained on a large dataset.
Güera and Delp [17]	HOHA	Temporal inconsistencies of deepfake video is taken into account. Inception-V3 + LSTM.	Didn't take into account of compressed videos.
Li et al. [11]	CEW	Used Long Term Recurrent Convolutional Networks. Measured the eye blinking rate. VGG16 + LSTM + FC	Applied to uncompressed videos.
Afchur et al. [16]	Downloaded from internet and processed.	Mesonet structures - Meso-4 and MesoInception-4 used. 2 inception modules + 2 classic convolution layers + 2 FC layers.	Accuracy is less for highly compressed video.
Li et al. [21]	UADFV and DeepfakeTIMIT	Face warping artifacts. Used 4 CNN models. Measured resolution inconsistency between the warped face area and face.	Compression has not been considered.
Matern et al. [13]	A combination of various sources.	Facial texture difference, and missing details in eye and teeth. Logistic regression model and neural network.	Not for compressed video.
Nguyen et al. [14]	Four major datasets.	VGG-19 + Capsule Network.	Accuracy is low for highly compressed data.
Proposed Model	FaceForensics++	Face Artifacts Analysis. XceptionNet + Classifier Network.	Designed for compressed video. Applicable to long videos. Faster.



(a) Training Phase

(b) Generation Phase

Fig. 2. Deepfake Video Creation by Autoencoder.

the faces from each frame. Finally all frames are normalized and resized as per the input of the various CNN modules. *Imagesize* is kept at $(299, 299, 3)$ for InceptionV3 and Xception net and $(224, 224, 3)$ for ResNet50. The data processing diagram is shown in Fig. 4.

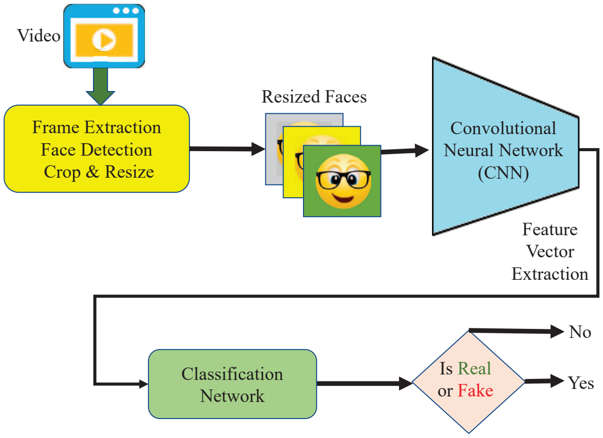


Fig. 3. A Detailed Representation of the Proposed Model.

- *Classification Network* : As the classification network (Fig. 5), we chose a combination of layers for better accuracy. The layers are a GlobalAveragePooling2D layer with 0.5 dropout followed by a fully connected layer with 1024 nodes, 0.5 dropout and 'relu' activation and finally a softmax layer which essentially classifies the detected video as real or manipulated. Fig 6 shows how the classifier works.

.

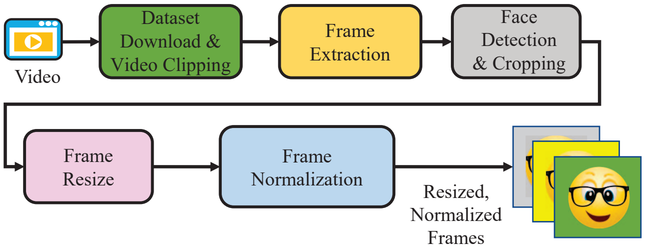

Video
Dataset Download & Video Clipping
Frame Extraction
Face Detection & Cropping
Frame Resize
Frame Normalization
Resized, Normalized Frames

Fig. 4. The Proposed Flow for Video Processing.

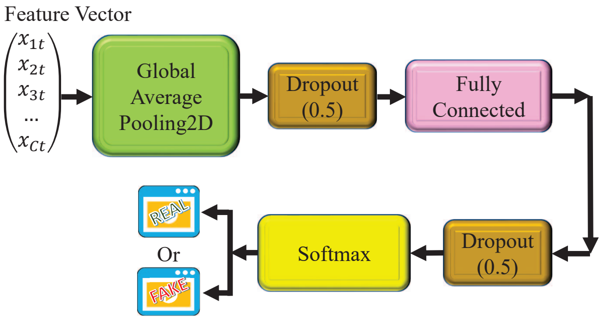


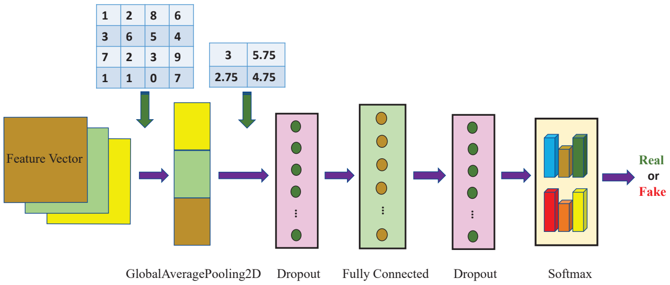
Fig. 5. Classifier Network Architecture.

VII. The Proposed Method - A Theoretical Perspective

Proposed Algorithm: The novelty of our algorithm is in making the complexity of detecting forged videos low. The algorithm is given in Algorithm 1. If the number of extracted frames from a video is n, the time complexity is $\mathbf{O}(n)$. Our algorithm stops checking the authenticity of a video when the first fake frame is detected. So, the best case scenario is when the first frame is detected as fake. The algorithm considers the video as fake. No more computation is needed. The worst case scenario is when the last frame is detected as fake.

Depthwise Separable Convolution: There are three elements in a convolution operation -

- Input image
- Feature detector or Kernel or Filter
- Feature map



1	2	8	6
3	6	5	4
7	2	3	9
1	1	0	7

3	5.75
2.75	4.75

Fig. 6. Classifier Network Work Flow.

Algorithm 1: How to Detect DeepFake Video?

```
Extract frames from the test video.
Store frames in a dataframe.
Detect and crop face from each frame.
Resize each image.
Save them in another dataframe.
Load the saved model.
for an image in the range of resized images do
    Check for authenticity
    if image is real then
        continue
    else
        Consider the video Fake
        break
```

The Kernel or Filter or Feature Detector is a small matrix of numbers. When it is passed over the input image, new feature maps are generated from the convolution operation between the filter value and the pixel value of the input image at each point (x, y). The complexity of the convolution operation is $N \times D_G^2 \times D_K^2 \times M$, where $D_F \times D_F \times M$ is the size of the input image and the filter size is $D_K \times D_K \times M$. M is the number of channels in the input image. The size of the feature matrix is denoted by $D_G \times D_G \times M$. The complexity is decreased in *Depthwise Separable Convolution*. It divides the convolution operation in two parts (1) Depthwise Convolution - Filtering stage and (2) Pointwise Convolution - Combination stage. In depthwise convolution the complexity is $M \times D_G^2 \times D_K^2$ while for pointwise convolution it is $N \times D_G^2 \times D_K^2 \times M$. The total complexity is expressed by the following expression:

$$Total\ Complexity = M \times D_G^2 \times D_K^2 + N \times D_G^2 \times D_K^2 \times M \quad (1)$$

The relative complexity is the following expression:

$$\frac{Complexity\ Depthwise\ Separable\ Conv.}{Complexity\ Standard\ Conv.} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2)$$

It is obvious from Eq. (2) that the complexity of standard convolution is much higher than the depthwise separable convolution, which implies that Xception Net is much faster and cheaper convolution than standard convolution.

GlobalAveragePooling Layer: It helps to reduce the number of parameters and eventually minimizing overfitting. It downsamples by computing mean or average of the width and height dimensions of the input.

Dropout Layer: It is very common for a deep network to overfit. The dropout layer prevents the overfitting of a neural network.

Soft-Max Layer: In order to predict the class of the video - pristine or manipulated, the softmax layer is used at the end of the network. It takes an M-dimensional vector and creates another vector of the same size but with values ranging from 0 to 1 making the sum of the values to 1.

Training Loss: During training, we minimize the *Categorical Cross Entropy Loss* to get optimal parameters of the network to best predict the class. It is a measure of performance of a classification model whose output is the probability ranging from 0 to 1.

VIII. Experimental Validation

A. Dataset

The FaceForensics++ dataset [6] by Google has videos at different compression levels. We used the deepfake videos of this dataset for training and evaluating our model, as it represents a realistic scenario for social media. The dataset details are given in Table III.

TABLE III
Dataset Details

Dataset Name	Compression = c23	
	No. of Original Videos	No. of Manipulated Videos
FaceForensics++	1000	1000

B. Experimental Setup

Transfer Learning: We used transfer learning for better accuracy and save training time. All CNN modules are trained on the Imagenet dataset. Resnet50, InceptionV3 and Xception Net were used as feature extractors in our experiment. Lower level layers extract basic features like lines or edges whereas middle or higher layers extract more complex and abstract features and features defining classification. We detected faces and cropped them from each frame of the video to get a better features set. We finally used Xception Net as feature extractor and connected it to a classifier network. We trained the classifier with the dataset and then fine tuned the network end-to-end.

Parameter Settings: During the normalization of each frame the mean and standard deviation are both set to 0.5. The Adam optimizer [22] has been used for training the whole network. The whole work is shown in Fig. 7. The details of parameter settings are shown in Table IV.

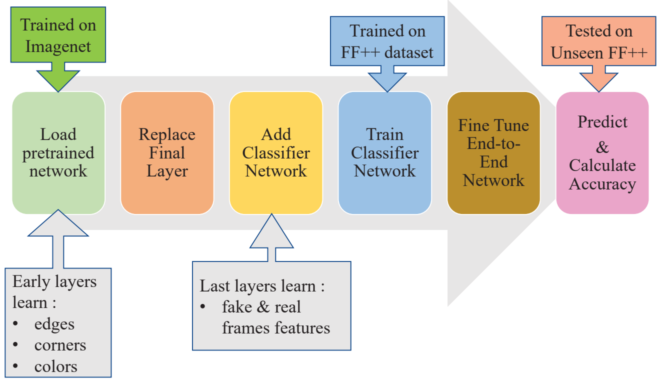


Fig. 7. End-to-End Work Flow.

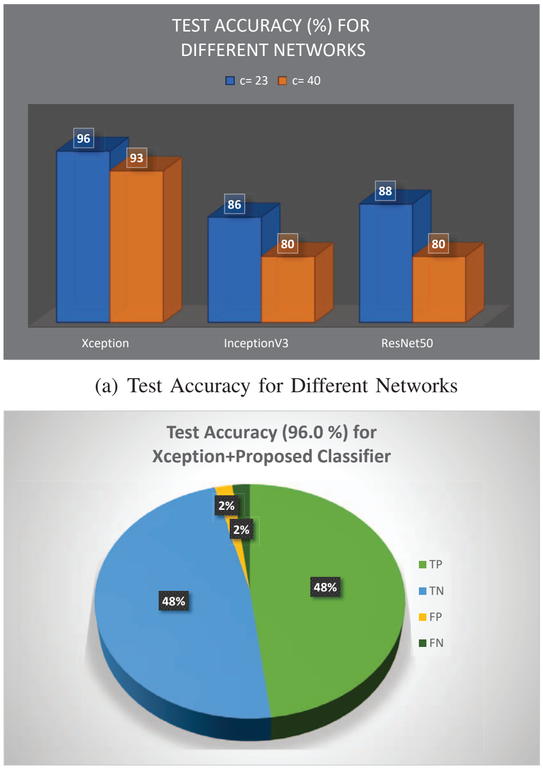
Implementation Details: We implemented our proposed framework in Keras with the TensorFlow backened. FFmpeg [23] is used to clip the videos. For training we used a Tesla T4 GPU with 64GB Memory. A GeForce RTX 2060 is used to evaluate the model.

TABLE IV
Parameter Details

Training	Parameters	
	learning rate	batch_size
Classifier	$5e^{-4}$	32
Fine Tunning	$5e^{-5}$	16

C. Experimental Verification

We verified our model with unseen data from FaceForensics++. Our model with Xception Net gave the best accuracy among all three CNN modules. The accuracy for compression level c=23 is better than c=40. Fig.8(a) shows the accuracy vs model with different CNN networks.



(a) Test Accuracy for Different Networks

(b) Test Accuracy Calculation

Fig. 8. Experimental Results

D. Analysis of Results

The results for the accuracy are shown in Table V.

All three CNN modules have been paired with our classifier network. Xception net combined with our classifier gave the best accuracy of 96.00% for compression level $c = 23$ and 93% for $c = 40$. Test accuracy for each model for different compression level is given in Table VI.

Test accuracy details for the proposed model are shown in Fig. 8(b). When we compared our result with the result published in the FaceForensics++ paper [6], we see accuracy

TABLE V
DATA TO CALCULATE TEST ACCURACY

Compression Levels (for 100 videos)	Network Names	Number of Test Data			
		TP	TN	FP	FN
FF++, c=23	ResNet50	40	48	02	10
	InceptionV3	42	44	06	08
	Xception	48	48	02	02
FF++, c=40	ResNet50	38	42	08	12
	InceptionV3	46	34	16	04
	Xception	45	48	02	05

TABLE VI
TEST ACCURACY

Testing Data	Compression	ResNet50	InceptionV3	Xception
FaceForensics++	c=23	88.00	86.00	**96.00**
	c=40	80.00	80.00	**93.00**

increase at both compression levels. For our experiment we did training with only $c = 23$ videos. Fig. 9 shows a comparative picture between our work and the FaceForensics++ paper.

IX. CONCLUSION AND FUTURE WORK

In this paper, we present a deep learning based approach to detect deepfake videos in social media with a high accuracy. We use a neural network based method to classify pristine and manipulated video. We compared three existing CNN modules and finally chose Xception net as the feature extractor paired with the proposed classifier for the most accurate model. We trained the network with intermediate compression and achieve high accuracy even at high loss scenario. Our proposed algorithm is the key factor in getting high accuracy even without training with highly compressed videos. The complexity of the algorithm is proportional to the number of frames extracted from the video. There is no restriction on video length. In future work we plan to deploy our model at edge devices with appropriate modification.

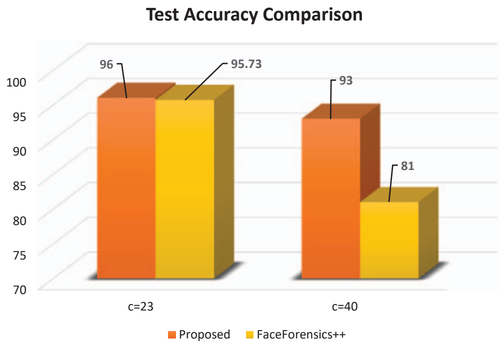


Fig. 9. Test Accuracy Comparison

REFERENCES

[1] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 353–360.

[2] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormaehlen, P. Perez, and C. Theobalt, "Automatic face reenactment," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4217–4224.

[3] J. Thies, M. Zollhofer, M. Niener, L. Valgaerts, M. Stamminger, and C. Theobalt, "Realtime expression transfer for facial reenactment," in *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2015*, ser. Art No.183, vol. 34, no. 6, 2015.

[4] A. Tewari, M. Zollhfer, H. Kim, P. Garrido, F. Bernard, P. Prez, and C. Theobalt, "MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3735–3744.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[6] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE International Conference on Computer Vision*, 2019, pp. 1–11.

[7] L. DAmiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A patchmatch-based dense-field algorithm for video copymove detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 669–682, 2019.

[8] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6226–6230.

[9] X. Ding, G. Yang, R. Li, L. Zhang, Y. Li, and X. Sun, "Identification of motion-compensated frame rate up-conversion based on residual signals," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 7, pp. 1497–1512, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] Y. Li, M. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking," in *Proc. IEEE International Workshop on Information Forensics and Security*, 2018, pp. 1–7.

[12] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8261–8265.

[13] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. IEEE Winter Applications of Computer Vision Workshops*, 2019, pp. 83–92.

[14] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2307–2311.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[16] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a compact facial video forgery detection network," in *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1–7.

[17] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. 15th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2018, pp. 1–6.

[18] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 80–87.

[19] H. R. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *IEEE Access*, vol. 7, pp. 41 596–41 606, 2019.

[20] A. Kumar, A. Bhavsar, and R. Verma, "Detecting deepfakes with metric learning," in *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, 2020, pp. 1–6.

[21] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *CoRR*, vol. abs/1811.00656, 2018. [Online]. Available: http://arxiv.org/abs/1811.00656

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[23] M. N. Fabrice Bellard and others., "Ffmpeg," http://ffmpeg.org, 2012.