FSSPOTTER: SPOTTING FACE-SWAPPED VIDEO BY SPATIAL AND TEMPORAL CLUES

Peng Chen[1,2]*, Jin Liu*[1,2]*, Tao Liang*[1,2]*, Guangzhi Zhou*[1,2]*, Hongchao Gao*[1,2]*, Jiao Dai*[1]*, Jizhong Han*[1]

1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100049, China

ABSTRACT

Recent advances in face generation and manipulation have enabled the creation of sophisticated face-swapped videos, also known as DeepFakes, which brings great potential threats to our society. Hence, it is crucial to develop effective approaches to distinguish them. Currently, face-swapped videos produced by existing methods are prone to exhibit some subtle spatial and temporal manipulated traces, which can be utilized as distinctive clues for face-swapped video detection. In this paper, we propose a unified framework, named FSSpotter, to explore rich spatial and temporal information in the video simultaneously. It consists of a Spatial Feature Extractor (SFE), which aims to discover spatial evidences within a single frame, and a Temporal Feature Aggregator (TFA), which is responsible for capturing temporal inconsistencies between frames. Moreover, a novel data processing strategy is adopted to highlight the inconsistencies of forged face with its surrounding regions. The evaluations on Deepfakes of FaceForensics++, DeepfakeTIMIT, UADFV and Celeb-DF datasets demonstrate that the proposed approach achieves better or comparable performance on AUC scores.

Index Terms— DeepFakes, Face-swapped video, DeepFakes detection, Face Forensics

1. INTRODUCTION

Face swapping [1] refers to replace the facial identity of a person with a target facial identity while preserving facial expression and illumination. Recently, with significant improvements in deep generative models [2, 3], some tools [4, 5] are released on open source platforms, which are used to seamlessly swap one's face to another's face in a video, significantly reducing the barriers to the creation of facial replacement videos and thus causing a hot deepfakes phenomenon. However, the misuse of such techniques will bring unimaginable threats to our society, especially when the malicious and forged contents spread virally on social networks. People worry that their faces will be swapped into a sex video, which makes a extremely bad impact on their careers and families[1]. Besides, politicians panic that those manipulation videos could be used as terrible political weapons, which will incite political violence or sabotage elections[2].

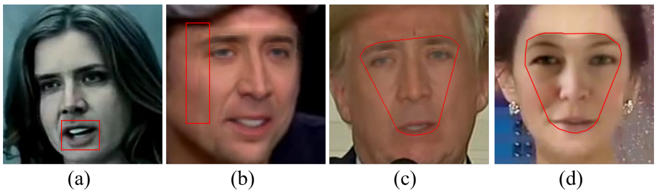

Fig. 1. Some examples of spatial traces: (a) geometric deformations of teeth, (b) doubled facial contour, (c) and (d) respectively show different image quality and illumination conditions between the forged face (inside red line) with its surrounding regions. (Zoom in to see better)

Fortunately, when producing a face-swapped video, existing methods are far from flawless and exhibit minor visible artifacts. By carefully analyzing the forged videos and its synthesis process including converting, splicing, color matching and blending, several subtle traces of manipulation are observed and can be summarized into three categories. From the perspective of a single frame, (1) there are many subtle differences in color and texture of facial areas between real images and fake ones, such as geometric deformations of teeth (Fig. 1(a)), a doubled facial contour (Fig. 1(b)) and so on. (2) In the interior of a fake image, the forged face and its surrounding regions may exist some inconsistent characteristics, such as different image quality (Fig. 1(c)) or illumination conditions (Fig. 1(d)). From the perspective of a video, (3) the facial areas of consecutive frames also leave some manipulated traces, such as temporal flickering, because each image in the forged video is processed separately and lacks contextual information.

Recently, many image-based approaches [6, 7, 8] have been proposed to detect face-swapped video by utilizing a simple CNN, but all of them only considered spatial information in a single frame while ignoring temporal information in a video. Some video-based approaches [9, 10] typically adopt two-stage strategy. Güera *et al.* [9] first employed a pre-trained CNN to extract deep representation of each frame

[1] https://www.indiatoday.in/trending-news/story/journalist-rana-ayyub-deepfake-porn-1393423-2018-11-21

[2] https://www.niemanlab.org/2019/06/how-could-deepfakes-impact-the-2020-u-s-elections/

978-1-7281-1331-9/20/$31.00 ©2020 IEEE

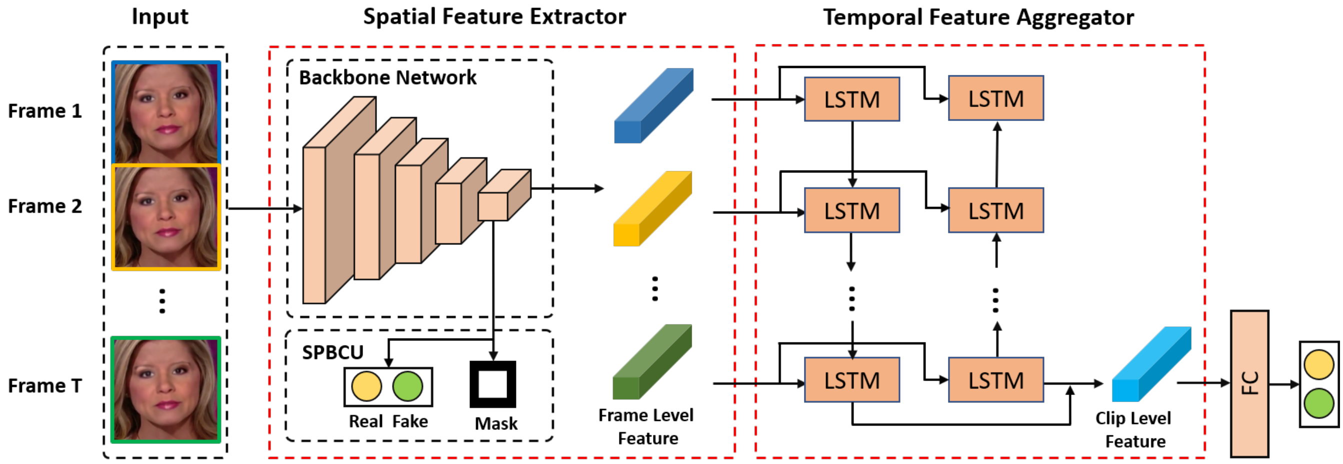

Fig. 2. The overall framework of our FSSpoter, which contains two main parts: Spatial Feature Extractor (SFE) and Temporal Feature Aggregator (TFA). The SFE is designed to discover spatial evidences within a single frame. It consists of a backbone network and a Superpixel-wise Binary Classification Unit (SPBCU). The TFA takes a bidirectional LSTM to capture temporal inconsistencies between frames.

and these features were then used to train a RNN to discover temporal traces. However, this work has two drawbacks: (a) It only focuses on temporal information while not fully utilizing CNN to explore rich spatial information in single frame, which is also important clues for face-swapped video detection. (b) The whole image is sent into the CNN to extract features, which contain too much irrelevant information from the background. Sabir *et al.* [10] extended this work by utilizing more advanced CNN architectures and trained the framework end-to-end, yet they still ignored the differences between the forged face with its surrounding regions.

In order to effectively utilize all three types of aforementioned manipulated traces to distinguish face-swapped video, we propose a unified framework named FSSpotter, consisting of a spatial feature extractor (SFE) and a temporal feature aggregator (TFA). The SFE exploits a CNN based backbone network which is responsible for discovering spatial evidences in single frame. Furthermore, a Superpixel-wise Binary Classification Unit (SPBCU) is specifically designed to guide the backbone network to particularly focus on the differences between forged face with its surrounding regions. The TFA takes a bidirectional LSTM [11] to capture temporal inconsistencies between consecutive frames. Additionally, compared with prior works [9, 10], we adopt a different face cropping method, which crops more contextual contents around the facial areas, and a simulation strategy for spatial inconsistencies to ensure that the SFE can learn to extract discriminative features based on the inconsistencies of facial region with its surrounding regions.

The primary contributions of this work are summarized as follows. First, we propose a unified framework named FSSpotter, which is composed of a SFE and a TFA, to seek spatial and temporal clues in the face-swapped video. Second, we try to boost the performance of our framework by designing a SPBCU and a novel data processing strategy. Third, Experiments on several datasets demonstrate that the proposed method achieves better or comparable performance on AUC scores.

2. RELATED WORK

In this section, we introduce previous works on face swapping and face manipulation detection.

Face Swapping Methods. Some early works on face swapping mainly utilized graphics-based approaches, focusing on the alignment of two faces and various face blending techniques [1, 12, 13]. Inspired by artistic style transfer, Iryna *et al.* [14] designed a fast face-swapped framework by treating expression as the content and identity as the style. Recently, the excellent ability of deep generative models, such as VAE [2] and GANs [3], were successfully employed to make forged videos more sophisticated. The Deepfake-AE [4] generated more realistic facial images, which is based on auto-encoder architecture with a shared encoder and two decoders of the source and target face. Nirkin *et al.* [15] proposed a unified framework, which is subject agnostic and can be applied to pairs of faces without requiring training on those faces, to produce photo-realistic and temporally coherent videos.

Face Manipulation Detection. Previous works on face-swapped video detection approaches fall into two categories: image-based [16, 6, 7, 17, 8, 18] and video-based [19, 20, 9, 10]. The former treated video as an image set and converted the problem to detect fake faces in a single frame, whereas the latter attempted to model the temporal information between frames.

A two-stream CNN [16] were proposed for tampered face detection, combining RGB space features with steganalysis features. Afchar *et al.* [6] built two kinds of CNN-based networks to detect manipulated face efficiently. Li *et al.* [7] argued that face swapping algorithm needs to warp generated images to match the original faces, which left distinctive warping artifacts and can be effectively captured by CNN models. Yang *et al.* [17] utilized the inconsistency in 3D head poses as clues to distinguish real and fake videos. Rössler *et al.* [8] released a large facial manipulation dataset called FaceForensics++ and trained a XceptionNet to classify them. Matern *et al.* [18] spotted fake faces by designing handcrafted features for visual artifacts regarding eyes, teeth and facial contours. However, all of these methods only consider spatial information in a single image while ignoring temporal information in video.

Some video-based methods sought for physiological signals as clues, like eye blinking [19] and rPPG [20], but quickly lost its effect when facing more training data and advanced generative models. Güera *et al.* [9] proposed a two-stage strategy composed of a CNN to extract frame-level features followed by a RNN to capture temporal inconsistencies between frames. However, this work only used a pre-trained InceptionV3 which wasn't fine-tuned on any fake video, resulting that the model can't capture spatial manipulated artifacts well. Besides, the whole image was sent into this model, causing too much irrelevant information from the background to be contained in the final feature maps. Sabir *et al.* [10] extended this work by adopting more powerful CNN architectures followed by a bidirectional RNN and trained the framework end-to-end, still ignoring the inconsistencies between the forged face with its surrounding regions. Compared with [9, 10], our approach utilizes SPBCU and a novel data processing strategy to discover various spatial and temporal inconsistencies as clues to distinguish face-swapped video.

3. METHODOLOGY

In this section, we first introduce the details of our proposed framework, which seeks out the spatial and temporal clues to distinguish the face-swapped video. Then, the data processing strategy and loss functions are described respectively.

3.1. Framework

The proposed framework consists of a SFE and a TFA, as shown in Fig. 2.

Spatial Feature Extractor. A video is first cut into consecutive clips, and each clip contains T frames. The SFE takes clips as input and generates the frame level features $\mathbf{F} \in \mathbb{R}^{B\times T\times C\times H\times W}$, where B , T, C, H and W, are batch size, number of frames, number of channels, height and width, respectively. It is composed of a backbone network and a superpixel-wise binary classification unit (SPBCU). In this work, we adopt the convolution layers of VGG16 with batch normalization as the backbone network, which is responsible for extracting spatial features in intra-frame. Besides, a SPBCU is exploited to promote the backbone network to extract more distinctive features.

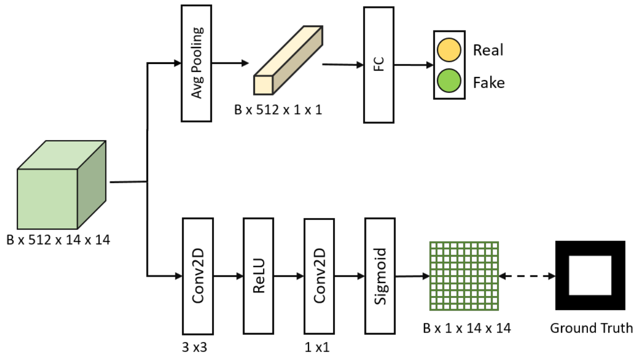

Fig. 3. Illustration of superpixel-wise binary classification unit (SPBCU). It is composed of a global binary classification branch and a superpixel-wise binary classification branch.

Superpixel-wise Binary Classification Unit. The SPBCU is a pluggable component that provides additional supervision for training the backbone network. For all frames in a clip, we observed little change in the shape of forged regions. Therefore, only the feature maps (i.e. `relu5_3`) of the first frame are sent into SPBCU, which allows for a large reduction in training time. The SPBCU consists of a superpixel-wise binary classification branch and a global binary classification branch, as shown in Fig. 3. The former contains a 3×3 convolution layer, which is used for increasing the receptive field, and a 1×1 convolution layer, which generates feature maps $\mathbf{M} \in \mathbb{R}^{B\times 1\times 14\times 14}$, determining whether each spatial point in it belongs to the forged areas or not. At the same time, the latter comprehensively considers holistic information to distinguish whether the frame is fake or not. The SPBCU further guides the backbone network to compare facial region with its neighboring regions, which enhances the generalization ability of the framework.

Temporal Feature Aggregator. The TFA employs a bidirectional LSTM with 512 hidden units to discover temporal inconsistencies between frames. The outputs of SFE are squeezed into $\mathbf{F}' \in \mathbb{R}^{B\times T\times 512}$ by an average pooling before being sent to the TFA. The final hidden state of the bidirectional LSTM is used as clip level features. Then, a fully-connected layer and a softmax layer are exploited to compute the probabilities of whether the clip is manipulated or not.

3.2. Data Processing Strategy

Two kinds of strategies are adopted to ensure that our proposed framework can learn spatial inconsistencies to make decisions.

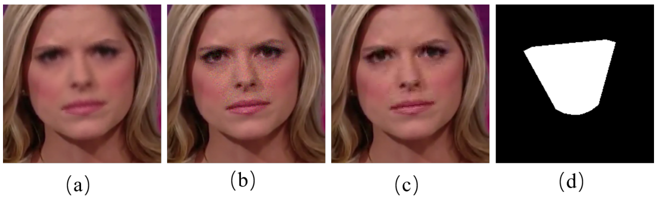

Fig. 4. Examples for spatial inconsistencies simulation. (a) for Gaussian blurring. (b) for Gaussian noise. (c) for JPEG compression. (d) is the mask of manipulated regions.

Face Cropping. Considering that the frames of video contain too much irrelevant information, it is critical to crop regions of suitable size. We first use dlib [21] to detect the facial landmarks. Then, we get the minimal square box by warping those landmarks. Finally, the box is enlarged by 1.6 times as cropped region, where not only the core areas of face but also sufficient surrounding areas are covered. All cropped images are resized to 224×224. The proposed framework is trained on the Deepfakes subset of FaceForensics++ [8], which also provides the mask of manipulated region. Thus, those masks are cropped by the same box size and resized to 14×14, used as the ground truth of SPBCU output.

Spatial Inconsistencies Simulation. We have observed that there are many subtle differences between the forged face with its surrounding regions in the fake image. However, for deep learning method, collecting enough supervised training data is truly a challenge. Especially for face manipulation detection, it is very costly to create a large-scale high-quality forged dataset. Inspired by [7], we address this problem by adopting a simulation for manipulated traces to automatically generate fake training data, which highlights spatial inconsistencies in the fake image. We first crop aligned face directly from an authentic image and randomly scale it between 64 and 128. Then, a degradation method, randomly sampled from Gaussian blurring, Gaussian noise and JPEG compression, is operated on cropped areas. Finally, we get a fake image by pasting the manipulated area into original image. Some examples are showed in Fig. 4.

3.3. The Loss Functions

We adopt Binary Cross Entropy as the loss function.The equation for the SPBCU loss is shown below.

$$\mathcal{L}_{\text{spbcu}} = \lambda_s * \mathcal{L}_{\text{superpixel}} + \lambda_g * \mathcal{L}_{\text{global}}, \tag{1}$$

where $\mathcal{L}_{\text{superpixel}}$ is the superpixel-wise binary classification loss and $\mathcal{L}_{\text{global}}$ is the global binary classification loss. λ_s and λ_g are the respective loss weights and both are set as 1 in this work.

The loss for the final binary classification is as follows:

$$\mathcal{L}_{\text{binary}} = \mathcal{L}_{\text{bce}}(\hat{y}, y), \tag{2}$$

where $\hat{y}$ is the predicted label and y is the ground truth($y = 0$ for real and $y = 1$ for fake).

4. EXPERIMENT

In this section, we first describe four face-swapped datasets. Then, the details of the implementation of proposed method are introduced. Finally, the results of our experiments are presented and analyzed.

4.1. Datasets

The FaceForensics++ [8] consists of 1000 original video sequences, which are manipulated by five different methods, with 720 in training and 140 for each of validation and test. Besides, the dataset covers three different versions based on compression including Raw, c23 and c40. In this experiment, we only use the Deepfakes subset of FaceForensics++ at c23 (FF++ / DF).

The DeepfakeTIMIT [22] contains 640 tampered videos generated by GAN-based approach from 32 subjects of VidTIMIT dataset. Among all the videos, 320 are of high quality and the rest 320 are of low quality. We choose subset of each subject from VidTIMIT dataset as corresponding authentic videos.

The Deepfake dataset UADFV is generated to validate the algorithm in [19]. The dataset includes 49 real videos and 49 fake videos respectively. Each video contains one subject, while the facial areas of all real videos are replaced by the face of the same subject.

The Celeb-DF [23] is constructed for evaluation of detection approaches. The dataset contains 408 real videos and 795 synthesized videos generated with refined DeepFake algorithms. In this experiment, we use the test dataset which consists of 61 fake videos and 32 real videos.

As mentioned before, a video is cut into consecutive clips, and each clip contains T frames. In this work, we set T as 8. The numbers of clips sampled from each dataset are shown in Table 1.

Table 1. The numbers of clips sampled from each dataset.

	FF++ / DF		DeepfakeTIMIT		UADFV	Celeb-DF
	train	test	HQ	LQ		
Real	60480	12169	5108	5108	3174	3937
Fake	60480	12169	5207	5212	3123	3695

4.2. Implementation Details

Jointly training SFE and TFA is time consuming and tedious to optimize. Thus, to better optimize different modules of the proposed framework, we adopt a two-stage training strategy, learning spatial inconsistencies and temporal clues separately.

In the first stage, we only optimize the backbone network with SPBCU. We initialize the backbone network by using

Table 2. AUC(%) performance of each method. All AUC scores are rounded to one digit after the decimal point.

Methods	FF++ / DF	DeepfakeTIMIT HQ	 LQ	UADFV	Celeb-DF
Two-stream [16]	70.1	73.5	83.5	85.1	55.7
Meso-4 [6]	84.7	84.3	87.8	84.3	53.6
DFWA [7]	79.2	93.2	**99.9**	**97.4**	53.8
Xception [8]	99.5	94.1	97.5	88.9	72.2
FSSpotter(Ours)	**100.0**	**98.5**	99.5	91.1	**77.6**

Table 3. AUC(%) performance of the backbone network with different components.

Methods	Intra-test	Cross-test			
	FF++ / DF	DeepfaktTIMIT HQ	 LQ	UADFV	Celeb-DF
Backbone	99.94	95.28	95.64	81.14	72.6
+Sim	99.95	95.1	93.58	92.11	75.1
+SPBCU	99.97	97.43	99.16	78.69	71.35
+SPBCU+Sim	99.97	97.33	98.99	90.03	76.26

pre-trained weights on ImageNet. The batch size is set as 16 with 7 real images, 8 face-swapped images and 1 simulated image. We use SGD optimizer with momentum of 0.9 and an initial learning rate of 10^{-3}, which decays 0.9 after each 1000 steps. The training process is terminated at 30000 steps.

In the second stage, we fix the backbone network and train the bidirectional LSTM with the final classification layer together. The SPBCU is dropped, because the backbone network has learned to focus on the inconsistencies between the forged face with its surrounding region. The initial learning rate of 10^{-2} decays 0.1 every 1000 steps. We stop the training process at 8000 steps.

4.3. Results and Analysis

We compare our method with recent deep learning based detection methods, including Two-stream NN [16], Meso-4 [6], DFWA [7] and Xception (trained on c23 version) [8]. Area under the receiver operating curve (AUC) [24] scores are reported as metric. Similar to [23], all AUC scores are rounded to five digit after the decimal point. Note that our proposed framework is only trained on FF++ / DF while directly evaluated on other datasets.

Table 2 shows the performance of all compared methods on these datasets. Both Xception and FSSpotter are trained on FF++ / DF, thus can achieve high performance on this dataset. Even if FSSpotter only takes a simple VGG16 as the backbone network, it is superior to Xception by 2.2% for UADFV, 5.4% for Celeb-DF and 4.4%, 2.0% for DeepfakeTIMIT HQ, LQ respectively. DFWA trained a ResNet50 on a dataset collected by authors, so it exhibits satisfactory performance on DeepfakeTIMIT and UADFV but drops significantly on FF++ / DF and Celeb-DF. The results demonstrate that FSSpotter can achieve better or comparable performance on all datasets even if it is not trained on some of them.

4.4. Ablation Studies

To show the effectiveness and better generalization of the proposed framework, we conduct ablation studies on two different settings: (a) training and testing on FF++ / DF (i.e. intra-test), and (b) training on FF++ / DF while directly evaluated on other datasets (i.e. cross-test).

Effect of SPBCU and Simulation: We start from the backbone network with a global binary classification branch, then gradually add components of the approach, including SPBCU and spatial inconsistencies simulation. From Table 3 we observe that: 1) by only introducing SPBCU, the performance improves significantly while drops on UADFV and Celeb-DF. We argue that the manipulated regions of FF++ / DF and DeepfakeTIMIT are square boxes (shown in Fig. 3), while the manipulated regions of UADFV and Celeb-DF are the shape of minimal face (shown in Fig. 4(d)). The model may slightly overfit to this specific shape due to training on FF++ / DF. 2) by only introducing spatial inconsistencies simulation, the improvement of UADFV and Celeb-DF are significant, which demonstrates that focusing on the inconsistencies between forged face with its surrounding regions is extremely effective for some datasets. 3) by combining both, it shows a promising performance on all datasets, which also indicates that the model has learned a generalisable representation for unseen manipulation types.

Table 4. AUC(%) performance of FSSpotter with different clip length $T = 1, 4, 8, 12$.

Clip Length	Intra-test	Cross-test			
	FF++ / DF	DeepfakeTIMIT HQ	 LQ	UADFV	Celeb-DF
T=1	99.97	97.33	98.99	90.03	76.26
T=4	99.98	98.3	99.33	90.74	76.53
T=8	99.98	98.46	99.5	91.06	77.59
T=12	99.98	98.47	99.36	91.31	76.31

Effect of different clip length (T): Although introducing temporal information can increase detection performance, it will cost more resource overhead during the training and testing phases. Thus, we analyze with clip length of $T = 1, 4, 8$ and 12 in FSSpotter and when $T = 1$ we do not need to use TFA. The results are shown in Table 4, we can see that $T = 8$ already achieves an acceptable performance, which indicates the importance of temporal information and the effectiveness of TFA.

5. CONCLUSION

In this work, we have presented FSSpotter, a unified framework which can effectively capture spatial and temporal inconsistencies as clues for face-swapped videos detection. Besides, a superpixel-wise binary classification unit and a novel data processing strategy are adopted to make the framework pay more attention on the spatial inconsistencies in the manipulated videos. Experimental results demonstrate that our method achieves better or comparable performance compared to other recent methods.

6. REFERENCES

[1] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar, "Face swapping: automatically replacing faces in photographs," in *ACM Transactions on Graphics (TOG)*. ACM, 2008, vol. 27, p. 39.

[2] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] "Deepfakes," https://github.com/deepfakes/faceswap, Accessed: 2019-07-30.

[5] "Deepfacelab," https://github.com/iperov/DeepFaceLab, Accessed: 2019-07-30.

[6] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, "Mesonet: a compact facial video forgery detection network," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.

[7] Yuezun Li and Siwei Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, vol. 2, 2018.

[8] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "FaceForensics++: Learning to detect manipulated facial images," 2019.

[9] David Güera and Edward J Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[10] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *Interfaces (GUI)*, vol. 3, pp. 1, 2019.

[11] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[12] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister, "Video face replacement," in *ACM Transactions on Graphics (TOG)*. ACM, 2011, vol. 30, p. 130.

[13] Zhang Xingjie, Joongseok Song, and Jong-Il Park, "The image blending method for face swapping," in *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 2014, pp. 95–98.

[14] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3677–3685.

[15] Yuval Nirkin, Yosi Keller, and Tal Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7184–7193.

[16] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis, "Two-stream neural networks for tampered face detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1831–1839.

[17] Xin Yang, Yuezun Li, and Siwei Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8261–8265.

[18] Falko Matern, Christian Riess, and Marc Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 83–92.

[19] Yuezun Li, Ming-Ching Chang, and Siwei Lyu, "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking," *arXiv preprint arXiv:1806.02877*, 2018.

[20] Umur Aybars Ciftci and Ilke Demir, "Fakecatcher: Detection of synthetic portrait videos using biological signals," *arXiv preprint arXiv:1901.02212*, 2019.

[21] Davis E King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[22] Pavel Korshunov and Sébastien Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.

[23] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu, "Celeb-df: A new dataset for deepfake forensics," *arXiv preprint arXiv:1909.12962v2*, 2019.

[24] James A Hanley and Barbara J McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.