2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.01477

Exploring Temporal Coherence for More General Video Face Forgery Detection

Yinglin Zheng [1] Jianmin Bao [2], Dong Chen [2], Ming Zeng [1]*, Fang Wen [2]
[1] School of Informatics, Xiamen University
[2] Microsoft Research Asia
{zhengyinglin@stu., zengming@}xmu.edu.cn, {jianbao, doch, fangwen}@microsoft.com

Abstract

Although current face manipulation techniques achieve impressive performance regarding quality and controllability, they are struggling to generate temporal coherent face videos. In this work, we explore to take full advantage of the temporal coherence for video face forgery detection. To achieve this, we propose a novel end-to-end framework, which consists of two major stages. The first stage is a fully temporal convolution network (FTCN). The key insight of FTCN is to reduce the spatial convolution kernel size to 1, while maintaining the temporal convolution kernel size unchanged. We surprisingly find this special design can benefit the model for extracting the temporal features as well as improve the generalization capability. The second stage is a Temporal Transformer network, which aims to explore the long-term temporal coherence. The proposed framework is general and flexible, which can be directly trained from scratch without any pre-training models or external datasets. Extensive experiments show that our framework outperforms existing methods and remains effective when applied to detect new sorts of face forgery videos.

1. Introduction

With the development of deep generative models, especially Generative Adversarial Networks (GANs) [29, 40, 7, 38, 31]. Current face manipulation techniques [31, 47, 48, 49, 28, 53, 52] are capable of manipulating the attributes or even the identity of face images. These forged images are even difficult to distinguish by humans, and thus may be abused for spreading political propaganda, damaging our trust in online media. Therefore, detecting face forgery is of paramount importance.

Most previous methods [60, 61, 43, 45] are trained for known face manipulation techniques. But they experience a dramatic drop in performance when the manipulation methods are unseen. Some recent works [34, 57, 18, 59, 21, 42, 36, 11] have noticed this problem and attempted to boost the generalization. However, these methods are vulnerable to common perturbations such as image or video compression, noise, and so on. They still show limited generalization capability. A particularly effective work is Face X-ray [32], which proposes to detect blending artifacts instead of generative artifacts. However, the blending artifacts are typically low-level information that is susceptible to post-processing operations. More recent work LipForensics [22] proposes to detect unnatural mouth motion using spatio-temporal neural networks. But they only pay attention to the mouth which may ignore the artifacts in the other region of the face.

In this paper, we propose to leverage temporal coherence for more general face forgery detection. We observe that most face video forgeries are generated in a frame-by-frame manner. Since each altered face is generated independently, it inevitably leads to obvious flickering and discontinuity of

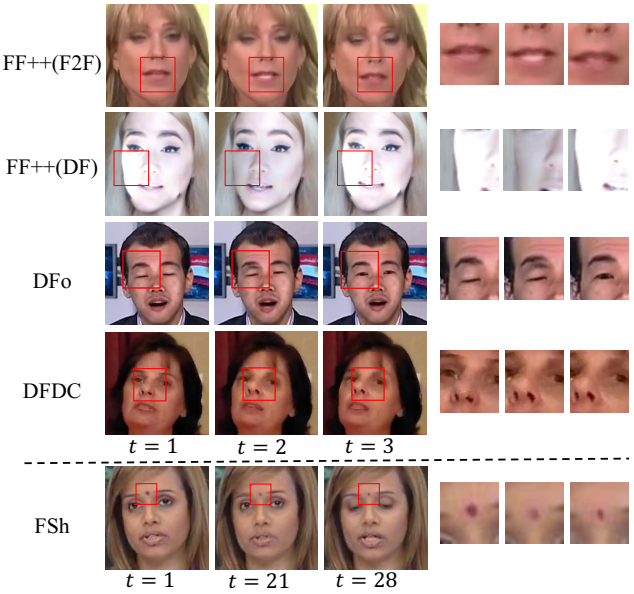

Figure 1. Temporal incoherence in existing datasets: FaceForensic++(FF++) [43], DeeperForensics(DFo) [28], Deepfake Detection Challenge Preview(DFDC) [16], and FaceShifter(FSh) [31]. In the top 4 rows, we show four temporal incoherence that happened between the neighborhood frames. In the last row, we show temporal incoherence happened between long-range frames.

*Corresponding author.

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.01477

the face area (see Figure 1). So we can leverage the temporal incoherence for more general and robust video face forgery detection. Previous works try to leverage spatio-temporal convolution network [21] or the recurrent neural network [5, 36] to learn temporal incoherence. However, we find that they all failed to learn the general temporal incoherence.

After careful investigation, we find that forged face videos mainly contain two types of artifacts, one is spatially related (*e.g*. blending boundary, checkboard, blur artifacts), the other is the temporal incoherence. Normally, the spatially related artifacts are more significant than the temporal incoherence. Without any specific design, current video face forgery detection methods [21, 5, 36] may rely more on spatial-related artifacts instead of the temporal incoherence for classification.

To encourage the spatio-temporal convolution network to learn the temporal incoherence, we redesign the convolution operator and propose a fully temporal convolution network (FTCN). The key idea is to restrict the network's capability for handling spatial-related artifacts. So we set the kernel size of all spatial (height and width) dimensions to 1 and keep the original kernel size of the time dimension in the 3D convolution operator. Due to the extremely low field for spatial dimension, the network learns to classify by temporal-related artifacts and hardly applies the spatial artifacts for detection. Also, we notice that even if the convolution operator is only time-dependent, its capability is sufficient to distinguish between real or fake.

Moreover, we find some discontinuity may happen in frames that are not in the neighborhood, for example, the wrinkles or moles of a face may gradually appear or disappear. To handle this issue, we propose to leverage Transformer [51] for capturing long-range dependencies along the time dimension. We add a light-weight Temporal Transformer after the proposed FTCN. The FTCN and the Temporal Transformer are trained end-to-end as the whole framework for general video face forgery detection.

Our approach is general and flexible. It can achieve impressive results without any pre-training knowledge or hand-crafted datasets. In contrast, previous work LipForensics [22] relies heavily on pre-training and Face X-ray relies on hand-crafted dataset. More importantly, without any manual annotations, our method can locate and visualize the temporal incoherence in the face forgery videos.

We conduct extensive experiments to compare its performance with the state-of-the-art in various challenging scenarios. We find that our method significantly outperforms previous methods in terms of generalization capability to unseen forgeries, and robust to various perturbations on videos. Furthermore, we perform ablation studies to validate the design choices of our framework.

Our contributions are summarized as follows:

- We explore to take full advantage of temporal coherence for face forgery detection and propose a framework that combined fully temporal convolution network (FTCN) and Temporal Transformer to explicitly detect temporal incoherence.
- Equipped with our detector, we can locate and visualize the temporal incoherence part of the face forgeries.
- Extensive experiments on various datasets demonstrate the superiority of our proposed methods with respect to generalization capability to unseen forgeries.

2. Related Work

Due to the emergence of high-fidelity face manipulation techniques, the detection of face forgery becomes an increasingly important research area. We will briefly introduce previous work on face forgery detection in this section.

Image Face Forgery Detection. Early studies put more emphasis on the spatial artifacts on the generated images, so they leverage the capability of convolution neural networks apply deep CNN models [3, 9, 25, 43] to train a binary classifier to distinguish real or fake. Meanwhile, a significant amount of works explores low-level image statistics(*e.g*. frequency, color) [42, 30, 19, 58, 20] or high-level semantics(*e.g*. identities) [41] of face images for forgery detection. Some recent studies [8, 4, 46, 14, 27, 56] aim to locate the visual artifacts in the forged images and make predictions based on the location results.

Video Face Forgery Detection. More recently, a great number of works start to take the temporal dimension into consideration and conducting face forgery detection at the video level. Li *et al*. [33] introduce leveraging eye blinking for detecting generated fake face videos. Amerini *et al*. [6] suggests using the optical flow between video frames. Mittal *et al*. [37] use the effective cues between audio and visual appearance for detecting fake videos. In contrast to these methods, our methods put more emphasis on general temporal incoherence. Unlike previous works, they either directly apply 3D convolution network [15, 21] on video or detect a specific kind of incoherence, such as irregular eye blinking [33], lip motions [22] or emotion [37]. In this work, we seek to detect general temporal incoherence, which can be any inconsistent region along time dimension.

Generalization to Unseen Manipulations. With the evolution of novel face manipulation techniques, many current detectors [43, 3] can experience a significant performance drop. Many works have noted this problem and presented some solutions for improving the generalization capability of detectors. FWA [34] explores to detect the resolution differences between altered face and background for general deepfake detection. LAE [18] and Multi-task [39] propose to learn a segmentation mask for the manipulated area in order to get a general detector. PatchForensics [11] suggests that the patch-based classifier can improve the per-

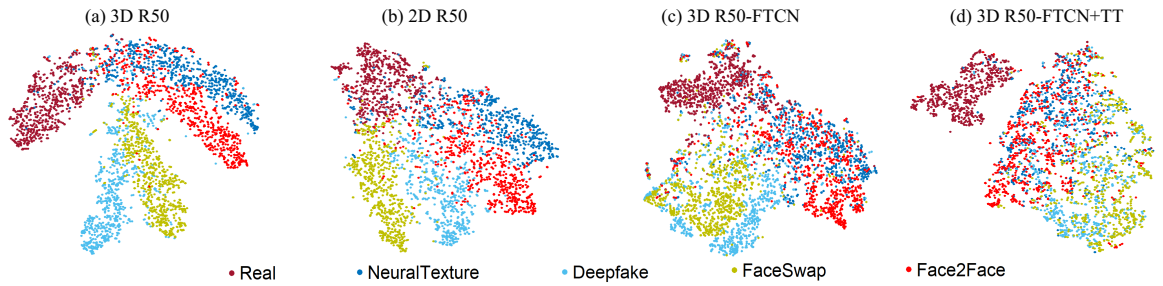

Figure 2. The t-SNE visualizations of features extracted from the last feature of different models on FF++ test set. Each dot represents the feature of a video clip. The t-SNE is computed with 40 for perplexity, 30 for PCA at 1000 iterations.

formance for generalized face forgery detection. Face X-ray [32] aims to improve the generalization capability by detecting the blending boundary artifacts instead of the artifacts from generative models. It can achieve impressive results in terms of generalization capability, but it is susceptible to many common perturbations. More recent LipForensics [22] indicates that many current deepfake techniques may suffer from unnatural mouth motions, so they apply spatial-temporal networks for the detection. However, they ignore the other regions in the face region, which may damage the performance.

3. Methods

3.1. Motivation

Currently, most face manipulation methods [31, 47, 48, 49, 53, 52] are specifically crafted at the image level. To generate a fake video, current techniques need to apply their methods for each frame independently. However, subtle changes in the appearance (*e.g*. noise, lighting, motion) often result in temporal incoherent results (*e.g*. flickering and discontinuous results as shown in Figure 1). Some prior face manipulation techniques [28, 35] have noticed this problem and apply post-processing tools to tackle this issue, but the generated videos still suffer from the problem of temporal incoherence to some extent. Therefore, how to detect temporal incoherence is worthy of more careful study.

Detecting the temporal incoherence is challenging since we do not have the location annotation of the incoherence in the video. A naive idea is to adopt the spatio-temporal convolution networks [21, 15] and expect the model to learn to distinguish real or fake by temporal incoherence. However, we find that forged face videos mainly contain two types of artifacts, one is the spatially related (*e.g*. blending boundary, checkboard, blur artifacts), the other is the temporal incoherence. Normally, the spatially related artifacts are more significant than the temporal incoherence. Without any specific design, the spatio-temporal convolution network distinguishes real or fake using spatial artifacts instead of temporal incoherence.

So the problem becomes how to encourage the spatio-temporal convolution network to learn the temporal incoherence. We take a fundamentally different approach, we propose a fully temporal convolution network. Concretely, we keep all the temporal-related convolution kernel size as the original but set all the spatial-related convolution kernel size to 1. We find this restriction can encourage the network to learn the temporal incoherence. To prove that, we take ResNet-50(R50) [24] as backbone and compare three types of classifiers:

1. A 3D R50 [23] network structure, which employs spatio-temporal convolutions.

2. A 2D R50 network structure, which uses 2D convolutions.

3. The proposed 3D R50-FTCN. We use 3D R50 as the backbone and set all the spatial-related convolution kernel size to 1 and keep the temporal-related kernel size.

To ensure fair comparison, all classifiers use the same training set FF++ [43] and the same training and inference settings. We show the t-SNE visualization of features extracted from different classifiers on FF++ test set in Figure 2. We have the following observations: Although all classifiers can distinguish between real and fake data, the distribution of fake data is completely different. Both the 3D R50 and 2D R50 will separate fake data generated by different face manipulation methods, even if we treat all fake data as one class in the training stage. It clearly shows that the features they extract contain the unique artifacts of each face manipulation algorithm. This would affect their generalization ability. On the contrary, the fake data of the 3D R50-FTCN classifier are more mixed together. It proves that the temporal network learns to classify by more general temporal incoherence.

On the other hand, some temporal incoherence exists in the *long-range* of video frames. However, prior studies [55, 26] indicate that temporal convolution struggles in dealing with the *long-range* dependencies. To tackle this issue, we add a Temporal Transformer(TT) [17] after the FTCN to detect the long-range temporal incoherence. The Temporal Transformer takes sequences of temporal features extracted by the FTCN as input and apply the class token to make the predictions. We also show the feature distri-

	layer	output size
$conv_1$	**5**×**1**×**1**, 64, stride 1, 1, 1	64×32×224×224
$pool_1$	1×5×5 max, stride 1, 4, 4	256×32×56×56
res_2	[1×1×1, 64; **3**×**1**×**1**, 64; 1×1×1, 256] ×3	256×32×56×56
$pool_2$	2×1×1 max, stride 2, 1, 1	256×16×56×56
res_3	[1×1×1, 128; **3**×**1**×**1**, 128; 1×1×1, 512] ×4	512×16×28×28
res_4	[1×1×1, 256; **3**×**1**×**1**, 256; 1×1×1, 1024] ×6	1024×16×14×14
res_5	[1×1×1, 512; **3**×**1**×**1**, 512; 1×1×1, 2048] ×3	2048×16×7×7
spatial-related average pool		2048×16×1×1

Table 1. Our 3D R50-FTCN model for video face forgery detection. Compared with the original 3D ResNet-50 model [10], we follow the rules described in Section 3.2.1 to obtain this structure. The dimension of 3D filter kernels are in $K_t \times K_h \times K_w$, The dimension of output maps are in $C \times N \times H \times W$. The input is 32×224×224. Residual blocks are shown in brackets.

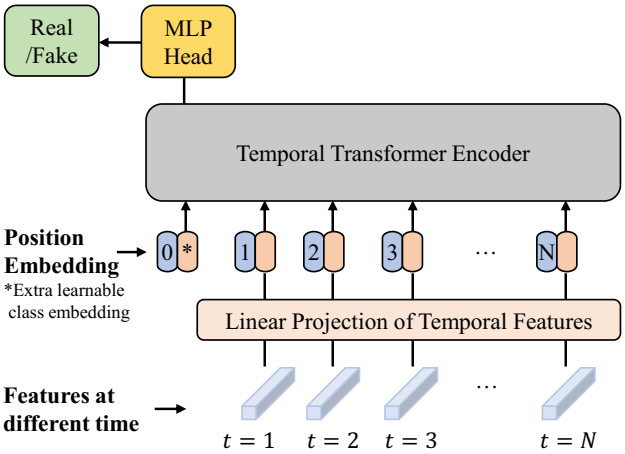

Figure 3. The Temporal Transformer for our video face forgery detection framework. We split the feature extracted from FTCN along the time dimension, and feed the resulting sequences of features to a standard Transformer encoder. We apply an extra learnable "classification token" to the sequence to learn the final discriminative feature and add an MLP head to distinguish real or fake.

bution of this classifier in Figure 2 (denoted as 3D R50-FTCN+TT). Using Transformer can further separate real and fake data, and can further gather the features of different face manipulation algorithms.

3.2. Overall Framework

In this section, we introduce the details of the proposed Fully Temporal Convolution network and Temporal Transformer. These two parts are trained end-to-end for video face forgery detection. Overall, given a suspect video $\mathbf{V}$, the first stage is the Fully Temporal Convolution Network (FTCN) that deals with local temporal flickering and inconsistency. It extracts temporal feature $\mathbf{F} = \mathbf{FTCN}(\mathbf{V})$. The second stage is the Temporal Transformer that aims to further model the long-term incoherence between each time slice of $\mathbf{F}$. Finally, an MLP head is used to do the final prediction.

3.2.1 Fully Temporal Convolution Network

3D CNNs [10, 50] are widely used on video-related tasks. Traditional 3D CNN models compute both spatial-temporal correlations via convolution with $K_t \times K_h \times K_w$ kernel, where $K_t \times K_h \times K_w$ are designed to be larger than 1 for most layers. These convolution layers are applied repeatedly, propagating signals progressively through the spatial and temporal dimensions. However, we find such spatial-temporal coupled kernels weaken the model's ability to capture purely temporal information. To encourage the spatio-temporal convolution network to learn the temporal incoherence, we redesign the convolution operator and propose a fully temporal convolution network (FTCN). The key idea is to restrict the capability of the network for handling spatial-related artifacts. So we set all the spatial sizes of the convolution kernel to 1. As we notice that some convolution layers may involve strides more than 1. In this situation, many locations may be ignored in the input features, thus we also design a rule to handle this. Suppose a 3D convolution is represented as 3DConv(K_t, K_h, K_w, S_t, S_h, S_w), where K_t, K_h, K_w are the kernel size for time, height, width dimension, S_t, S_h, S_w are the stride for time, height, width dimension. We replace it with 3DConv(K_t, 1, 1, 1, 1, 1) and add a max-pooling after the convolution operator if S_h or $S_w > 1$.

Take the popular 3D R50 structure [10] as an example, the resulting 3D R50-FTCN model is shown in Table 1. The input video clip has 32 frames each with 224×224 pixels. Also, we change the last global average pooling to spatial-related average pooling to keep the feature along time dimension unchanged. All the 3D convolutions in Table 1 share a similar kernel shape with like $K_t \times 1 \times 1$, which can be regarded as 1D temporal convolutional filters for temporal dimension. Such networks can be regarded as fully temporal convolution networks (FTCN), which mainly learn the discriminative features along the temporal dimension.

3.2.2 Temporal Transformer

The Temporal Transformer aims to learn the *long-range* discrepancies along the time dimension. With the FTCN, we obtain the temporal features $\mathbf{F} \in R^{C \times N \times H \times W}$ (C=2048, N=16, H=1, W=1). The temporal feature $\mathbf{F}$ can be represented as a sequence of features $\mathbf{F_t} \in R^C, t \in \{1, 2, ..., N\}$, which is a 1D sequence of token embeddings in standard Transformer [51]. The N is input sequence length, C is the feature dimension of sequence. The overview of our Temporal Transformer is depicted in Fig-

ure 3.

Similar to the settings in ViT [17], we apply a trainable liner projection $\mathbf{W}$ to map the feature dimension from C to D. To enable the classification in Temporal Transformer, we add a learnable embedding to the sequence of embedded features ($\mathbf{z_0^0} = \mathbf{F_{class}}$), which serves as the representative features learned on the input sequences. Following the settings in ViT[17], we also include learnable 1D position embeddings to retain positional information. Suppose the position embedding is $\mathbf{E_{pos}}$. So the input sequence z_0 for the Temporal Transformer can be defined as:

$$\mathbf{z}_0 = [\mathbf{F}_{\text{class}}, \mathbf{W}\mathbf{F}_1, \mathbf{W}\mathbf{F}_2, \cdots, \mathbf{W}\mathbf{F}_N]^T + \mathbf{E}_{pos}, \quad (1)$$

where $\mathbf{F}_t$ is the t-th time slice in feature $\mathbf{F}$, $\mathbf{W} \in \mathbb{R}^{D\times C}$, $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$.

The Temporal Transformer mainly consists of L standard Transformer Encoder blocks [51], each standard Transformer encoder block consists of a multi-head self-attention(MSA) [51] block and an MLP block. We also apply the commonly-used LayerNorm(LN) before each block. Another important structure is the residual connections [24], which is applied to each block. The activation function we used for the Temporal Transformer is GELU. So the features get for the l-th layer can be defined as:

$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \dots L \quad (2)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1 \dots L \quad (3)$$

So based on the class-token output $\text{LN}(\mathbf{z}_L^0)$ of the last encoder. We can apply an MLP head for the final fake probability:

$$y = \text{MLP}(\text{LN}(\mathbf{z}_L^0)). \quad (4)$$

In our experiments, the binary cross-entropy loss is applied on the final prediction y.

4. Experiments

4.1. Experimental Settings

Training Datasets. We adopt the most commonly used benchmark dataset FaceForensics++(FF++)[44] for training. It contains 1000 original videos and 4000 fake videos. The fake videos are manipulated by four methods: Face2Face(F2F) [49], FaceSwap(FS) [2], NeuralTexture(NT) [48], and Deepfake(DF) [1]. We trained on the high-quality (HQ) subset of the FF++, which is the light compression version.

Testing Datasets. To evaluate the generalization capability of our framework, we test our model on the following datasets: 1) FF++ that contains four types of manipulation as mentioned above; 2) FaceShifter[31](FSh) and 3) DeeperForensics[28](DFo) employ the real videos from FF++ for high-fidelity face swapping; 4) DeepFake Detection Challenge Preview dataset[16](DFDC), where each original video is filmed in the challenging environment; and 5) Celeb-DF-v2[35](CDF) is a new DeepFake dataset including 518 videos from different sources.

Evaluation Metrics. Following the evaluation metrics in previous works [32, 22], we report the area under the receiver operating characteristic curve (AUC). Since most previous works are image-based, following the setting in LipForensics [22], we report video-level AUC for a fair comparison. For the image-based method, we average the model predictions for each frame across the entire video. Therefore, all models utilize the same number of frames for classification.

Implementation. We take the 3D R50 as the basic structure of our proposed FTCN. The Temporal Transformer we use is one layer of standard Transformer Encoder [51], whose self-attention heads, hidden size, and MLP size are set to 12, 1024, 2048, respectively. For the training setting, we use a batch size of 32 and SGD optimizer with momentum, and the weight decay is set as 1e-4. We apply a warm-up strategy for the training of our methods. Concretely, the learning rate first increases from 0.01 to 0.1 in the first 10 epochs and then cosinely decayed to 0 for the last 90 epochs. For more details, please refer to the supplementary material.

Baselines. We mainly compare our methods with various state-of-the-art methods. These methods are mainly about boosting the generalization capability as well as some popular baselines. 1) **Xception** [43] explores the performance of face manipulation detection by the popular Xception [13] model. 2) **CNN-aug** [54] find that current CNN-generated images can be easily detected by a CNN model. 3) **Patch-Forensics** [11] suggests that the patch-based classifier can achieve impressive results for face forgery detection. 4) **Face X-ray** [32] aims to improve the generalization capability by detecting the blending boundary artifacts instead of the artifacts from generative models. 5) **CNN-GRU** [45] introduces GRU [12] into CNN model to model temporal coherence. 6) **Multi-task** [39] applies an autoencoder-like architecture for deepfake detection. 7) **FWA** [34] explores to detect the resolution differences between altered face and background for improving generalization capability. 8) **Two-branch** [36] presents multi-task learning on FaceForensics++ dataset. 9) **LipForensics** [22] is a recent work that studies the irregular mouth motions for general and robust face forgery detection.

4.2. Generalization to Unseen Manipulations

The differences between face forgery datasets mainly lie in the variations of source videos and face manipulation methods. To evaluate the cross-manipulations generalization capability of different face forgery detectors and prevent the possible bias introduced by different source videos, we conduct experiments on FF++, as it provides fake videos created by multiple face forgery methods for the same source videos. Following the setting in [22], we evaluate face forgery detectors with the leave-one-out strat-

Method	Train on remaining three				Avg
	DF	FS	F2F	NT	
Xception [43]	93.9	51.2	86.8	79.7	77.9
CNN-aug [54]	87.5	56.3	80.1	67.8	72.9
PatchForensics[11]	94.0	60.5	87.3	84.8	81.7
CNN-GRU [45]	97.6	47.6	85.8	86.6	79.4
Face X-ray[32]	99.5	93.2	94.5	92.5	94.9
LipForensics-Scratch[22]	93.0	56.7	98.8	98.3	86.7
LipForensics[22]	99.7	90.1	**99.7**	99.1	97.1
ours	**99.9**	**99.9**	**99.7**	**99.2**	**99.7**

Table 2. **Generalization to unseen manipulations.** We report the video-level AUC(%) on the FF++ dataset, which consists of four manipulation methods(DF, FS, F2F, NT). We train on three methods and test on the other one method. The results of other methods are from [22].

Model	#params	Pre-train	Extra data	Avg
Face X-ray[32]	65.8M	N	Y	94.9
LipForensics-Scrtach[22]	36.0M	N	N	86.7
LipForensics[22]	36.0M	Y	N	97.1
ours	**26.6M**	N	N	**99.7**

Table 3. **Comparison with state-of-the-art methods.** We report the number of parameters, pre-training, extra data usage, and the average video level AUC(%) on four unseen manipulation methods(DF, FS, F2F, NT), all the models are trained on the remaining three methods in FF++.

egy. To be concrete, as there are four types of fake video in FF++, each type is used once as a test set while the remaining three types form the training set. Both training and testing are conducted on the HQ version of FF++ dataset.

Table 2 shows that our method achieves excellent generalization(99.7%) to novel forgeries, surpassing on average most approaches by large margins. Although four types of manipulations(Deepfake, FaceSwap, Face2Face, NeuralTexture) in FF++ use different methods and focus on different tasks, our framework can learn a generalized discriminative feature on three of the manipulations and generalized to the remaining one. Our framework outperforms recent state-of-the-art methods Face X-ray [32] and LipForensics [22] by 4.8% and 2.6% in terms of AUC, respectively. We also present the number of parameters of Face X-ray and LipForensics on Table 3, our method achieves the highest performance with the minimal number of parameters, without any pre-training or external training data, which further demonstrate the superiority of our framework.

4.3. Generalization to Unseen Datasets

In a real-world scenario, suspicious videos are likely to be created by unseen methods from unseen source videos, thus the cross-dataset generalization would be crucial. To evaluate the cross-dataset generalization capability, we trained the face forgery detectors on all the four types of fake data in FF++, and perform the evaluation on four unseen datasets, including Celeb-DF-v2(CDF) [35], DFDC [16], FaceShifter [31] and DeeperFoensics [28]. As shown in Table 4, our model achieves the best performance on every dataset, with especially strong results on FaceShifter and DeeperForensics. On CDF and DFDC dataset, all the methods obtain relatively low scores, one possible explanation is the scenario gaps between different datasets.

Method	CDF	DFDC	FSh	DFo	Avg
Xception [43]	73.7	70.9	72.0	84.5	75.3
CNN-aug [54]	75.6	72.1	65.7	74.4	72.0
PatchForensics [11]	69.6	65.6	57.8	81.8	68.7
CNN-GRU [45]	69.8	68.9	80.8	74.1	73.4
Multi-task [39]	75.7	68.1	66.0	77.7	71.9
FWA [34]	69.5	67.3	65.5	50.2	63.1
Two-branch [36]	76.7	—	—	—	—
Face X-ray [32]	79.5	65.5	92.8	86.8	81.2
LipForensics [22]	82.4	73.5	97.1	97.6	87.7
ours	**86.9**	**74.0**	**98.8**	**98.8**	**89.6**

Table 4. **Generalization to unseen datasets.** We report the video-level AUC(%) on four unseen datasets: Celeb-DF-v2(CDF), Deepfake Detection Challenge Preview(DFDC), FaceShifter(FSh), and DeeperForensics(DFo). We train on FF++ and test on these unseen datasets. The results of other methods are from [22].

4.4. Robustness to Unseen Perturbations

For real-world scenarios, it is of great importance for face forgery detectors to be robust to unseen perturbations. We conduct experiments to verify the robustness of our methods. Following [28], we consider four popular perturbations: 1) Block-wise distortion; 2) Change of color saturation; 3) Gaussian Blur; 4) Resize: downsample the image by a factor then upsample it to the original resolution. Each perturbation is divided into five intensity levels as [28]. The results are reported in Figure 4. On average, our methods achieve better robustness to unseen perturbations. It is worth mentioning that our method does not apply any pre-training knowledge during training, which is obviously helpful for robustness.

4.5. Ablation study

We perform comprehensive studies on the FF++ [43] dataset to validate our design of the overall framework. We start by verifying the design of 3D R50-FTCN. **Why remove the spatial convolution?** To validate why we remove all the spatial convolution in the proposed 3D R50-FTCN, we construct multiple variants of 3D ResNet-50(3D R50), including the following models:

1. **3D R50**: The original model of 3D R50, with spatio-temporal convolutions.

2. **3D R50-Spatial**: Based on 3D R50, replace all the 3DConv(K_t, K_h, K_w, 1, 1, 1) with 3DConv(1, K_h, K_w, 1, 1, 1).

3. **3D R50-FTCN-FK3**: We replace the first 3D convolution layer of 3D R50-FTCN with 3DConv(5, 3, 3, 1, 1, 1), which involves spatial-related convolution.

4. **3D R50-FTCN-FK5**: We replace the first convolution of 3D R50-FTCN with 3DConv(5, 5, 5, 1, 1, 1),

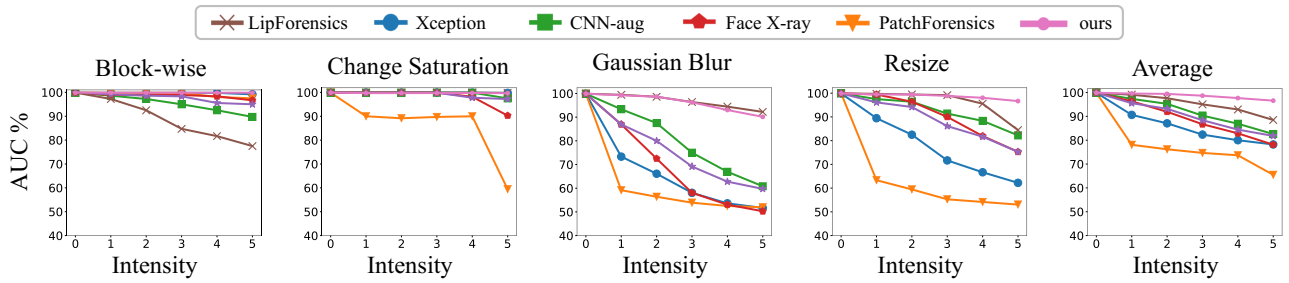

Figure 4. **Robustness to unseen perturbations.** We report the video-level AUC(%) of our methods under 5 different levels of four particular types of perturbations: Block-wise distortion, Change Saturation, Gaussian Blur, and Resize.

which involves more spatial-related convolution than **3D R50-FTCN-FK3**.

5. **3D R50-FTCN-Shuffle**: We use the same network structure as 3D R50-FTCN, but take a "spatial-shuffled" clip as input. The spatial-shuffle operation shuffles the pixel order in the clips along the spatial dimensions while keeping the pixel-wise temporal continuity (all images in the clip share the same shuffle pattern), we show the shuffled clips in the Supplementary Material.

6. **3D R50-FTCN**: The proposed FTCN, its structure is described in Section 3.2.1, and shown in Table 1.

We apply these models to train a binary classifier and evaluate the performance under the setting of training only on F2F, and testing on all four methods of FF++ (F2F, FS, DF, NT). All the variants are trained with exactly the same setting except the model architecture as described above. For each model, we report the performance of the best checkpoint, which is selected based on the average AUC among four methods on the validation set.

The results are shown in Table 5. We can identify some important conclusions from the results: 1) Compare the results of 3D R50 and 3D R50-Spatial, involving temporal information into face forgery detection can benefit the performance of generalization. 2) For 3D R50, 3D R50-FTCN-FK5, 3D R50-FTCN-FK3, 3D R50-FTCN, the spatial-related convolution involves less and less, but the generalization capability gets better and better, so less spatial-related convolution leads to better results. 3) Even if we damage the spatial information by pixel shuffle, the 3D R50-FTCN can still achieve a reasonable result, this suggests that the 3D R50-FTCN mainly learns to distinguish by temporal-related information.

Is limited model capability that benefits generalization capability? For 3D R50, 3D R50-FTCN-FK5, 3D R50-FTCN-FK3, 3D R50-FTCN, the spatial-related convolution involves less and less, and the model capability becomes weaker and weaker. Thus it is natural to ask whether the generalization capability benefits from the limited model capability. We conduct experiments to verify this. We design several variants of our proposed 3D R50-FTCN and trained on the F2F in FF++:

Model	Train on F2F				
	DF	FS	F2F	NT	Avg
3D R50	80.0	89.5	**100**	91.6	90.3
3D R50-Spatial	77.9	53.6	**100**	79.1	77.7
3D R50-FTCN-FK3	97.4	94.1	**100**	95.3	96.8
3D R50-FTCN-FK5	94.2	93.0	**100**	93.2	95.1
3D R50-FTCN-Shuffle	97.3	92.5	**100**	93.2	95.8
3D R50-FTCN	**98.0**	**95.9**	**100**	**96.0**	**97.5**

Table 5. **Ablation study of variants design of FTCN** Video-level AUC(%) is reported on FF++ datasets.

Model	Train on F2F				
	DF	FS	F2F	NT	Avg
3D R50	80.0	89.5	**100**	91.6	90.3
3D R50-SP	86.2	85.3	**100**	86.7	89.6
3D R50-FHCN	76.1	49.5	**100**	82.7	77.1
3D R50-FWCN	84.8	73.2	99.5	76.2	83.4
3D R50-FTCN	**98.0**	**95.9**	**100**	**96.0**	**97.5**

Table 6. **Ablation study of 3D R50 variants with different model capability.** Video-level AUC(%) is reported.

1. **3D R50-SP**: 3D R50 with the same amount of parameters as 3D R50-FTCN, this model is created by reducing the number of channels in 3D R50.

2. **3D R50-FHCN**: Replace 3DConv(K_t, K_h, K_w, S_h, S_w) with 3DConv(1, K_h, 1, 1, 1, 1) and add a MaxPool(1, S_h, S_w) if $S_h > 1$ or $S_w > 1$.

3. **3D R50-FWCN**: Replace 3DConv(K_t, K_h, K_w, S_h, S_w) with 3DConv(1, 1, K_w, 1, 1, 1) and add a MaxPool(1, S_h, S_w) if $S_h > 1$ or $S_w > 1$.

The results are reported in Table 6, 3D R50-SP and 3D R50-FTCN share a similar number of parameters, but 3D R50-FTCN presents better results. This validates that the performance gain is mainly from the fully temporal design. Moreover, 3D R50-FHCN and 3D R50-FWCN share exactly the same amount of parameters and computation cost with 3D R50-FTCN but present lower performance. This further indicates that temporal artifacts are more general as well as the effectiveness of our proposed 3D R50-FTCN.

Influence of video clip size. To find the optimal clip size, we trained 3D R50-FTCN with clip sizes of 8, 16, 32, 64. All the models are trained on F2F and test on all four methods in FF++. We change only the clip size and keep the

clip size	Train on F2F				
	DF	FS	F2F	NT	Avg
8	79	86.2	99.8	85.6	87.7
16	95.4	95.3	**100**	94.8	96.4
32	98.0	95.9	**100**	96.0	97.5
64	**98.2**	**96.6**	**100**	**96.7**	**97.9**

Table 7. **Ablation study of using different clip sizes for the training of FTCN.** Video-level AUC(%) is reported.

Model	Train on F2F				
	DF	FS	F2F	NT	Avg
3D R50 FTCN	98.0	95.9	**100**	96.0	97.5
3D R50 FTCN+TT L×1	**98.1**	**99.6**	**100**	**98.0**	**98.9**
3D R50 FTCN+TT L×2	97.8	98.6	**100**	97.7	98.5
3D R50 FTCN+TT L×3	97.0	98.4	**100**	97.4	98.2

Table 8. **Ablation study of using different layers of Transformer encoders in our framework.** Video-level AUC(%) is reported.

unrelated hyper-parameters unchanged.

Table 7 shows that as clip size increases, the performance boost. There is a tiny performance gain when clip size changes from 32 to 64, the possible reasons could be 1) There are not enough temporal convolution layers to capture such a long clip. 2) Video face alignment suffers from large clip size and large motion, as it would be hard to find a crop region that covers all the faces in the clip. As clip size growth brings performance boost along with more computation cost, a good trade-off would be clip size 32.

Effectiveness of Transformer. To validate the effectiveness of our Temporal Transformer, we perform an ablation study on the framework. We train three variants of our framework: 1) we train a model which only has the 3D R50 FTCN; 2) based on our framework, we change the number of encoder layers in Temporal Transformer to 2 (3D R50 FTCN+TT L×2); 3) based on our framework, we change the number of encoder layers in Temporal Transformer to 3 (3D R50 FTCN+TT L×3). The results are presented in Table 8. We can find the following observations: 1) the Temporal Transformer can improve the performance of generalization capability. 2) more layers of Standard encoder for Temporal Transformer can not further improve the performance, which shows that one layer of standard Transformer encoder is sufficient in our framework.

4.6. Localization of Temporal Incoherence

Without training with any explicit annotations, our method could be readily extended to localize temporally incoherent regions. In test time, for an input clip, we slide a window across the spatial domain. For regions outside the sliding window, we remove their content by replacing their RGB values with zeros. The modified clip is then fed into our forgery classifier to estimate the fake probability of the window area. Figure 5 shows that our method could robustly distinguish real and fake clips, and accurately localize the regions even with subtle temporal defects. For better visualization of temporal incoherence, please check the video results in the supplementary material.

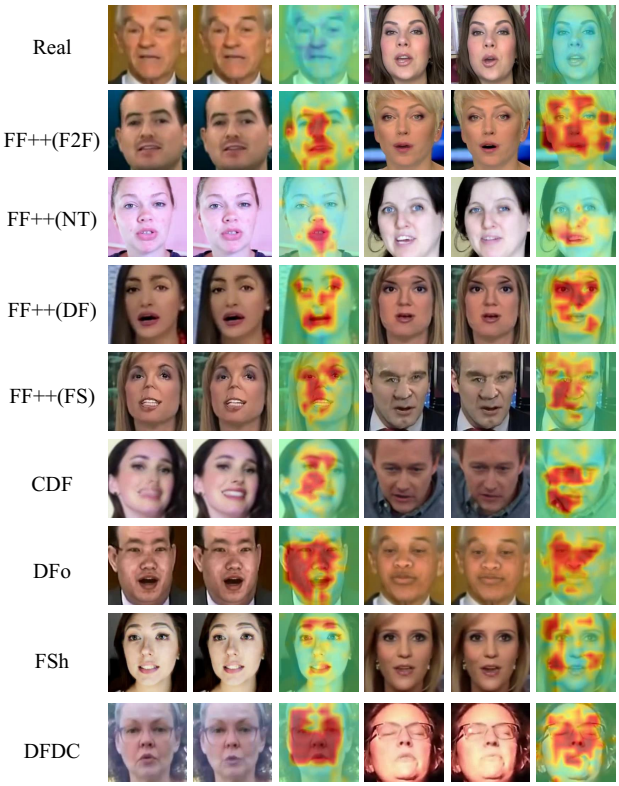

Figure 5. **Visualization of temporal defect localization on different datasets.** Each row shows two examples. For each example, the first two columns are consecutive frames in a video clip, the third column visualizes the localized defect regions, where warmer color indicates higher probability of forgery.

5. Conclusion

This paper investigates the effectiveness of temporal cues for more robust and general video face forgery detection. We propose to first encode short-term flickering with a Fully Temporal Convolution Network, then explore more subtle long-term incoherence with a Temporal Transformer. Extensive experiments evident the significant effects of the temporal information for video face forgery detection, and show the superior capabilities both on robustness and generalization of our proposed solution against previous methods. We hope our study will attract the community's attention to the temporal incoherence in deepfake detection.

6. Acknowledgements

Yinglin Zheng and Ming Zeng were partially supported by NSFC(No.62072382), Fundamental Research Funds for Central Universities, China(No.20720190003).

References

[1] Deepfakes. https://github.com/deepfakes/faceswap. [Accessed: 2020-09-02].

[2] Faceswap. https://github.com/MarekKowalski/FaceSwap. [Accessed: 2020-09-03].

[3] 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)*, pages 1–7. IEEE, 2018.

[4] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR Workshop*, pages 38–45, 2019.

[5] Irene Amerini and Roberto Caldelli. Exploiting prediction error inconsistencies through lstm-based classifiers to detect deepfake videos. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pages 97–102, 2020.

[6] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[7] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *CVPR*, pages 6713–6722, 2018.

[8] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019.

[9] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[11] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. *arXiv preprint arXiv:2008.10588*, 2020.

[12] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[13] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017.

[14] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *CVPR*, pages 5781–5790, 2020.

[15] Oscar de Lima, Sean Franklin, Shreshtha Basu, Blake Karwoski, and Annet George. Deepfake detection using spatiotemporal convolutional networks. *arXiv preprint arXiv:2006.14749*, 2020.

[16] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. Towards generalizable forgery detection with locality-aware autoencoder. *arXiv preprint arXiv:1909.05999*, 2019.

[19] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, pages 7890–7899, 2020.

[20] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. *arXiv preprint arXiv:2003.08685*, 2020.

[21] Ipek Ganiyusufoglu, L Minh Ngô, Nedko Savov, Sezer Karaoglu, and Theo Gevers. Spatio-temporal features for generalized detection of deepfake videos. *arXiv preprint arXiv:2010.11844*, 2020.

[22] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. *arXiv preprint arXiv:2012.07657*, 2020.

[23] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[25] Chih-Chung Hsu, Chia-Yen Lee, and Yi-Xiu Zhuang. Learning to detect fake face images in the wild. In *2018 International Symposium on Computer, Consumer and Control (IS3C)*, pages 388–391. IEEE, 2018.

[26] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, pages 3588–3597, 2018.

[27] Ashraful Islam, Chengjiang Long, Arslan Basharat, and Anthony Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, pages 2886–2895. IEEE, 2020.

[29] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *ICCV*, pages 3677–3685, 2017.

[30] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*, 2018.

[31] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2020.

[32] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *CVPR*, pages 5001–5010, 2020.

[33] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

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

[35] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, Seattle, WA, United States, 2020.

[36] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. *arXiv preprint arXiv:2008.03412*, 2020.

[37] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2823–2832, 2020.

[38] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447*, 2018.

[39] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2019.

[40] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *ICCV*, pages 7184–7193, 2019.

[41] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. Deepfake detection based on the discrepancy between the face and its context. *arXiv preprint arXiv:2008.12262*, 2020.

[42] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, pages 86–103. Springer, 2020.

[43] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, pages 1–11, 2019.

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

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

[46] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.

[47] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, pages 716–731. Springer, 2020.

[48] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[49] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, pages 2387–2395, 2016.

[50] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[52] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. End-to-end speech-driven realistic facial animation with temporal gans. In *CVPR Workshops*, pages 37–40, 2019.

[53] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *IJCV*, pages 1–16, 2019.

[54] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, volume 7, 2020.

[55] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[56] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[57] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. On the generalization of gan image forensics. In *Chinese Conference on Biometric Recognition*, pages 134–141. Springer, 2019.

[58] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2019.

[59] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning to recognize patch-

wise consistency for deepfake detection. *arXiv preprint arXiv:2012.09311*, 2020.

[60] 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)*, pages 1831–1839. IEEE, 2017.

[61] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *CVPR*, pages 1053–1061, 2018.