Learning Self-Consistency for Deepfake Detection

Tianchen Zhao* Xiang Xu Mingze Xu Hui Ding Yuanjun Xiong Wei Xia
Amazon/AWS AI
ericolon@umich.edu, {xiangx,xumingze,huidin,yuanjx,wxia}@amazon.com

Abstract

We propose a new method to detect deepfake images using the cue of the source feature inconsistency within the forged images. It is based on the hypothesis that images' distinct source features can be preserved and extracted after going through state-of-the-art deepfake generation processes. We introduce a novel representation learning approach, called pair-wise self-consistency learning (PCL), for training ConvNets to extract these source features and detect deepfake images. It is accompanied by a new image synthesis approach, called inconsistency image generator (I2G), to provide richly annotated training data for PCL. Experimental results on seven popular datasets show that our models improve averaged AUC over the state of the art from 96.45% to 98.05% in the in-dataset evaluation and from 86.03% to 92.18% in the cross-dataset evaluation.

1. Introduction

Deepfakes are synthetic media in which the identity or expression of a target subject is replaced by that of another source subject. They are predominantly generated by image stitching, which includes face detection, warping, and blending. Attacks using deepfakes have caused a significant amount of negative social impact, and also motivated methods to detect these forged videos. Most of these defense methods [60, 26, 37, 51, 46, 5, 23, 21, 6] target detecting suspicious artifacts left in the stitching process, such as eye blinking [26], face warping [27], blending boundaries [23], and fake prototypes [48]. In the wake of these defenders, forgery techniques are also evolving on reducing these artifacts to avoid detection, forming an enduring arms race.

In this paper, we propose a new method to detect deepfakes generated by stitching-based methods. Unlike other methods focusing on detecting artifacts described above, our approach uses the cue of *inconsistency of source features within the forged images*. Conceptually, images carry content-independent [14], spatially-local information that can uniquely identify their sources. We call them the *source features*. They could come from either imaging pipelines (*e.g.*, PRNU noise [30], specifications [14]), encoding approaches (*e.g.*, JPEG compression patterns [1], compression rates) or image synthesis models [58]. We hypothesize that these source features are still preserved after the modified image having gone through the state-of-the-art deepfake generation processes [28, 8, 15, 22, 39, 35]. Therefore, a forged image would contain different source features at different positions, whereas those of a pristine image must be consistent across all positions. By extracting the local source features and measuring their self-consistency, we can detect forged images.

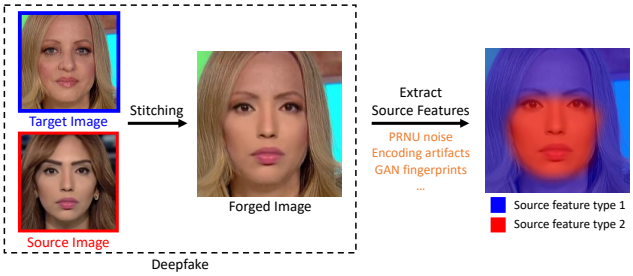

Figure 1: The forged image is generated by stitching target and source images. We hypothesize that each of them carries distinct source features that can uniquely identify their sources. Therefore, the forged image contains different source features at different locations, whereas those of a pristine image must be consistent across all positions. By extracting the local source features and measuring their self-consistency, we can detect forged images.

Specifically, we use a convolutional neural network (ConvNet) to extract source features in the form of downsampled feature maps. Each feature vector represents the source features of a corresponding location in the input image. To train this ConvNet, we introduce a novel representation learning method, called pair-wise self-consistency learning (PCL), which uses the consistency loss for supervision. We calculate the cosine similarity between very pairs of feature vectors in the source feature map, and compute the consistency loss on all pairs according to whether their

*Currently at University of Michigan, Ann Arbor. The work was conducted while at Amazon/AWS AI.

corresponding image locations come from the same source image. That is, we penalize the pairs that refer to locations from the same source image for having a low-similarity score and those from different source images for having a high-similarity score. We attach a non-linear binary classifier on the learned source feature map to perform the deepfake detection. We train it with an additional loss to produce the image-level real vs. fake labels.

The consistency loss in PCL needs pixel-level annotation about whether a location has been modified. It is generally not available in deepfake detection datasets, on which the re-annotation could be laborious and error-prone. We use synthesized data generated from inconsistency image generator (I2G) to tackle this issue. It generates forged images following the latest techniques in deepfake generation methods. To save computational cost and enable online generation, I2G only stitches together pristine source and target images instead of the synthesized ones from deep networks. We randomly sample the forgery mask for stitching during generation, which becomes the pixel-level annotation we need for PCL. Experimental results show that, although using a simplified generation process, the models learned with synthesized data from I2G effectively extract discriminative source features in both pristine and deepfake images.

We evaluate PCL on seven recent deepfake detection datasets and observe superior detection accuracy. Following the in-dataset evaluation, our method achieves the AUC scores of 99.79%, 99.98%, and 94.38% on FF++, CD2, and DFDC-P datasets, respectively. Because PCL uses the cue of source feature inconsistency which is less taken care of by current deepfake generation methods, we conjecture that a model trained with PCL on one dataset could effectively detect deepfakes generated by methods not seen in this dataset. To verify this, we adopt the cross-dataset evaluation protocol introduced in [23] to test our models and observe affirmative results. We achieve the AUC scores of 99.11%, 99.07%, 99.41%, 98.30%, and 90.03% on FF++, DFD, DFR, CD1, and CD2 datasets, respectively. We further visualize the consistency map of the learned source features on both real and fake images. We observe the consistency maps can lead to localization of the modified region.

It is worth noting that, as the race between forgers and defenders continues, the cue of source feature inconsistency can be negated. It can be done by either using entire face synthesis techniques [17, 29, 4] that directly output the whole fake image, such as GAN, or future development of stitching methods that completely removes or ambiguates the source feature. However, the state-of-the-art deepfake generation methods have not yet adopted these techniques. Thus the effectiveness of our method on detecting deepfake images should only be evaluated on the images generated by existing deepfake detection methods, as depicted in deepfake detection datasets we used [41, 10, 28, 8, 9, 15].

2. Related Work

Deepfake Generation. There are four common types of deepfake [47]: entire image synthesis, modification of facial attribute or expression, and face identity swap. 3D models [44], AutoEncoders [43, 49], or Generative Adversarial Networks [16, 17, 29, 4] are used to generate the fake image segment, which is then blended back to the original image.

Deepfake Detection. To detect the whole image synthesis, recent research [58, 36, 13, 52] observes that GAN-generated images contain specific cues that can be easily detected, and the trained models have exhibited good generalization ability across different synthesis methods. To support the research on detecting the other types of face manipulations, several deepfake datasets are released [57, 19, 41, 10, 28, 8, 15, 61] and countermeasures have been introduced. FakeSpotter [51] is proposed with a layer-wise neuron behavior for fake face detection. Recurrent neural networks [42] and various types of 3D ConvNets [6] are utilized to detect the manipulation artifacts across the video frames. However, binary classifiers are criticized for their interpretability, and several localization methods are introduced through either multi-task learning [37] or attention-based mechanisms [5, 59]. To improve the generalization ability, DSP-FWA [27] and Face X-ray [23] also make their data generation pipeline and the latter focuses on predicting the blending boundaries in fake video frames.

Our approach also lies in this line but has several key differences. First, from the methodology perspective, we focus on detecting deepfakes by using a less attended cue of inconsistency of source features within the forged images. Second, from the network design perspective, our consistency predictor only contains a few parameters and can serve as a plugin module upon any common backbones.

Consistency Learning. The concept of inconsistency has been studied in the image forensic literature [34], where similarity scores are computed among image patches [34, 33, 14, 32, 60, 2]. Zhou *et al*. [60] propose a two-stream network to detect both tampered faces and low-level inconsistencies, but the training requires steganalysis feature extraction. Huh *et al*. [14] use a Siamese network to predict the metadata inconsistency by iteratively comparing random patches from different raw images. Nirkin *et al*. [38] use signals from the proposed face identification and context recognition networks to detect deepfakes.

In this paper, we introduce consistency learning to deepfake detection and propose an end-to-end learning architecture that estimates the image self-consistency with one forward pass, while capturing the internal relations among patches within an image. Besides, instead of using only raw images, we design I2G to address several challenges to fit face forgery detection better by supplying PCL with training images that are finely stitched from multiple sources.

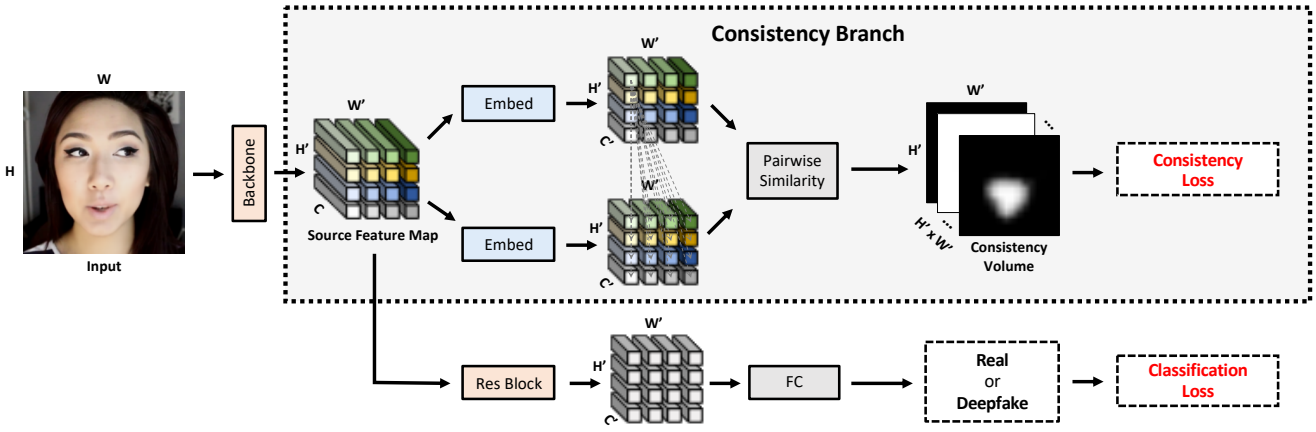

Figure 2: *Visualization of PCL architecture.* The consistency branch focuses on measuring the consistencies of image patches according to their source features. A classification branch is applied after the source feature map and predicts the binary score for deepfake detection.

3. Our Approach

Given an input image, our goal is to detect if the identity or expression of the subject is replaced with that of another subject. Observing that deepfakes are stitched by images from different sources with distinct source features, we explore learning effective and robust representations for deepfake detection by measuring the source feature consistency within the image. More specifically, we propose a multi-task learning architecture, as shown in Fig. 2. The consistency branch is optimized to predict a consistency map for each image patch, indicating its source feature consistencies with all others. The classification branch is applied to the source features and outputs binary labels for evaluation purposes. The model is trained on both consistency and classification loss, with annotations supplied by I2G.

3.1. Pair-Wise Self-Consistency Learning (PCL)

The consistency branch computes the pairwise similarity scores of all possible pairs of local patches in an image and predicts a 4D consistency volume $\widehat{\mathbf{V}}$. Given a pre-processed video frame X of size $H{\times}W{\times}3$ as input, we first feed it into the backbone and extract the source feature F of size $H'{\times}W'{\times}C$ from an intermediate convolution layer, where H', W', and C are height, width, and channel size, respectively. For each patch $\mathcal{P}_{h,w}$ in the source feature map, we compare it against all the rest to measure their feature similarities, and obtain a 2D consistency map $\widehat{M}^{\mathcal{P}_{h,w}}$ of size $H'{\times}W'$ of consistency scores in the range of $[0, 1]$, where the superscript indicates the position of the base patch. To be specific, for any pair of patches $\mathcal{P}_i$ and $\mathcal{P}_j$, we compute their dot-product similarity [53] using their extracted feature vector f_i and f_j, both of size C, to estimate their consistency score:

$$s(f_i, f_j) = \sigma\left(\frac{\theta(f_i)\theta(f_j)}{\sqrt{C'}}\right), \tag{1}$$

where θ is the embedding function, realized by 1×1 convolutions, C' is the embedding dimension, and σ is the Sigmoid function. We iterate this process over all patches $\{\mathcal{P}_{h,w} | 1 \le h \le H', 1 \le w \le W'\}$ in the source feature map, and finally get the 4D consistency volume $\widehat{\mathbf{V}}$ of size $H'{\times}W'{\times}H'{\times}W'$. To provide visualization clues about the region of modification, we fuse the 4D consistency volume $\widehat{\mathbf{V}}$ over all patches and generate a 2D global heatmap $\widehat{\mathcal{M}}$ of size $H'{\times}W'$ (described in the Appendix). We up-sample $\widehat{\mathcal{M}}$ to $\widehat{\mathbb{M}}$ of size $H{\times}W$ to match the input size for visualization.

The optimization of consistency branch requires the 4D "ground truth" consistency volume $\mathbf{V}$. Given the mask $\mathcal{M}$ of size $H{\times}W$ indicating the manipulated region of input X, we first create its coarse version $\mathcal{M}$ matching the size of $H'{\times}W'$ through bi-linear down-sampling. We obtain the ground truth 2D consistency map $M^{\mathcal{P}_{h,w}}$ for the (h, w)-th patch by computing the element-wise difference between its own value $\mathcal{M}_{h,w}$ and all others,

$$M^{\mathcal{P}_{h,w}} = 1 - |\mathcal{M}_{h,w} - \mathcal{M}|, \tag{2}$$

where $\mathcal{M}_{h,w}$ is the scalar value in position (h, w), and $M^{\mathcal{P}_{h,w}}$ is in size of $H'{\times}W'$. For each entry of $M^{\mathcal{P}_{h,w}}$, a value close to 1 denotes the two patches are consistent, and close to 0 otherwise. To obtain the ground truth 4D global map $\mathbf{V}$, we compute $M^{\mathcal{P}_{h,w}}$ for all patches. Note that $\mathbf{V}$ of a pristine image should be $\mathbf{1}$, a 4D volume in which all values are equal or close to one.

We use the binary cross-entropy (BCE) loss to supervise the consistency prediction over the 4D consistency volume $\widehat{\mathbf{V}}$, and more formally,

$$\mathcal{L}_{PCL} = \frac{1}{N} \sum_{h,w,h',w'} \text{BCE}(\mathbf{V}_{h,w,h',w'}, \widehat{\mathbf{V}}_{h,w,h',w'}), \tag{3}$$

where h and $h' \in \{1, 2, ..., H'\}$, w and $w' \in \{1, 2, ..., W'\}$, and N equals to $H'{\times}W'{\times}H'{\times}W'$.

The consistency branch learns the representations that predict the self-consistency of the input according to their source features, which by our claim could significantly benefit the deepfake detection in both effectiveness and robustness. Nevertheless, these features cannot directly make inferences for evaluation purposes.

The classification branch is thus applied after the source feature map to predict if the input is real or fake. More specifically, the extracted source feature is fed into another convolution operation. A global average pooling and fully-connected layer are built after that as the classifier, which outputs the probability score for the input of being real or fake. We use the two-class cross-entropy (CE) loss $\mathcal{L}_{CLS}$ to supervise the training in the classification branch.

The overall loss function of our model is as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{PCL} + \mathcal{L}_{CLS}, \tag{4}$$

with hyper-parameter λ. The ablation study in Section 4.5 suggests that a choice of large λ value significantly improves the performance. This observation demonstrates that the representations learned from the consistency branch play a dominant role in the success.

3.2. Inconsistency Image Generator (I2G)

Training PCL requires patch-level annotations of the manipulated regions, which is not always available in current existing datasets. To provide this training data, we propose *inconsistency image generator (I2G)* to generate "self-inconsistent" images from the pristine ones, along with the ground truth masks $\mathcal{M}$ discussed in Section 3.1. To guarantee sufficient amount and diversity of the training data with least efforts, I2G reduces the computational cost by replacing facial image synthesis using the GAN or VAE [28, 8, 15, 22] with real images. It follows that I2G can support dynamic data generation on CPU during the training and be utilized as a part of the data augmentation for deepfake detection.

Similar self-supervised approaches [27, 23, 35, 8] have been studied in other tasks or methods for deepfake detection. I2G particularly addresses several challenges to fit PCL better. **First**, because face images have some strong structural bias, stitching with the face hull regions may create undesired correlations between the source feature inconsistency and the face boundary. I2G uses elastic deformation [40] to improve the variety of the mask $\mathcal{M}$ and thereby eliminates those spurious correlations. **Second**, because attackers will intentionally try to remove source features to make deepfake images more realistic, PCL needs to make use of source features that are not vulnerable to these approaches. I2G randomly selects one from an exhaustive set of blending methods in data generation so the representation learned by PCL can be robust to source feature removal attempts. **Third**, we expect the learned representation to generalize to a wide range of sources, even unseen ones during training. I2G adds image augmentation to the generation process to achieve this goal. The augmentation methods include JPEG compression, Gaussian noise/blur, brightness contrast, random erasing, and color jittering.

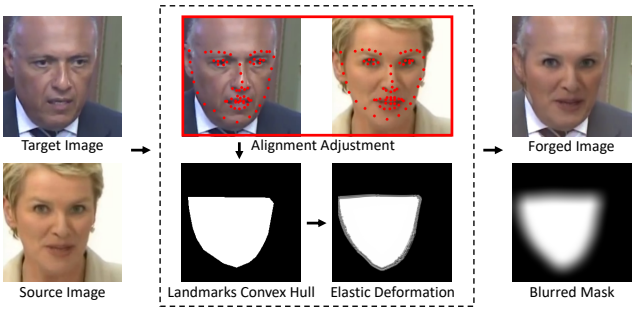

Figure 3: *Illustration of the workflow of I2G.* For each source and target image pair, a morphed mask is generated by taking a convex hull of the landmarks followed by elastic deformation and Gaussian blur. The masked region of the target image is replaced by that of the source with blending techniques [39, 35].

Algorithm 1 Inconsistency Image Generator (I2G)

Input: Target video frame X^t of size $(H, W, 3)$.
Output: Generated video frame X^g and mask $\mathcal{M}$.
Landmark Detector $\mathcal{K} : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{68\times 2}$.

1: Get the video frame X^t and its landmarks $\mathcal{K}(X^t)$.
2: Find a random source frame X^s of different ID, which satisfies $\|\mathcal{K}(X^t) - \mathcal{K}(X^s)\|_2 < \epsilon$ for threshold $\epsilon > 0$.
3: Align X^s to X^t using landmarks.
4: Compute convex hull $\mathcal{H}$ of $\mathcal{K}(X^t)$.
5: Get mask $\mathcal{M}$ by elastic deforming and blurring $\mathcal{H}$.
6: Get X^g by blending X^t with X^s with $\mathcal{M}$.

The workflow of I2G is summarized in Alg. 1 and illustrated in Fig. 3. Given a target video frame X^t, we take its 68-point facial landmarks and retrieve another frame from different videos with different identities, so that the faces in the two frames have similar landmarks measured in ℓ_2 norm. For a pair of images, we first align their faces with the pre-computed landmarks and then detect the facial region by taking the convex hull of the landmarks. Elastic deformation [40] is also employed to morph the convex hull: we generate the smooth deformations using random displacement vectors sampled from a Gaussian distribution with a standard deviation of 6 to 12 pixels on a coarse 4 by 4 grid, and compute per-pixel displacements using bi-cubic interpolation. The deformed mask is further blurred by a Gaussian kernel of size 16. Finally, the facial region of the source frame within the mask is stitched to the target frame using various blending methods [39, 35]. I2G outputs a forged video frame and the corresponding mask $\mathcal{M}$.

4. Experiments

We evaluated the performance of our approach (PCL + I2G) against multiple state-of-the-art methods on seven publicly-available datasets. First, we showed that our model achieves convincing performance under the in-dataset setting, where training and testing are conducted on the same dataset. To demonstrate the superior generalization ability of our model, we conducted the cross-dataset evaluation by training the model with only I2G-augmented real videos and testing on unseen datasets. The ablation studies explored the contribution of each component in our model, such as the effect of PCL and I2G.

4.1. Implementation Details

Pre-processing. For each raw video frame, face crops are detected and tracked by using [24] and landmarks are detected by public toolbox [3]. We normalize all face crops with ImageNet mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$, and resize them to 256×256. We also use standard data augmentations, including JPEG compression, Gaussian noise/blur, brightness contrast, random erasing, and color jittering.

Network Architecture. We adopt ResNet-34 [12] as backbone and initialized with pretrained weights on ImageNet [7]. Given a pre-processed video frame X of size H×W×3, we first feed it into the backbone and extract the features F after the `conv3` layer of size H'×W'×256, where H'=$H/16$ and W'=$W/16$. Here each patch corresponds to a 16×16 region in the original image.

Training. For each epoch, we randomly sample 32 frames from every video, and the total number of training samples from K videos is 32×K. The model is trained for 150 epochs using Adam optimizer [18] with batch size 128, betas 0.9 and 0.999, and epsilon 10^{-8}. The learning rate is linearly increased from 0 to 5×10^{-5} in the first quarter of the training iterations and is decayed to zero in the last quarter. The hyper-parameter λ is set to be 10 by default.

4.2. Settings

Training Data. Each of our training samples is of the form $(X, \mathbf{V}, y)$, where X is the input video frame, $\mathbf{V}$ is the ground truth 4D consistency volume, and y is the binary label. For a real frame, $\mathbf{V}$ is a 4D tensor of ones, indicating the image is self-consistent, and y is zero. There are two types of forged samples in our settings. The first type is the deepfake video frame from the existing deepfake datasets, for which we find the corresponding real video frame and compute the structural dissimilarity (DSSIM) [55] between them. The mask $\mathcal{M}$ is then generated by taking a Gaussian blur of DSSIM followed by thresholding. $\mathbf{V}$ is computed from $\mathcal{M}$ by Eq. 2 and y is one. The second type is the fake image **augmented by I2G** on real images, where $\mathbf{V}$ is computed with the mask from I2G and y is one. By training with I2G-augmented datasets, whenever a real data $(X^t, \mathbf{1}, 0)$ is sampled during the training, there is a 50% chance that it is dynamically transformed by I2G into a fake data $(X^g, \mathbf{V}, 1)$ where X^g and $\mathcal{M}$ are the outputs of I2G as described in Alg. 1 and $\mathbf{V}$ is computed with $\mathcal{M}$ by Eq. 2.

More specifically, in the **in-dataset experiments**, our train set includes both the pristine and deepfake videos from the train split of the dataset (to be evaluated). Unless otherwise noted, we also utilize the fake samples augmented by I2G as data augmentation. In the **cross-dataset experiments**, we follow prior work [23] and train with *only real videos* from the raw version of FaceForensics++ (FF++) [41], augmented by I2G. Note that the cross-dataset setting is more close to the real-world scenarios where the potential attack types are not aware.

Test Data. FaceForensics++ (FF++) [41] is by far the most popular benchmark for deepfake detection. Its raw version contains 700 videos for testing, including 140 pristine and 560 fake videos from 4 different algorithms, which are Deepfakes (DF) [11], Face2Face (F2F) [44], FaceSwap (FS) [20] and NeuralTextures (NT) [45]. DeepfakeDetection (DFD) [10] dataset is released incorporated with FF++, supporting the deepfake detection research. Celeb-DF-v1 (CD1) & -v2 (CD2) [28] datasets consist of high-quality forged celebrity videos using advanced synthesis process. Deepfake Detection Challenge (DFDC) [8] public test set is released for the Deepfake Detection Challenge, and DFDC Preview (DFDC-P) [9] is its preliminary version. DFDC and DFDC-P contain many extremely low-quality videos, making them exceptionally challenging. DeeperForensics-1.0 (DFR) [15] modifies the pristine videos in FF++ with new face IDs and more advanced techniques. More detailed statistics are provided in the Appendix.

Evaluation Metrics. We report the deepfake detection results with the most commonly used metrics in the literature, including the area under the ROC curve (AUC) and average precision (AP). A higher AUC or AP value indicates better performance. To provide a comprehensive benchmark for future work, we report our performance on all datasets in terms of AUC, AP, as well as Equal Error Rate (EER) in the Appendix. Unless otherwise noted, the evaluation results in the experiments are at video-level, computed by averaging the classification scores of the video frames.

4.3. In-Dataset Evaluation

In-dataset evaluation is abundantly adopted in the literature, where the focus is on specialization but not generalization. To compare against the existing work, we consider three of the most popular datasets, which are FF++, CD2, and DFDC-P. Given a dataset, our model is trained on both real and deepfake data from train split, and performance is

Method	Backbone	Train Set	Test Set (AUC (%))				
			DF	F2F	FS	NT	FF++
MIL [54]	Xception	FF++	99.51	98.59	94.86	97.96	97.73
Fakespotter [51]	ResNet-50	FF++, CD2, DFDC	-	-	-	-	98.50
XN-avg [41]	Xception	FF++	99.38	99.53	99.36	97.29	98.89
Face X-ray [23]	HRNet	FF++	99.12	99.31	99.09	99.27	99.20
S-MIL-T [25]	Xception	FF++	99.84	99.34	99.61	98.85	99.41
PCL + I2G	ResNet-34	FF++	**100.00**	**99.57**	**100.00**	**99.58**	**99.79**

Table 1: *In-dataset evaluation results on FF++.* Our method performs better on all manipulation types with a smaller backbone.

Method	Backbone	Train Set	Test Set (AUC (%))
			CD2
Fakespotter [51]	ResNet-50	CD2	66.80
Tolosana *et al*. [46]	Xception	CD2	83.60
S-MIL-T [25]	Xception	CD2	98.84
PCL + I2G	ResNet-34	CD2	**99.98**

Table 2: *In-dataset evaluation results on CD2.* We achieve saturated performance in terms of AUC.

Method	Backbone	Train Set	Test Set (AUC (%))
			DFDC-P
Tolosana *et al*. [46]	Xception	DFDC-P	91.10
S-MIL-T [25]	Xception	DFDC-P	85.11
PCL + I2G	ResNet-34	DFDC-P	**94.38**

Table 3: *In-dataset evaluation results on DFDC-P.* Our method improves the best existing result by 3.28% in terms of AUC.

evaluated with the corresponding test set.

The results for FF++, CD2, DFDC-P are shown in Table 1, Table 2, Table 3, respectively. On average, compared to the state of the art, our approach improves the AUC score on these three datasets, from 96.45% to 98.05%. Our models achieve the state of the art with near-perfect performance on CD2 (99.98%) and all four manipulations of FF++ (100.00% on DF, 99.57% on F2F, 100.00% on FS, 99.58% on NT), surpassing all existing work. For DFDC-P, our model outperforms the state-of-the-art result by 3.28% in terms of AUC score. Note that the results reported for DFDC-P are comparably lower, due to the fact that a non-negligible portion of the dataset is of extremely low quality, *e.g.*, human faces in some videos are hardly recognizable. We also compare with prior work in terms of frame-level AUC on CD2, and outperform the state of the art [31] by 8% (see Appendix for more details).

4.4. Cross-Dataset Evaluation

The generalization ability is an important indicator of the superiority of an algorithm. In the real world, the defense method cannot get any prior knowledge of the attacks. The cross-dataset evaluation is a widely-used approach to evaluate the generalization ability of an algorithm.

Table 4 presents the cross-dataset evaluation results on FF++ and DFD, where we only used the real videos of FF++ for training. The performance of our model is on par with Face X-ray [23] on FF++, achieving convincing results with more than 99.00% in terms of AUC. Interestingly, the performance gap between our model and Face X-ray [23] is much larger (99.07% vs. 93.47%) on DFD. It is possible that the test data of FF++ is highly correlated with its training data, since they are very likely to be collected from the same source, whereas the correlation disappears in DFD. The results demonstrate that predicting the source feature consistency can effectively generalize across different source cues, without overfitting to any spurious correlation among data from the same generation method.

We further evaluate our model on five more advanced datasets, as shown in Table 5. In particular, our model outperforms the state of the art on CD1 and CD2, by about 18.00% and 13.00% in terms of AUC, and provides pioneering cross-dataset baselines on DFR (99.51%) and DFDC (67.52%). On DFDC-P, our performance is comparable with Face X-ray [23], where we get a lower AUC but a higher AP score, as shown in Table 6. We compute the average AUC score among five out of seven datasets (except DFR and DFDC that has no published benchmarks) and find that our model's performance outperforms the state of the art (92.18% vs. 86.03%). Meanwhile, we observe that both our model and state-of-art-art methods cannot achieve appealing results on DFDC/DFDC-P datasets, which motivates us to do failure analysis in Section 4.7 and 4.6.

4.5. Ablation Studies

Effect of PCL. We use λ to balance between the consistency and classification losses, as shown in Eq. 4. By setting $\lambda = 0$, we disable PCL and get a network architecture equivalent to vanilla ResNet-34 with binary classification loss. To exhibit the advantage of our consistency loss, we train models with increasing λs and evaluate their cross-dataset generalizations with four test sets. As shown in Table 7, we follow the cross-dataset setting and train all models on real data of FF++ augmented by I2G and report the AUC score for performance comparison. We observe that training with $\lambda > 0$ significantly outperforms the training with $\lambda = 0$. Especially, the performance on DFDC is im-

Method	Backbone	Train Set	Test Set (AUC (%))					
			DF	F2F	FS	NT	FF++	DFD
Face X-ray [23]	HRNet	FF++ (real data)	99.17	98.57	98.21	**98.13**	98.52	93.47
PCL + I2G	ResNet-34	FF++ (real data)	**100.00**	**98.97**	**99.86**	97.63	**99.11**	**99.07**

Table 4: *Cross-dataset evaluation results on FF++ and DFD.* Our model is on par with Face X-ray [23] on FF++, but has better performance on DFD by 5.67% in terms of AUC, with fewer network parameters.

Method	Backbone	Train Set	Test Set (AUC (%))				
			DFR	CD1	CD2	DFDC	DFDC-P
Dang *et al*. [5]	Xception + Reg.	UADFV [57], DFFD [5]	-	71.20	-	-	-
DSP-FWA [27]	ResNet-50	FF++	-	-	69.30	-	-
Xception [41]	Xception	FF++	-	-	73.04	-	-
Masi *et al*. [31]	LSTM	FF++	-	-	76.65	-	-
Face X-ray [23]	HRNet	FF++	-	80.58	-	-	**80.92**
PCL + I2G	ResNet-34	FF++ (real data)	**99.41**	**98.30**	**90.03**	**67.52**	74.37

Table 5: *Cross-dataset evaluation results on DFR, CD1, CD2, DFDC and DFDC-P datasets.* Our model out-performs the state of the art on CD1 and CD2, by about 18.00% and 13.00% in terms of AUC, and provides pioneering cross-dataset baselines on DFR (99.51%) and DFDC (67.52%). For DFDC-P, we have a lower AUC score but a higher AP score in comparison with the state of the art (see Table 6).

Method	Backbone	Train Set	Test Set (AP (%))	
			CD1	DFDC-P
Face X-ray [23]	HRNet	FF++	73.33	72.65
PCL + I2G	ResNet-34	FF++ (real data)	**98.97**	**82.94**

Table 6: *Cross-dataset evaluation results on CD1 and DFDC-P datasets.* Our models can identify the attack video more precisely.

proved by 15.8% in terms of AUC. The results validate that it is beneficial to use large λ during training, which also suggests that PCL plays a dominant role in the success.

Effect of I2G as Joint-Training. We have been training with the consistency loss on datasets augmented by I2G. I2G generates fake data dynamically, enhancing the training data variety, thereby improving the performance and generalization. To demonstrate the effect of I2G , we conduct ablation studies by training on either DF or DFDC-P and benchmarking on DFR, CD2, DFDC, and DFDC-P test sets. Table 8 shows that models trained on data augmented by I2G outperform the straightforward combination of routine data augmentations and blending methods used in the baseline. In particular, we train models on the DF with or without I2G, whereas the performance of the latter is improved by an average AUC score of 7.18% over four test sets. The model trained with the DFDC-P train set has noticeably better performance on DFDC and DFDC-P test sets comparing to previous models, but generalizes poorly on other datasets such as DFR. I2G improves model's generalization, for example, performance on DFR is raised from 51.61% to 92.25% in terms of AUC, with a minor sacrifice to the performance on DFDC-P.

Effect of I2G as Pre-Training. When even computing the DSSIM masks of deepfakes is not feasible, one can always use our consistency loss to *pretrain* the model with any real data augmented by I2G. After that, any standard training for deepfake detection or related tasks can be conducted with the whole dataset. In particular, we conducted an experiment where we first pretrained a ResNet-34 on the real data of DF augmented by I2G for consistency prediction, and finetuned with both real and fake data of DF for classification. We report the evaluation results on DFR, CD2, DFDC, DFDC-P with 99.57%, 91.88%, 68.95%, 79.17% (84.89% on average) in terms of AUC, respectively. These results are unsurprisingly lower than those of joint-training, but still significantly outperform the baseline (79.19% on average). Besides, our pretrained models may not necessarily be trained from the established deepfake datasets. I2G can be potentially applied to any face image or video datasets, such as IMDb-Face [50] and YouTube Faces [56], providing a stronger pretrained model for deepfake-related research.

Choice of Patch Size. We evaluate the effectiveness of using different patch sizes. Conceptually, larger patches get coarser consistency maps that may reduce their efficacy on forgery detection, while smaller patches may not contain enough information of source features and induce extra computation cost. In particular, we evaluate our cross-dataset model from Table 4 with the patch size of 4×4, 8×8, 16×16, and 32×32 on FF++, and get 98.32%, 98.35%, 99.11%, and 98.74% in terms of AUC, respectively.

4.6. Qualitative Results

PCL not only improves the representation learning for deepfake detection, but also can be used to generate the in-

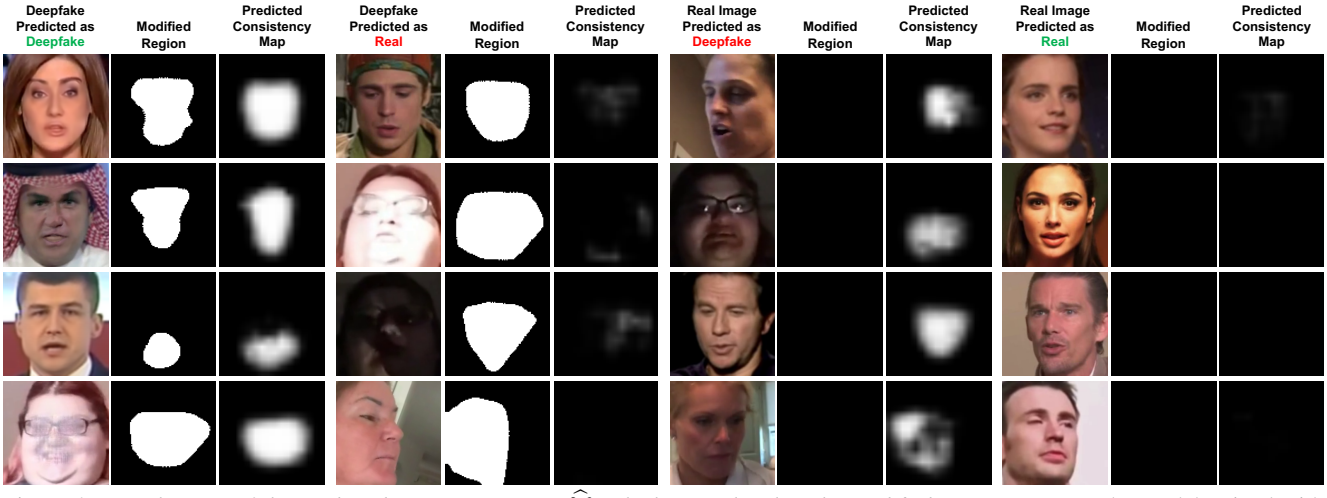

Figure 4: *Visualization of the predicted consistency maps $\widehat{\mathbb{M}}$, which try to localize the modified regions.* We use the model trained with real videos of FF++ augmented by I2G in the cross-dataset, and the predictions are computed from the predicted consistency volume, as mentioned in Section 3.1. The ground truth modified regions are generated by DSSIM, as discussed in Section 4.2.

Method	Hyper-Parameter	Test Set (AUC (%))				Avg
		DFR	CD2	DFDC	DFDC-P	
I2G	$\lambda = 0$	95.12	78.18	51.72	69.93	73.74
PCL + I2G	$\lambda = 1$	99.1	86.52	60.65	74.13	80.10
PCL + I2G	$\lambda = 10$	99.41	90.03	**67.52**	**74.37**	**82.83**
PCL + I2G	$\lambda = 100$	**99.78**	**90.98**	63.22	74.36	82.09

Table 7: *Ablation study on the effect of PCL on DFR, CD2, DFDC, and DFDC-P datasets.* The use of large λ significantly improves the cross-dataset performance, especially on DFDC.

Method	Train Set	Test Set (AUC (%))				Avg
		DFR	CD2	DFDC	DFDC-P	
PCL	DF	90.42	84.59	66.26	75.49	79.19
PCL + I2G	DF	**99.64**	**91.92**	**73.08**	**80.83**	**86.37**
PCL	DFDC-P	51.61	82.82	69.14	**95.53**	74.78
PCL + I2G	DFDC-P	**92.25**	**87.65**	**71.12**	94.38	**86.35**

Table 8: *Ablation study on the effect of I2G as joint-training on DF and DFDC-P datasets.* I2G can enhance the variety of training data, thereby improving the generalization of our model.

terpretable visualizations clues (Sec. 3.1) about the modified region. Figure 4 visualizes some examples generated by PCL along with the corresponding input images and ground truth. When feeding a real image, in most cases, the visualization is a pure blank image, indicating that the input's source features are consistent. When testing the deepfakes, the predicted consistency map can adequately match with the ground truth. We also compute the average value of the consistency volume from real images, and get 0.9854 and 0.9866 using in- and cross-dataset models, respectively. These statistical numbers indicate that PCL predicts all entries in the consistency volume of correctly predicted samples to be consistent with high average confidence, rather than simply segmenting the full face region. We also investigate some failure cases, the inconsistencies are caught by mistake in our model, which might be caused by the lighting and unusual texture. Besides, we observe that lower quality samples lead to false-negative predictions due to either high compression or high/low exposure.

4.7. Limitations

Although our results are encouraging, our approaches still have limitations, which raise opportunities for future work. First, as the game between the forger and the detector is an arms race, one can expect the cues that any published detection method relies on to be removed with the best efforts in the near future. For example, entire face synthesis trains a generative model that directly outputs the whole image, which should be self-consistent by our hypothesis; it is unknown if PCL can handle this type of face forgery. Second, as the false prediction samples indicated, our model can be further improved on low-quality data.

5. Conclusion

We proposed pair-wise self-consistency learning (PCL) to detect face forgeries generated by stitching-based techniques and localize the manipulated regions, based on a less attended cue: the inconsistency of source features within the modified images. PCL only contains a few parameters and can serve as a plugin module upon common backbone networks. We also developed a new light-weight image synthesis method, called inconsistency image generator (I2G), to efficiently support PCL training by dynamically generating forged images along with annotations of their manipulated regions. Experimental results showed that PCL and I2G are competitive against state-of-the-art methods on seven popular datasets, providing a strong baseline for future research.

References

[1] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017.

[2] Luca Bondi, Luca Baroffio, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Processing Letters*, 24(3):259–263, 2016.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 1021–1030, Oct. 22–29 2017.

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, Jun. 14–19 2020.

[5] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, Jun 14–19 2020.

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

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, Jun. 20–25 2009.

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

[9] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv:1910.08854*, 2019.

[10] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research, 2019. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[11] FaceSwapDevs. Deepfakes, 2019. https://github.com/deepfakes/faceswap.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, Jun. 26–Jul. 1 2016.

[13] Yihao Huang, Felix Juefei-Xu, Run Wang, Xiaofei Xie, Lei Ma, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. FakeLocator: Robust localization of GAN-based face manipulations via semantic segmentation networks with bells and whistles. *arXiv:2001.09598*, 2020.

[14] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proc. European Conference on Computer Vision*, pages 101–117, Sep 8–14 2018.

[15] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2895, Jun. 14–19 2020.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, Jun. 16–20 2019.

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *arXiv:1912.04958*, 2019.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations*, 2015.

[19] Pavel Korshunov and Sebastien Marcel. DeepFakes: a new threat to face recognition? assessment and detection. *arXiv:1812.08685*, 2018.

[20] Marek Kowalski. FaceSwap, 2018. https://github.com/MarekKowalski/FaceSwap.

[21] Akash Kumar and Arnav Bhavsar. Detecting deepfakes with metric learning. *arXiv:2003.08645*, 2020.

[22] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 14–19 2020.

[23] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, Jun. 14–19 2020.

[24] Wei Li, Yuanjun Xiong, Shuo Yang, Siqi Deng, and Wei Xia. SMOT: Single-shot multi object tracking. *arXiv:2010.16031*, 2020.

[25] Xiaodan Li, Yining Lang, Yuefeng Chen, Xiaofeng Mao, Yuan He, Shuhui Wang, Hui Xue, and Quan Lu. Sharp multiple instance learning for DeepFake video detection. In *Proc. ACM Multimedia*, Oct. 12–16 2020.

[26] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *Proc. International Workshop on Information Forensics and Security*, 2018.

[27] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 16–20 2019.

[28] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, Jun. 14–19 2020.

[29] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, Jun. 16–20 2019.

[30] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Determining digital image origin using sensor imperfections. In *Proc.*

Image and Video Communications and Processing, 2005.

[31] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *Proc. European Conference on Computer Vision*, Aug. 23–28 2020.

[32] Owen Mayer and Matthew C Stamm. Learned forensic source similarity for unknown camera models. In *International Conference on Acoustics, Speech and Signal Processing*, pages 2012–2016, 2018.

[33] Owen Mayer and Matthew C Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15:1331–1346, 2019.

[34] Owen Mayer and Matthew C Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020.

[35] Jacek Naruniec, Leonhard Helminger, Christopher Schroers, and Romann M. Weber. High-resolution neural face swapping for visual effects. In *Proc. Eurographics Symposium on Rendering*, 2020.

[36] João C. Neves, Ruben Tolosana, Ruben Vera-Rodriguez, Vasco Lopes, Hugo Proença, and Julian Fierrez. GANprintR: Improved fakes and evaluation of the state-of-the-art in face manipulation detection. *Journal of Selected Topics in Signal Processing*, 14(5):1038–1048, 2020.

[37] Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *Proc. Biometrics Theory, Applications and Systems*, Sep. 23–26 2019.

[38] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. Deepfake detection based on the discrepancy between the face and its context. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[39] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 2003.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Medical Image Computing and Computer-assisted Intervention*, 2015.

[41] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 1–11, Oct. 27–Nov. 2 2019.

[42] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, Jun. 16–20 2019.

[43] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. IEEE/CVF International Conference on Computer Vision Workshops*, pages 1274–1283, Oct. 22–29 2017.

[44] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, Jun. 26–Jul. 1 2016.

[45] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. In *ACM Transactions on Graphics*, volume 38, pages 1–12, 2019.

[46] Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez, and Ruben Vera-Rodriguez. DeepFakes evolution: Analysis of facial regions and fake detection performance. *arXiv:2004.07532*, 2020.

[47] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *arXiv preprint arXiv:2001.00179*, 2020.

[48] Loc Trinh, Michael Tsang, Sirisha Rambhatla, and Yan Liu. Interpretable and trustworthy deepfake detection via dynamic prototypes. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2021.

[49] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Proc. Advances in Neural Information Processing Systems*, Dec. 6–12 2020.

[50] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. *arXiv:1807.11649*, 2018.

[51] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. FakeSpotter: A simple yet robust baseline for spotting ai-synthesized fake faces. *arXiv:1909.06122*, 2019.

[52] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, Jun 14–19 2020.

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, Jun. 18–22 2018.

[54] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[56] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–534, Jun. 20–25 2011.

[57] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[58] Ning Yu, Larry Davis, and Mario Fritz. Attributing fake images to GANs: Analyzing fingerprints in generated images. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 7556–7566, Oct. 27–Nov. 2 2019.

[59] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[60] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-

stream neural networks for tampered face detection. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 21–26 2017.

[61] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. WildDeepfake: A challenging real-world dataset for deepfake detection. In *Proc. ACM International Conference on Multimedia*, pages 2382–2390, Oct. 2020.