Learning Second Order Local Anomaly for General Face Forgery Detection

Jianwei Fei[1*] Yunshu Dai[1*] Peipeng Yu[2] Tianrun Shen[3] Zhihua Xia[2,1†] Jian Weng[2]

[1]Nanjing University of Information Science and Technology [2]Jinan University [3]Nanjing University

{fjw826244895,daiyunshu0102,ypp865,xia_zhihua}@163.com

tiruns@yeah.net cryptjweng@gmail.com

Abstract

In this work, we propose a novel method to improve the generalization ability of CNN-based face forgery detectors. Our method considers the feature anomalies of forged faces caused by the prevalent blending operations in face forgery algorithms. Specifically, we propose a weakly supervised Second Order Local Anomaly (SOLA) learning module to mine anomalies in local regions using deep feature maps. SOLA first decomposes the neighborhood of local features by different directions and distances and then calculates the first and second order local anomaly maps which provide more general forgery traces for the classifier. We also propose a Local Enhancement Module (LEM) to improve the discrimination between local features of real and forged regions, so as to ensure accuracy in calculating anomalies. Besides, an improved Adaptive Spatial Rich Model (ASRM) is introduced to help mine subtle noise features via learnable high pass filters. With neither pixel level annotations nor external synthetic data, our method using a simple ResNet18 backbone achieves competitive performances compared with state-of-the-art works when evaluated on unseen forgeries.

1. Introduction

Recent progress in face synthesis technologies allows the low-entry production of sophisticated fake facial content which causes severe trust issues. Concerns are growing over the nefarious use of such face forgery technologies. To deal with this problem, many methods have been developed to detect face forgeries using different traces, such as obvious visual artifacts [24, 29, 39], frequency domain cues [10, 12, 20, 33], temporal anomalies [3, 28, 37, 38, 47], or multimodality conflicts [1, 5, 30]. But these traces are not so generalizable that the detectors may fail when encountering unseen forgeries. Thus recent works are looking for
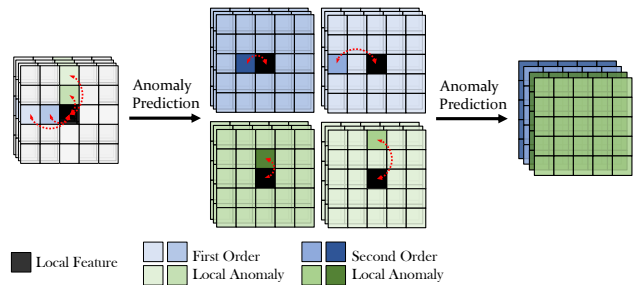
Figure 1. Second Order Local Anomaly (SOLA) learning module decomposes the neighborhood and predicts the first and second order anomaly maps.

more universal forgery traces, so as to boost the generalizability.

All the face forgery algorithms, whether face swapping or reenactment, need to blend the forged regions into the original background. The two parts inevitably hold different features, especially in high frequency regions, resulting in anomaly in forged images. On such observation, some methods are proposed to capture the feature inconsistency [4, 22, 52]. [52] propose a patch wise consistency learning method to mine anomalies in forged face images and succeed in generalizing to unseen forgeries. However, these works treat regions with various distances equally, which violates the fact that natural images hold different dependencies of short and long distances. Moreover, they usually require either pixel level annotations or external synthetic data. Although pixel level annotations can be created in lab environments, the need for them limits the usage of forged faces transmitted in the real world.

In this work, we focus on capturing forgery traces from the perspective of local anomaly. Specifically, given a face image, we extract its deep features with a CNN backbone, and divide the neighborhood of a local feature into 4 groups by different orientations and distances, thereby modeling more fine-grained local anomalies and generating 4 groups of anomaly maps. These anomaly maps are then decomposed again to calculate the second order anomaly which

have a wide range of response to the first order anomaly. The overview is shown in Fig. 1, and more details are in Sec. 3.3. Our method requires neither pixel level annotations nor external data but achieves great cross-domain performance with a small backbone.

Our contributions can be summarized into four-fold:

- We propose a second order local anomaly (SOLA) learning module for isolating forgery faces. SOLA calculates anomalies of a local region in fine-grained fashion and magnifies the forgery traces via calculating the second order anomalies. Classifiers using the anomaly features are proved to have remarkable generalization on unseen forgeries.
- We propose a local enhancement module (LEM) that can be inserted after different stages of the backbone to ensure the discrimination between real and forged regions in deep local features.
- We design an adaptive spatial rich model (ASRM) which inherits the high pass peculiarity of SRM filters in forensics and is able to adaptively adjust the filters according to data.
- Our method requires neither pixel level annotations nor external synthetic data to improve the generalization of face forgery detectors.

2. Related work

Face forgeries can be seen as a game of AI versus AI since most detection technologies are deep learning based. Although these methods perform well under in-domain evaluations, they often suffer performance degeneration on unseen forgeries.

Generalization to Unseen Forgeries. Many efforts are made to improve the generalizability of detectors such as combining auxiliary localization tasks to guide the network focus more on forged regions [8, 22], improving cross entropy loss with metric learning for better class discrimination [20, 28], and introducing domain adaptation to alleviate overfitting on a single domain [35, 41]. Recently, methods using frequency cues [26, 33] or combining features from different domains also perform well [27, 28] in cross-domain evaluations. In [20], one branch process the RGB input, while another uses the DCT transformation to extract high frequency features from different bands. Outputs of the two branches are fused to form more generalized forgery features. Many works similarly fuse features from RGB and frequency domains for more general representation [4, 15, 17, 27]. However, their methods of extracting high frequency features can not fit the data adaptively to capture the most discriminative features.

Anomaly-based Face Forgery Detection. Some works introduce the idea of anomaly detection into image forensics [6, 11] as well as face forgery detection [16, 18], and achieve good generalization. For example, Wu et al. [48]

calculate the differences between local feature and dominant feature of an image, turning the forgery localization into local anomaly detection. Hu et al. [16] propose a dynamic inconsistency aware network to capture both global inconsistency of adjacent frames and local inconsistency of key regions within frames.

Recent methods similarly utilize the patch level anomalies for face forgery detection. They are mainly based on the observation that face forgery algorithms always blend forged regions with original face context, resulting in pixel statistical property anomalies within one image [52]. An example of the pixel statistical property is camera noise [7] which is a high frequency feature left during the imaging process and has been used for detecting image integrity [44]. [52] and [4] propose to divide deep feature maps into patches, and calculate the similarities between patches to form a more generalized pattern that indicates the image integrity. They achieve both great performances under in-domain and cross-domain evaluations. Unlike these methods that explore global level consistency, our method explores more fine-grained local anomaly instead.

3. Method

The overview of our method is illustrated in Fig. 2. Given an input face image, it is fed to the RGB branch, as well as a parallel noise branch whose first layer is our Adaptive Spatial Rich Model (ASRM). These intermediate features extracted from these two branches are fused through Dual Channel Attention Module (DCAM) after each block. Then the Local Enhancement Module (LEM) divide the fused feature maps into patches and enhance the category attributes of each local feature inside. Finally, the Second Order Local Anomaly (SOLA) learning module calculates fine-grained local anomalies of the first and second order, and predicts whether the input is forged.

3.1. Adaptive SRM in Noise Branch

Motivation. Learning from the RGB data is insufficient to capture high frequency features that are crucial for image forensics. To solve this problem, SRM (Spatial Rich Model) [13] has been widely applied for preprocessing [15] and layer initialization [34] to extract high frequency noise. But if used for preprocessing, the handcrafted filters in SRM can not adaptively update to fit the data. Besides, if used for initialization, the backpropagation will adjust the filters and break their high pass peculiarity. In this work, we address this dilemma by introducing a constraint to let SRM adaptively update while maintaining their high frequency learning ability.

Design of ASRM. We start with original SRM following [13]. SRM models the noise residuals $R_{i,j}$ using high pass filters by:

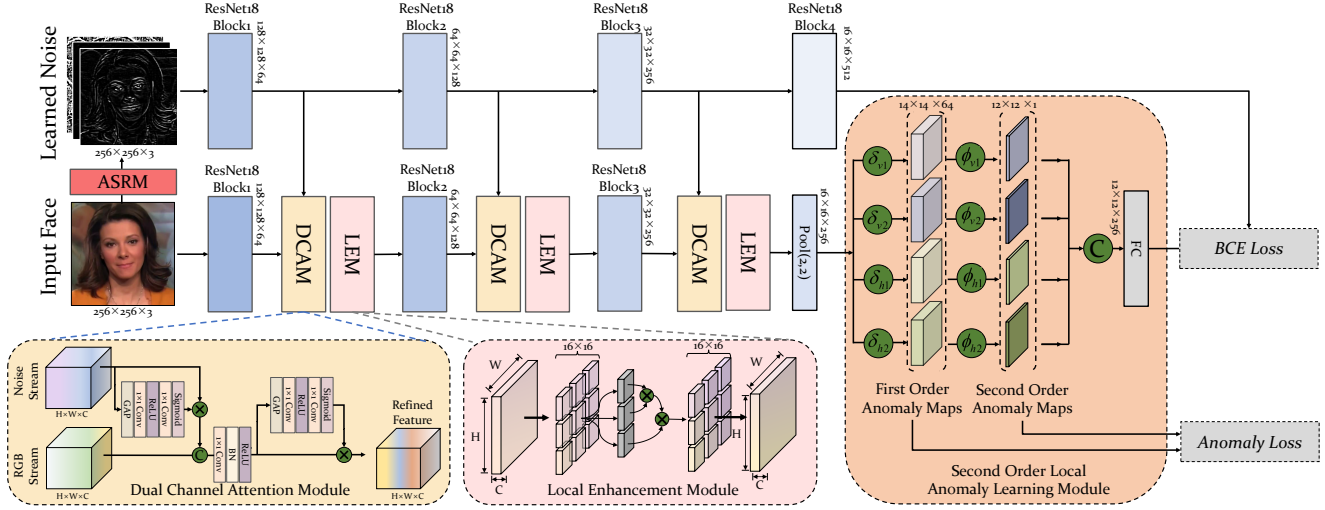$$R_{i,j} = P(\mathcal{N}_{i,j}) - I_{i,j}, \qquad (1)$$

Figure 2. Overview of the proposed method (with ResNet18 backbone).

where $\mathcal{N}_{i,j}$ is the neighborhood of center pixel $I_{i,j}$, $P(\cdot)$ is a predictor of $I_{i,j}$ based on $\mathcal{N}_{i,j}$. The residual is then quantitated by q, following with rounding and truncation:

$$R_{i,j} \leftarrow trun(round(\frac{R_{i,j}}{q})). \qquad (2)$$

We can see that the $P(\cdot)$ can be realized by a standard convolution. Following [2], let $\boldsymbol{I} * \boldsymbol{w}$ denotes a standard convolution on input image $\boldsymbol{I}$ with kernel $\boldsymbol{w}$, and o notes an impulse filter with central value being 1 and others being 0, we then have:

$$\boldsymbol{h} = \boldsymbol{I} * \boldsymbol{w} = \boldsymbol{I} * \hat{\boldsymbol{w}} - \boldsymbol{I} = \boldsymbol{I} * (\hat{\boldsymbol{w}} - o), \qquad (3)$$

where the central value of $\hat{\boldsymbol{w}}$ is 0. Then the central value of $\boldsymbol{w}$ is -1 and $\boldsymbol{I} * \boldsymbol{w}$ is equal to Eq. 1.

We select 3 out of 30 filters in SRM as they have been experimentally proved to be effective enough [53]. The filters are repeated 3 times for RGB images as shown in Fig.3. To let their center values be -1, we quantitate them by 2, 4, and 12 respectively. We can see that after quantitation, the
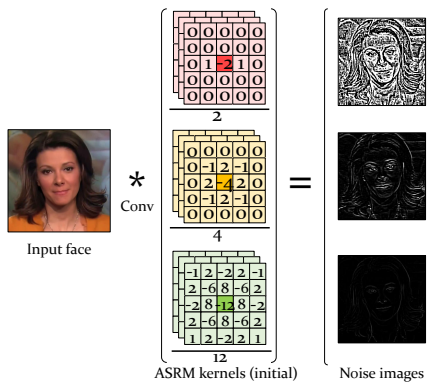
sum of the remaining elements is 1 which perfectly fit the requirements of Eq. 3. To allow the filters learnable during network training while keep their peculiarity in high pass filtering, we reset their central element to -1 and normalize the remaining elements to force their sum to be 1:

$$\begin{cases} \boldsymbol{w}_{0,0} = -1, \\ \sum_{i \neq 0, j \neq 0} \boldsymbol{w}_{i,j} = 1, \end{cases} \qquad (4)$$

where i, j denotes the index of elements in kernels and $(0, 0)$ is the central one. We execute this constraint after each backpropagation.

ASRM is the first layer used for the preprocessing of the noise branch. Different from using fixed SRM kernels, ASRM enables the kernels to adaptively update during the backpropagation and extract discriminative noise features that can not be directly learned from RGB data.

The noise features extracted by ASRM provide subtle high frequency cues, but different channels of the feature maps have different contributions to generalization. Besides, different channels in the fused features are of different importance in forgery representations. Thus, we propose the Dual Channel Attention Module (DCAM) to fuse the intermediate features of RGB and noise branches. DCAM uses channel attention [46] twice to emphasize channels of more importance before and after the fusion.

3.2. Local Enhancement Module

SOLA calculates local anomalies on such an assumption: each pixel in the feature maps is a local feature vector associated with a corresponding local region whose size is the ratio of input size to feature size. However, a feature vector is associated with a larger region due to the expansion of receptive field after several layers. The feature vectors are

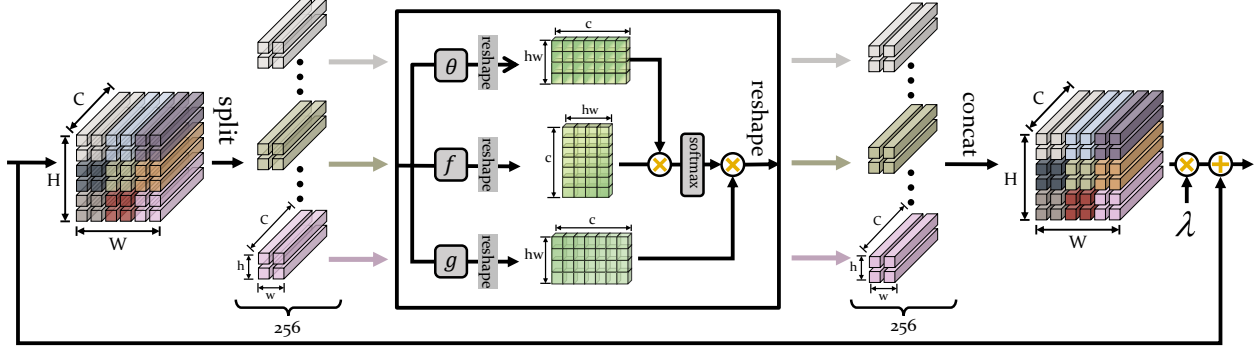Figure 3. The Adaptive Spatial Rich Model (ASRM) convolution.

Figure 4. The Local Enhancement Module (LEM).

thereby affected by information of outer regions and lose their discrimination in representing local regions. We notice that some works have been proposed to capture long range dependencies in images [45,50] with self attention. Following this idea, we propose the Local Enhancement Module (LEM) to improve the discrimination of local feature vectors by assigning them with different weights.

Specifically, LEM takes as input the intermediate feature maps $\mathbf{F}^l \in \mathbb{R}^{h_l \times w_l \times c_l}$ where l denotes the hierarchy. We spatially divide $\mathbf{F}^l$ into 16×16 non-overlapping patches $\mathbf{p}_k^l, k = 1, 2, ..., 256$ since size of the final feature maps is 16×16. $\mathbf{p}_k^l$ is transformed using embedding functions θ and f to calculate a weight matrix $\mathbf{W}_k^l$ by:

$$\mathbf{W}_k^l = softmax(\theta(\mathbf{p}_k^l)^T \otimes f(\mathbf{p}_k^l)). \tag{5}$$

The elements in $\mathbf{W}_k^l$ indicate the relevance between local features in $\mathbf{p}_k^l$ and separate local features by assigning different weights, given that local features of forged regions are more relevant to that of forged regions and vice versa. $\mathbf{p}_k^l$ is also transformed by g and then enhanced by:

$$\hat{\mathbf{p}}_k^l = \mathbf{W}_k^l \otimes g(\mathbf{p}_k^l). \tag{6}$$

As illustrated in Fig.4, LEM concats $\hat{\mathbf{p}}_k^l$ to form the enhanced feature maps $\hat{\mathbf{F}}^l$ which is then multiplied by a learnable scale parameter λ and added with the input feature.

3.3. Second Order Local Anomaly Learning

In this work, we propose the Second Order Local Anomaly (SOLA) learning module to detect face forgeries by fine-grained local anomalies. Let $\mathbf{F}_{i,j} \in \mathbb{R}^C$ denotes the (i^{th}, j^{th}) feature vector that corresponds to a local region in the input image. To obtain more fine-grained representation of local anomalies, SOLA first decomposes the neighborhood of $\mathbf{F}_{i,j}$ into horizontal/vertical and nearest/next nearest neighbors by direction and distance respectively. In practice, we only predict the anomalies of upper and left neighbors due to the symmetry. Then the first order anomaly

maps of each $\mathbf{F}_{i,j}$ are predicted by Eq. 7:

$$\begin{cases} \mathcal{M}_{v1,(i,j)} = \delta_{v1}(\mathbf{F}_{i,j}, \mathbf{F}_{i,j-1}), \\ \mathcal{M}_{v2,(i,j)} = \delta_{v2}(\mathbf{F}_{i,j}, \mathbf{F}_{i,j-2}), \\ \mathcal{M}_{h1,(i,j)} = \delta_{h1}(\mathbf{F}_{i,j}, \mathbf{F}_{i-1,j}), \\ \mathcal{M}_{h2,(i,j)} = \delta_{h2}(\mathbf{F}_{i,j}, \mathbf{F}_{i-2,j}), \end{cases} \tag{7}$$

where $\delta(\cdot)$ is the anomaly predictor realized by 1×2 or 2×1 convolutions (dilation = 2 for next nearest neighbors, otherwise 1), v and h denote vertical and horizontal neighbors, 1 and 2 denote the nearest and next nearest neighbors. One kernel in $\delta(\cdot)$ can predict only one anomaly map, thus we set each $\delta(\cdot)$ to contain 64 kernels so that it can generate 64 anomaly maps to capture multiple anomalies.

Inspired by the idea of the Laplace operator that computes the second derivative, we propose the second order local anomaly learning to obtain magnified responses to the first order anomalies. To this end, SOLA applies the local decomposition again on the first order anomaly maps and calculates the second order local anomaly maps by Eq. 8. A pixel in the second order anomaly maps is responded to a region in the first order anomaly maps so as to capture a wider range of fine-grained anomalies.

$$\begin{cases} \mathcal{M}'_{v1,(i,j)} = \phi_{v1}(\mathcal{M}_{v1,(i,j)}, \mathcal{M}_{v1,(i,j-1)}), \\ \mathcal{M}'_{v2,(i,j)} = \phi_{v2}(\mathcal{M}_{v2,(i,j)}, \mathcal{M}_{v2,(i,j-2)}), \\ \mathcal{M}'_{h1,(i,j)} = \phi_{h1}(\mathcal{M}_{h1,(i,j)}, \mathcal{M}_{h1,(i-1,j)}), \\ \mathcal{M}'_{h2,(i,j)} = \phi_{h2}(\mathcal{M}_{h2,(i,j)}, \mathcal{M}_{h2,(i-2,j)}), \end{cases} \tag{8}$$

$\phi(\cdot)$ denotes the second order anomaly predictor that is similar to $\delta(\cdot)$ but contains only one single kernel. Finally, we stack 4 outputs of $\phi(\cdot)$ to form a 4 channel anomaly map and pass it to the classifier. The classifier only contains a convolution layer followed by a global average pooling layer and a fully-connected layer.

3.4. Loss Function

We first give the overall loss function for supervised training where the forgery masks are available, and then

describe the weakly supervised training strategy using our single side loss.

Supervised Training. let $\mathbf{M} \in \{0,1\}^{H \times W}$ denotes the forgery mask, we divide it into patches $\mathbf{m}_{i,j}$ (16×16 pixels in our case), we use its averaged value $AVG(\mathbf{m}_{i,j})$ to represent the forgery score of (i^{th}, j^{th}) patch in the face image. We use a *hard thresholding* fashion to calculate the first and second order anomaly scores of two adjacent patches as shown in (9a) and (9b) respectively. Without loss of generality, we use (i', j') to denote a neighbor of (i, j).

$$a_{i,j} = \begin{cases} 1 & \text{if} \quad |AVG(\mathbf{m}_{i,j}) - AVG(\mathbf{m}_{i',j'})| > 0, \\ 0 & \text{else.} \end{cases} \quad (9a)$$

$$a'_{i,j} = \begin{cases} 1 & \text{if} \quad |a_{i,j} - a_{i',j'}| > 0, \\ 0 & \text{else.} \end{cases} \quad (9b)$$

By iterating (9a) and (9b) for each $\mathbf{m}_{i,j}$, we obtain the ground truth of the first order anomaly maps $\tilde{\mathcal{M}} \in \{0,1\}^{h \times w \times 64}$ (repeated 64 times), and the second order anomaly maps $\tilde{\mathcal{M}}' \in \{0,1\}^{h' \times w'}$. The local anomaly maps can be optimized using pixel wise binary cross entropy (BCE):

$$\mathcal{L}_A = \beta \sum_* BCE(\mathcal{M}_*, \tilde{\mathcal{M}}_*) + \gamma \sum_h BCE(\mathcal{M}'_*, \tilde{\mathcal{M}}'_*), \quad (10)$$

where $* \in \{v1, v2, h1, h2\}$, β and γ are the weights for the first and second order anomaly maps. The overall loss for training our model is:

$$\mathcal{L}_{total} = \alpha L_{cls} + L_A, \quad (11)$$

L_{cls} is the BCE loss for the classifier and α is the weight.

Weakly Supervised Training. Although SOLA learning module predicts patch level anomalies, we can train it in a weakly supervised manner where only image level labels are available. To this end, we have two hypotheses: (1) The pixels of anomaly maps of real faces should be all zero or close to zero. In the meantime, there should be a portion of nonzero pixels in the anomaly maps of forged faces. (2) Anomaly maps vary according to different forgery algorithms, but they should be always nonzero. On these hypotheses, we introduce a single side loss in Eq. 12 that only penalizes the anomaly maps of real faces. This loss ensures multiple patterns of anomalies in forged faces, thereby improving the generalization:

$$\mathcal{L}_A = \sum_{I \in real} \sum_* \underbrace{\beta ||\mathcal{M}_* - \tilde{\mathcal{M}}_*||_1}_{\text{First Order}} + \underbrace{\gamma ||\mathcal{M}'_* - \tilde{\mathcal{M}}'_*||_1}_{\text{Second Order}}, \quad (12)$$

where $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$ are both all-zero.

4. Experiments

4.1. Experiments details

Datasets. We use a wide range of popular face forgery datasets to evaluate the proposed method, including FaceForensics++ (FF++) [36], Celeb-DF v2 (CD2) [25], Deep-fakeDetection Dataset (DFD) [31], and FaceShifter (Fshi) [21]. In FF++, 1000 original videos are forged by four forgery algorithms: DeepFakes, Face2Face [43], FaceSwap, and NeuralTextures [42]. All face patches are cropped according to their masks if available, otherwise, we use RetinaFace [9] to detect faces and crop the patches. To preserve enough background, the cropped patches are set to be 2.6 times the size of masks or bounding box. All face patches are resized to 256×256 and normalize to $[0, 1]$ by dividing 255.

Evaluation Metrics. We report frame level AUC (area under the receiver operating characteristic curve) as most of the previous works do. The experimental results of other methods which we use for comparison are directly cited.

Implementation Details. All the experiments are implemented with Pytorch [32] with 4×NVIDIA RTX 3090 24GB. The backbone is initialized with ImageNet pre-trained weights and trained using Adam optimizer [19] with learning rate 1e-3, betas 0.9 and 0.999, and epsilon 1e-8. The batch size is 32 and number of epochs is 50 without early stopping.

Methods	Backbone	DF	F2F	FS	NT
Xception [36]	Xception	99.38	99.53	99.36	99.50
PBD [39]	Xception	97.00	95.00	98.00	98.00
Face X-ray [22]	HRNet	99.17	99.06	99.20	98.93
S-MIL [23]	Xception	99.84	99.34	99.61	98.85
SOLA -*weakly sup*	ResNet18	**100**	**99.67**	**100**	**99.82**
SOLA -*sup*	ResNet18	100	99.56	99.98	99.76

Table 1. In-domain performences on FF++.

4.2. In-Domain Evaluations

We first report the performance of our method under in-domain evaluations. The results of FF++ are shown in Table 1. -*weakly sup* and -*sup* denote SOLA trained with different strategies. Previous methods have achieved great performance, but our method still surpasses the best competitors by about 1% in terms of average AUC . Note that our method only uses a relatively small backbone ResNet18 while most of other methods use a large backbone like Xception. On the whole, weakly supervised SOLA performs slightly better, proving that the single side loss allows SOLA to capture different anomalies precisely.

4.3. Cross-domain Evaluations

In this section, we focus on the more challenging cross-domain evaluations to explore the performances of our method on unseen forgeries. Table 2 shows the results of cross-domain evaluations on FF++. Here, the models are trained on only one dataset and evaluated on all four datasets. Although Xception achieves 99.42% average AUC of in-domain evaluations, it can not generalize to

Model	Backbone	Train Set	Test Set				Avg
			DF	F2F	FS	NT	
Xception [36]	Xception		99.38	75.05	49.13	80.39	75.99
Face X-ray [22]	HRNet	DF	99.17	94.14	**75.34**	93.85	90.63
SOLA -*weakly sup*	ResNet18		100	**97.29**	63.59	98.45	89.83
SOLA -*sup*	ResNet18		**100**	96.95	69.72	**98.48**	**91.28**
Xception [36]	Xception		87.56	99.53	65.23	65.90	79.56
Face X-ray [22]	HRNet	F2F	98.52	99.06	72.69	91.49	90.44
SOLA -*weakly sup*	ResNet18		99.61	**99.67**	84.24	**97.48**	95.25
SOLA -*sup*	ResNet18		**99.73**	99.56	**93.50**	96.02	**97.20**
Xception [36]	Xception		70.12	61.70	99.36	68.71	74.97
Face X-ray [22]	HRNet	FS	93.77	92.29	99.20	86.63	92.97
SOLA -*weakly sup*	ResNet18		93.18	97.59	**100**	94.93	96.43
SOLA -*sup*	ResNet18		**99.11**	**98.13**	99.98	**92.07**	**97.32**
Xception [36]	Xception		93.09	84.82	47.98	99.50	81.35
Face X-ray [22]	HRNet	NT	99.14	**98.43**	70.56	98.93	91.77
SOLA -*weakly sup*	ResNet18		**99.95**	94.83	57.32	**99.82**	87.97
SOLA -*sup*	ResNet18		99.64	97.69	**90.20**	99.76	**96.82**

Table 2. Cross-domain evaluations on FF++.

other forgeries well and its performances of cross-domain evaluations drop sharply. Meanwhile, Face x-ray [22] is designed to detect the blending boundaries instead of specifical artifacts caused by forgery algorithms and made significant progress in generalization. We can see that weakly supervised SOLA performs closely to Face x-ray. Face x-ray requires pixel level forgery masks to locate the boundaries and train their model, but our method also achieves a great improvement of generalization ability with only image level labels. With the help of pixel level annotations, supervised SOLA surpasses all other methods in terms of average AUC of cross-domain evaluations. The overall performance of supervised SOLA on FF++ exceeds Xception and Face x-ray by 17.69% and 4.21%.

Model	Backbone	Train Set	Test Set
			FShi
Xception [36]	Xception	FF++	72.00
PBD [39]	Xception	FF++	57.80
FWA [24]	ResNet152	FF++	65.50
Face X-ray [22]	HRNet	FF++	92.80
LipForensics [14]	ResNet18	FF++	97.10
SOLA -*weakly sup*	ResNet18	FF++	97.27
SOLA -*sup*	ResNet18	FF++	**98.72**

Table 3. Cross-domain evaluation results on FaceShifter (trained on FF++). Our method has better performance than state-of-the-art works and is also competitive while using only one dataset.

Results of cross-domain evaluations on a more advanced face forgery algorithm FaceShifter (Fshi) are shown in Table 3. Following other works, we train our model on FF++

Model	Train Set	Test Set
		CD2
F^3 Net [33]	FF++(c23)	65.17
FWA [24]	Self-made	57.32
MADD [51]	FF++(c23)	67.44
MTD-Net [49]	FF++(c23)	70.12
Two Branch [28]	FF++(c40)	73.41
F3Net [20]	FF++(c23)	65.20
LRL [4]	FF++(c23)	**78.26**
GFF [27]	FF++(c23)	65.20
MADD [51]	FF++(c23)	67.44
SPSL [26]	FF++(c23)	76.88
LipFor [14]	FF++(c23)	82.40
SOLA -*weakly sup*	DF(c23)	72.47
SOLA -*sup*	DF(c23)	76.02

Table 4. Cross-domain evaluations on CD2. Our methods trained using only DF-c23 outperforms most methods.

and evaluate it by Fshi. Weakly supervised and supervised SOLA both exceed state-of-the-art method, by 0.17% and 1.62% AUC respectively. In addition to the evaluations with two supervision fashions, we also evaluate weakly supervised SOLA trained with only one dataset. Although the performances of weakly supervised SOLA are slightly lower than the supervised SOLA when trained using only one dataset, they are generally still better than most of the recent methods. These results demonstrate the local anomaly traces are universal in the forged faces and can be generalized to unseen forgeries.

The comparisons of cross-domain evaluation on CD2 are given in Table 4. Note that the compared methods are trained with different datasets, so we present the results for reference only. Most methods use all four datasets of FF++ so as to obtain good generalization on CD2. However, weakly supervised SOLA gets 72.47% AUC when trained using only the compressed DF dataset, exceeding most state-of-the-art competitors. Supervised SOLA further improve the result to 76.02% which is 3.55% higher than the Two-Branch [28] that similarly fuses RGB and frequency features.

Model	Train Set	Test Set			
		DF	F2F	FS	NT
Xception [36]	CD2	**87.69**	75.17	54.19	72.89
SOLA -*weakly sup*		85.77	**85.72**	**86.78**	**85.64**
Xception [36]	DFD	94.79	76.96	47.13	84.91
SOLA -*weakly sup*		**95.51**	**85.51**	**60.81**	84.01

Table 5. Cross-domain evaluations on DF, F2F, FS, and NT by training on CD2 and DFD.

Table 5 shows the cross-domain results of our method trained on CD2 and DFD and tested on FF++. Here, SOLA can be only trained without supervision given that the ground truth forgery masks of both the datasets are not available. The overall AUCs are 85.98% and 81.46% when trained on CD2 and DFD respectively, exceeding the Xception by 13.49%, 5.57%, proving that our method generalizes well on multiple datasets even with only image level annotations.
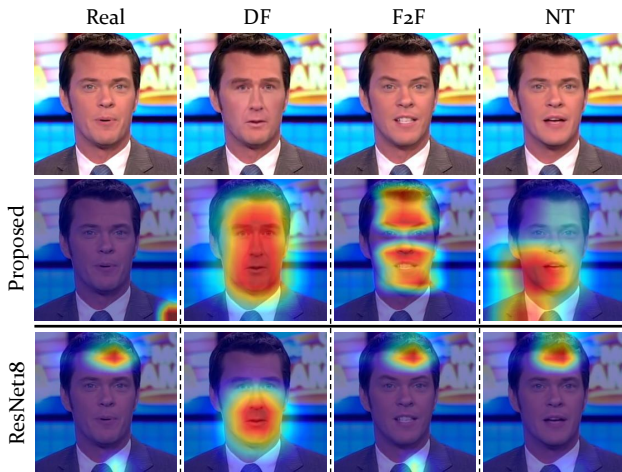

Figure 5. Grad-CAM [40] for the *forgery* class of different of forged faces. We can see the proposed SOLA can respond to various forgeries while the vanilla ResNet18 can not.

To explore which region our method pays attention when encountering unseen forgeries, we use Grad-CAM [40] to generate heat maps as illustrated in fig. 5. The warm color marks the regions that respond strongly to the prediction of *forgery*. Here, a ResNet18 and our method are both trained on DF. Obviously, they can well concentrate on the forged regions of faces created by DF and have almost no response to real faces. However, our method not only captures forged regions on DF more comprehensively, but also responds to unseen forgeries while ResNet18 can not. We can see that ResNet18 fails to capture the forged regions on F2F and NT, thus it responses the same to F2F and NT as to the real face. This agrees with the weak performance of deep learning models in cross-domain evaluations and reveals the generalization ability of our method on unseen forgeries.

4.4. Ablation Studies

The effect of ASRM. So far, we have been training our method with the proposed ASRM. In this section, we demonstrate its effectiveness by replacing it with several variants. Table 6 shows the performances of different noise extraction strategies in the noise branch: -*w/o noise* denotes SOLA without noise branch, -*srm* and -*lsrm* denote SOLA using fixed SRM filters and learnable SRM filters (use filters as initialization of the first layer without any constraint). All variants are trained on CD2 and DFD without pixel level annotations. The results show that the noise branch effectively improves the performance of SOLA on DFD, by 16.5%, 16.68%, and 16.67% with different SRM variants.

Model	CD2	DFD
SOLA -*w/o noise*	98.15	83.31
SOLA -*srm*	73.05	99.81
SOLA -*lsrm*	94.67	**99.99**
SOLA -*asrm*	**98.70**	99.98

Table 6. Comparsion results of different variants of SRM.

Fig. 6 shows outputs of all three channels of standard SRM, LSRM, and ASRM across training epochs. The input face is the same as that in Fig. 3. Although the standard SRM extracts high frequency noise, it does not focus on forged regions. Meanwhile, the LSRM loses its high pass peculiarity and fails to extract noise features in the training process. But we can see that ASRM gradually extracts the high frequency noise of face regions, especially the facial organs, providing discriminative cues that can not be learned from CNNs directly.

The effect of LEM. To confirm the effect of LEM, we train the weakly supervised SOLA with different combinations of ASRM and LEM. As shown in Table 7, the models are trained on DF and evaluated on DF, FShi, and DFD. Generally, both ASRM and LEM can improve the overall performances while ASRM contributes more to the generalization and increases the AUC on Fshi and DFD by 8.78% and 10.16%.
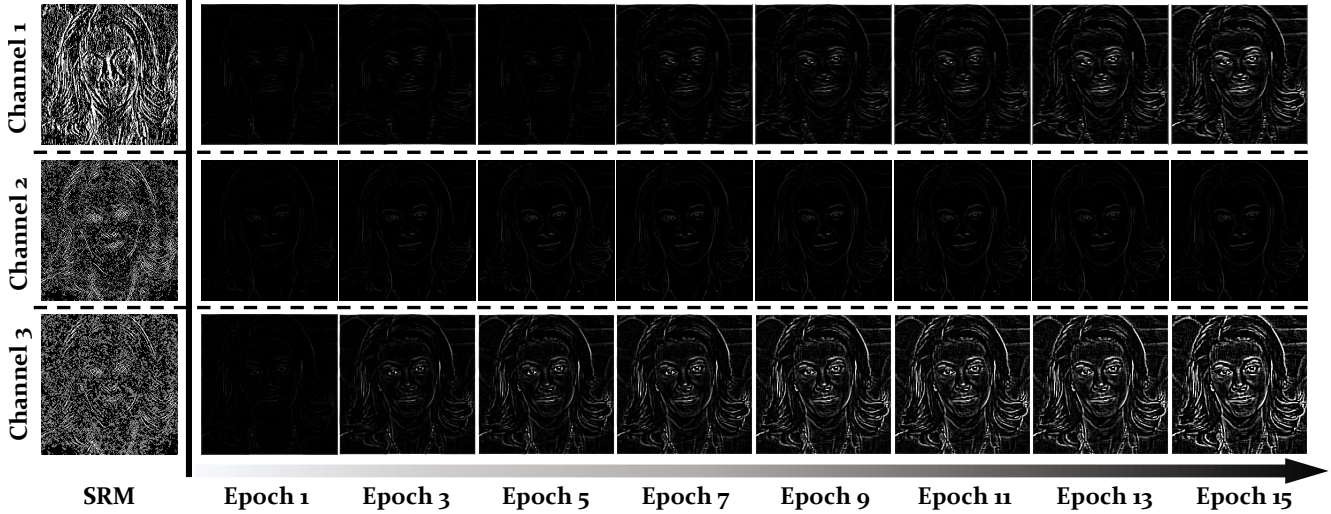
Figure 6. The outputs of different SRM variants over training epochs.

The effect of Patch Size. In the previous experiments, we use the 3 out of 4 blocks in ResNet18 and an additional pooling layer to obtain feaures of size $16 \times 16 \times 256$. To further evaluate the effect of different patch sizes, we change the pooling size to obtain features of size 8×8 and 32×32. The results are shown in Table 7. Although SOLA with different patch sizes all get good results, SOLA with patch size 16 achieves the optimal performances across all datasets. These results accord with the conclusions in [52] as well.

ASRM	LEM	Patch Size	Test Set		
			DF	FShi	DFD
		16	99.90	72.24	78.33
√		16	99.87	81.02	88.49
	√	16	99.98	73.64	80.25
√	√	8	100	80.17	91.10
√	√	16	**100**	**86.53**	**92.61**
√	√	32	100	85.99	87.85

Table 7. Performances of weakly supervised SOLA with different patch size and the effect of LEM.

Backbone	FF++(c0)	FF++(c23)	CD2
ResNet18	99.84	98.10	68.33
ResNet50	99.90	99.14	74.98
ResNet101	**99.90**	**99.25**	**75.05**

Table 8. Comparsion results of different backbones.

The effect of Backbones. We further test the effect of different backbone models on both in-domain and cross domain settings. As in Table 8, we choose 2 other models in ResNet family and report the results on FF++ and CD2

(trained on FF++(c23)). While the overall performances especially the cross-domain performance are much better with ResNet50, the increase is very limited with a deeper ResNet101.

4.5. Limitations

While the proposed method predicts fine-grained local anomalies of different distances and directions, the diversity of different predictors in SOLA is not explicitly ensured, especially under the weakly-sup setting. This could lead to the degeneration of local anomaly predictors and weaken the representations of local anomalies.

5. Conclusion

In this work, we revisit face forgery detection from the perspective of local anomaly detection and propose the SOLA learning module to predict different types of local anomalies in the first and second order. Besides, we design a weakly supervised strategy to train SOLA without pixel level annotations. We also introduce a adaptive spatial rich model to mine subtle high frequency traces using learnable high pass kernels. Experiments on multiple datasets demonstrate that our method achieves competitive performance with a small backbone and generalize to unseen forgery types well.

Acknowlegdement

References

[1] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 660–661, 2020. 1

[2] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018. 3

[3] Peng Chen, Jin Liu, Tao Liang, Guangzhi Zhou, Hongchao Gao, Jiao Dai, and Jizhong Han. Fsspotter: Spotting face-swapped video by spatial and temporal clues. In *2020 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2020. 1

[4] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1081–1088, 2021. 1, 2, 6

[5] Komal Chugh, Parul Gupta, Abhinav Dhall, and Ramanathan Subramanian. Not made for each other-audio-visual dissonance-based deepfake detection and localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 439–447, 2020. 1

[6] Davide Cozzolino and Luisa Verdoliva. Single-image splicing localization through autoencoder-based anomaly detection. In *2016 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2016. 2

[7] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2019. 2

[8] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020. 2

[9] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020. 5

[10] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7890–7899, 2020. 1

[11] Soroush Fatemifar, Shervin Rahimzadeh Arashloo, Muhammad Awais, and Josef Kittler. Client-specific anomaly detection for face presentation attack detection. *Pattern Recognition*, 112:107696, 2021. 2

[12] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning*, pages 3247–3258. PMLR, 2020. 1

[13] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012. 2

[14] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5039–5049, 2021. 6

[15] Bing Han, Xiaoguang Han, Hua Zhang, Jingzhi Li, and Xiaochun Cao. Fighting fake news: Two stream network for deepfake detection via learnable srm. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021. 2

[16] Ziheng Hu, Hongtao Xie, YuXin Wang, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Dynamic inconsistency-aware deepfake video detection. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 736–742, 8 2021. 2

[17] Gengyun Jia, Meisong Zheng, Chuanrui Hu, Xin Ma, Yuting Xu, Luoqi Liu, Yafeng Deng, and Ran He. Inconsistency-aware wavelet dual-branch network for face forgery detection. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021. 2

[18] Hasam Khalid and Simon S Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 656–657, 2020. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6458–6467, 2021. 1, 2, 6

[21] 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. 5

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020. 1, 2, 5, 6

[23] 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 *Proceedings of the 28th ACM international conference on multimedia*, pages 1864–1872, 2020. 5

[24] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *CVPR Workshops*, 2019. 1, 6

[25] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 3207–3216, 2020. 5

[26] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 772–781, 2021. 2, 6

[27] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16317–16326, 2021. 2, 6

[28] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *European Conference on Computer Vision*, pages 667–684. Springer, 2020. 1, 2, 6, 7

[29] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92. IEEE, 2019. 1

[30] 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. 1

[31] Andrew Gully Nick Dufour. Deepfakedetection dataset. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. 5

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[33] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European Conference on Computer Vision*, pages 86–103. Springer, 2020. 1, 2, 6

[34] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2016. 2

[35] Yuan Rao and Jiangqun Ni. Self-supervised domain adaptation for forgery localization of jpeg compressed images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15034–15043, 2021. 2

[36] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019. 5, 6, 7

[37] Yiwei Ru, Wanting Zhou, Yunfan Liu, Jianxin Sun, and Qi Li. Bita-net: Bi-temporal attention network for facial video forgery detection. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2021. 1

[38] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *CVPR Workshops*, 2019. 1

[39] Steven Schwarcz and Rama Chellappa. Finding facial forgery artifacts with parts-based detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942, 2021. 1, 5, 6

[40] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 7

[41] Ke Sun, Hong Liu, Qixiang Ye, Jianzhuang Liu, Yue Gao, Ling Shao, and Rongrong Ji. Domain general face forgery detection by learning to weight. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2638–2646, 2021. 2

[42] 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. 5

[43] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 5

[44] Davide Cozzolino Giovanni Poggi Luisa Verdoliva. Extracting camera-based fingerprints for video forensics. *Proc. of CVPRW*, 2019. 2

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 4

[46] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[47] Xi Wu, Zhen Xie, YuTao Gao, and Yu Xiao. Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2952–2956. IEEE, 2020. 1

[48] 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*, pages 9543–9552, 2019. 2

[49] Jiachen Yang, Aiyun Li, Shuai Xiao, Wen Lu, and Xinbo Gao. Mtd-net: Learning to detect deepfakes images by multi-scale texture difference. *IEEE Transactions on Information Forensics and Security*, 16:4234–4245, 2021. 6

[50] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 4

[51] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021. 6

[52] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15023–15033, 2021. 1, 2, 8

[53] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018. 3