Thinking in Patch: Towards Generalizable Forgery Detection with Patch Transformation

Xueqi Zhang[1,2](✉), Shuo Wang[1], Chenyu Liu[3], Min Zhang[1,2], Xiaohan Liu[3], and Haiyong Xie[1,2](✉)

[1] University of Science and Technology of China, Hefei 230026, China
[2] Key Laboratory of Cyberculture Content Cognition and Detection, Ministry of Culture and Tourism, Anhui 230027, China
{xqzhang7, minz}@mail.ustc.edu.cn, {shuowangcv, hxie}@ustc.edu.cn
[3] National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data, Beijing 100040, China
liuxiaohan@cetc.com.cn, 2011010090@bupt.cn

Abstract. Nowadays, synthetic faces can completely trick human eyes, which raises social concerns for malicious dissemination of such fake content. As a result, face forgery detection has become a significant research topic. Due to the different distributions of synthetic data in different generation algorithms, it is a great challenge to improve the generalization ability of the face forgery detection algorithm. To address this challenge, we propose a general two-stream patch-based face forgery detection network (*FDPT*), which introduces a patch transformation to encourage the model to focus on stable information in different data. Specifically, a random transformation is designed to help CNN stream extract local subtle artifacts from images. Meanwhile, a sequence transformation is employed to enhance the global spatial representation ability of the image through the CNN-GRU stream. Finally, a fusion strategy is used to improve the detection accuracy. We conduct extensive experiments to show that FDPT achieves state-of-the-art performance on two popular benchmarks. Moreover, FDPT outperforms the recently proposed generalization methods when applied to forgery generated by unseen face manipulation techniques (*e.g.*, 84.39% → 95.53% on Face2Face dataset).

Keywords: face forgery detection; generalization; patch transforamtion.

1 Introduction

With the development of artificial intelligence technologies, researchers have proposed various deep-learning-based generation algorithms to synthesize images and videos. Since a Reddit user first used such algorithms in 2017 [11], fake content generation has gradually penetrated into politics, media, and many other fields. It has become a serious problem that abusing fake images for malicious purposes (*e.g.*, influencing public opinions) will bring negative impacts on the society, economics, and even politics. Therefore, it is necessary to study algorithms for image forgery detection, especially for face forgery detection.

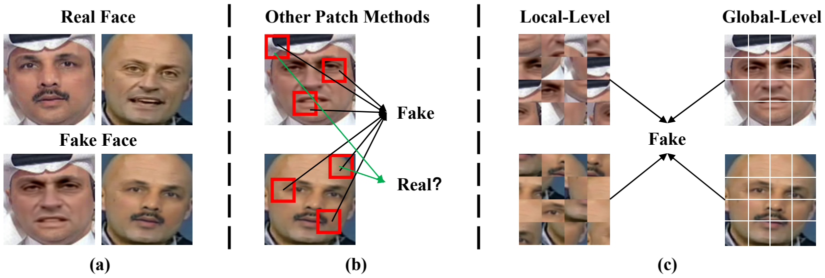

Fig. 1. Comparisons between our method and other patch-based methods. (a) There are two fake faces (bottom) generated by DeepFakes [1] based on the two real faces (top). (b) Other methods based on patch learning assume all patches cropped from a fake image as fake (black line), but some patches may come from the real part (green line). (c) Our method is trained with the global label instead of the aforementioned assumption. FDPT uses patch transformation to capture the local subtle artifacts and global spatial features of the image.

Researchers have made numerous attempts in order to address the challenge of face forgery detection. For instance, a series of earlier works classify an image into a real/fake category by using handcrafted features [7,12,25]. However, they require high-resolution images and exhaustive feature tuning. In the past few years, convolutional neural networks (CNNs) have shown a powerful ability in a number of visual tasks. Therefore, recent works have begun to use deep learning methods to achieve forgery detection [4,15,22,26,31]. However, most of these methods are trained with known face manipulation techniques and have the problem of insufficient generalizability.

Generalizability is highly desired in face forgery detection; in other words, models should perform well not only on the face data used in training (*i.e.*, known face manipulation datasets), but also on other unseen face forgery datasets (*i.e.*, generated by unknown face manipulation techniques). Due to the lack of generalizability, most existing methods for face forgery detection are effective on seen datasets (known face manipulation techniques) and can achieve a detection accuracy up to 98%. However, they tend to suffer from over-fitting and perform poorly (50% or even lower) on unseen datasets. Therefore, forgery detection methods without sufficient generalizability are unsuitable for practical applications.

There exists many challenges in the process of improving generalizability, for example, large-scale dataset dependence, and forged class limitation [14,21,30]. To address such challenges, Chai *et al.* extract the local features from small patches to improve the representation of the image [8]. Compared with the global features of the whole image, the local and subtle features are more stable in different datasets. However, Chai *et al.* assume that all patches cropped from a forgery face are considered fake (as shown in Fig. 1 (a) and (b)). It is clearly not suitable to apply such method to forgery images that consist of many real face parts where some belong to a real one and some do not (as shown in Fig. 1 (b)). Therefore, although it has excellent generalization performance within the entire face forgery datasets, it is not effective across different partial face forgery datasets (*e.g.*, the four manipulations of FaceForensics++ [27]).

Inspired by [8], we find that local information learning is a good method to solve the lack of generalizability. To remove the assumption in [8] that degrades the generalizability, we use a patch transformation strategy to help the model focus on stable artifacts, rather than limiting model learning to a local patch. The model can be trained with the global label without the aforementioned assumption (*i.e.*, a real/fake image corresponds to a real/fake label). Specifically, we randomly shuffle the image patches to help the CNN stream emphasize local artifacts from the image. In addition, we convert the image into a patch sequence and capture the global spatial features by using CNN-GRU stream. Compared to the method proposed in [8], our method not only focuses on the local subtle artifacts (local-level) by the CNN stream but also learns global spatial features (global-level) from the CNN-GRU stream (shown in Fig. 1 (c)). Finally, we fuse different levels of features to further improve the performance and generalizability of the model.

We summarize our contributions as follows. Firstly, we propose a patch random transformation strategy to help the CNN stream focus on the local subtle artifacts of the image. It provides a solution to distinguish the differences between real and fake faces. Secondly, we employ a patch sequence transformation strategy to enrich the global representation of images by the CNN-GRU stream, which firstly introduces spatial features between patches in face forgery detection task. Lastly, we conduct extensive experiments to show the effectiveness of our proposed method; moreover, we achieve meaningful gains in many generalizability experiments.

2 Related Work

2.1 Fake Face Generation

The studies on Face forgery can be divided into two categories: entire face forgery and partial face forgery.

Entire face forgery. The generative adversarial networks (GANs) are usually used to synthesize images [6,16]. PGGAN [18] and StyleGAN [19] are proposed to focus on the high-level attributes (*e.g.*, pose and identity when trained on human faces) in an image and generate a high-resolution image. Glow is a flow-based generation model by using modified 1×1 invertible convolutions and achieves excellent results in interpolation generation [20].

Partial face forgery. It usually contains many meticulous sub-tasks, such as identity swap, expression swap, and attributes manipulation. For this category, StarGAN [9] and FaceApp [2] are proposed to achieve face attributes manipulation by modifying the partial attributes of the face image (*e.g.*, hair, gender, age, *etc*) during the training stage. Similar work includes recently proposed FaceSwap [3], Face2Face [29], DeepFakes [1], and NeuralTextures [28]. Rossler *et al.* collect the fake face videos from four popular generation methods [1,3,28,29] and propose a dataset named FaceForensic++ to facilitate the evaluation of detection methods [27].

2.2 Forgery Detection

We divide the studies on forgery detection into two categories: generalizable forgery detection and patch based forgery detection.

Generalizable Forgery Detection. Recently, many methods achieve a high accuracy on known datasets in forgery detection. However, their accuracies on unseen datasets decrease significantly. To solve the generalizability problem, recent works [8,14,21,30] have been proposed. Specifically, Du *et al.* employ a locality aware strategy to enhance the representation of images [14] and achieve incremental improvement. Wang *et al.* improve the generalizability by adding blur and random noise during the training phase [30]. However, this method relies on a large training set. Li *et al.* propose the Face X-Ray [21], which uses noise as well as error level analysis to extract the blending boundary of fake faces. Although it can achieve a certain level of generalizability, it is only applicable to specific manipulation types of fake faces; in other words, it achieves high generalizability between different face swap technologies, but is not suitable for detecting fake faces in the entire face synthesis.

Forgery Detection with Patches. Most of recent excellent face forgery detection methods are based on an overall image [13,21,26,30]. But they often ignore key local details in the image. To avoid this problem, many methods leverage the local perspectives instead of global detection [22,23,24,32]. Specifically, Zhou *et al.* propose a model to learn local features from patches [32]. Mayer *et al.* use the similarity between patches to judge whether the image is forgery [23,24]. Chai *et al.* propose a patch-based classifier to focus on local artifacts and obtain excellent generalization on the entire face forgery dataset [8]. They all assume that all patches cropped from an image belong to the same class as the input image, which is not consistent with reality (as shown in Fig. 1 (b)).

3 Approach

3.1 FDPT Architecture for Face Forgery Detection

As shown in Fig. 2, FDPT is a two-stream face forgery detection network (*i.e.*, a CNN stream and a CNN-GRU stream). It benefits from two different but complementary visual features. Specifically, the CNN stream learns on local subtle artifacts through the pre-processing of patch random transformation, and the CNN-GRU stream learns global spatial features between patches through patch sequence transformation. Then, a fusion strategy is used to improve the accuracy and generalization ability.

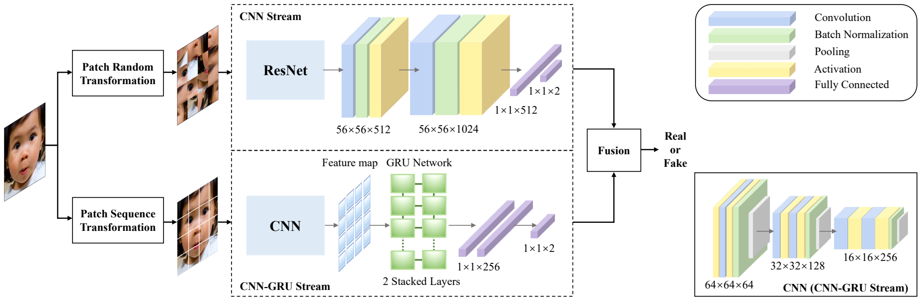

Fig. 2. FDPT architecture. The CNN stream focuses on local subtle artifacts through a patch random transformation. The CNN-GRU stream first obtains the global spatial features by passing input image through the patch sequence transformation. The CNN module of CNN-GRU stream is expanded in the lower right corner.

3.2 Local Subtle Artifacts Learning

The process of our local subtle artifacts learning stream (*i.e.*, the CNN stream) can be divided into four steps. First, we utilize a traditional data augmentation strategy to enrich the training set by randomly cropping the images in batch data. Second, a cropped image is self-shuffled by a patch random transformation method. Third, we use a CNN module to learn and extract the subtle artifacts of these shuffled images. In our method, the CNN stream is generic and could be implemented on any backbone feature extractor (*e.g.*, ResNet [17]). Finally, two convolution blocks and fully connected layers are employed to predict the authenticity of the input image, where the prediction is normalized by the Sigmoid function and we denote the normalized result as the prediction forgery score.

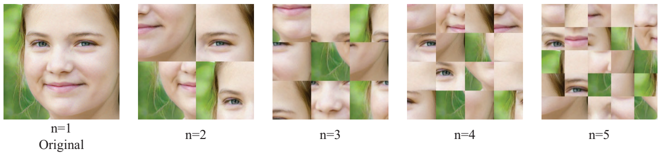

Fig. 3. Patch random transformation with different n.

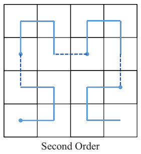

Fig. 4. Hilbert curves.

Patch Random Transformation. Most forgery detection methods extract global artifacts in an entire image to distinguish real from fake. However, there are different global artifacts in data generated by different face generators. As a result, most of the face forgery detection methods cannot be generalized across different generators. We leverage the patch random transformation to disturb the global artifacts existing in an entire image, thus to retain the local subtle artifacts and represent the face image in a more stable fashion. In patch random transformation, we divide the input image into patches, then randomly shuffle and assemble these patches into a new image. The purpose of these operations is to generate a new image after random transformation and force the stream to focus on local subtle features in the training stage.

Denote an input image by $\boldsymbol{I} \in \mathbb{R}^{W \times H \times 3}$ and the split parameter by n, where W and H are the width and the height of the image, respectively. The image is divide into $n \times n$ non-overlapped patches. The size of each patch is $\frac{W}{n} \times \frac{H}{n} \times 3$. These patches are reconstructed into a new image. Fig. 3 illustrates an original image and the transformed new image.

Note that there is no need to pre-process that in the training stage during the inference (*i.e.*, patch random transformation). The input image is randomly cropped and put into the CNN stream to get a prediction forgery score. If given a video, we extract multiple frames randomly and average the predictions.

3.3 Global Spatial Features Learning

As shown in Fig. 2, we employ a hybrid CNN-GRU module to extract global spatial features between patches. Given the patches split from an image, we organize patches in a specific order and expect GRU to capture the dependency among patches. Note that if patches are organized vertically or horizontally, sequential learning can not correlate them well due to long-distance between adjacent patches [5]. Inspired by the work in [5], we leverage the Hilbert curve to organize the patches and maintain the local correlation in the spatial domain. Then, we use a sequential module GRU to extract spatial features from the reordered patch images.

Patch Sequence Transformation. The patch sequence transformation consists of two steps: (1) we first split an image into several local patches on average and connecting patches in order, (2) we then adopt a sequential learning method (*i.e.*, GRU) to capture the global spatial features between the patches. Note that it is important to determine the order of patches fed to GRU. A common solution is to organize the patches either horizontally or vertically; however, such sequences do not better capture local information. For instance, if we connect patches horizontally, the adjacent patches in the vertical direction will be separated by an entire line of patches. Thus, it is difficult to learn the correlation characteristics between patches due to long-distance interval. To solve this problem, the space-filling curve is proposed. It maps data in multi-dimensional space to one-dimensional space and keeps the relevance of adjacent parts. We leverage the Hilbert curve to reconstructed patches in our method. Compared with other curves, the Hilbert curve maintains a better spatial local property, which is more favorable for sequential learning. The second-order Hilbert curve that we used is shown in Fig. 4.

The CNN-GRU stream works in a similar way as the CNN stream. More specifically, given an image $\boldsymbol{I} \in \mathbb{R}^{W \times H \times 3}$, we first divide it into $n \times n$ non-overlapped patches and the size of each patch is $\frac{W}{n} \times \frac{H}{n} \times 3$. We then transform the feature learning from a multi-dimensional to one-dimensional sequence by using a space-filling curve (*i.e.*, connect $n \times n$ patches into a sequence according to Hilbert curve). After that, we put each patch into a CNN module to extract the patch feature, where the CNN module can be any excellent backbone feature extractor. In our method, we simplified the CNN module to reduce the parameters during the training. Therefore, the CNN module in the CNN-GRU stream contains three calculation blocks and each block has two convolution layers, two activation layers, one batch normalization layer, and one max pooling layer. More detailed structure is shown in Fig. 2. Last, we feed the features of patches into the GRU module in the order of Hilbert curve to capture the correlation between patches. Our GRU module has two stacked GRU layers and three fully connected layers, and we normalize the outputs of the last layer using the Sigmoid function to predict the authenticity of the input image. As a result of these steps, the CNN-GRU stream can describe the correlation between patches and capture the global spatial representation by the space-filling curve.

3.4 Fusion Strategy

As mentioned in Section 3.2 and Section 3.3, the proposed CNN and CNN-GRU streams mine the local subtle artifacts and the global spatial features respectively. The two patch-related forgery features are different but complementary. Therefore, we adopt a fusion strategy to promote the final performance.

More specifically, we consider the prediction forgery score set $P = \{P_C, P_G\}$ in the evaluation phase, where $P_C \in [0,1]$ and $P_G \in [0,1]$ means the prediction forgery scores of the CNN stream and the CNN-GRU stream, respectively. The final forgery score P_{fusion} is calculated by Eq. (1), where the closer P_{fusion} is to 1, the more likely the input image is a forgery.

$$P_{fusion} = \frac{P_C + P_G}{2} \tag{1}$$

4 Evaluation

4.1 Experiment Setting

Dataset. We evaluate our method on two benchmark datasets: the entire face forgery detection dataset and the partial face forgery detection dataset. For the entire face forgery detection dataset, the real images come from FFHQ and the fake images are generated by StyleGAN/PGGAN (labeled as StyleGAN and PGGAN in corresponding datasets in the sequel). For the partial face forgery detection dataset, we use FaceForensics++ collected by [8]. FaceForensics++ contains 1,000 real videos and 4,000 manipulated fake videos, where these manipulated videos are generated by four face manipulation algorithms, namely, DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTexture (NT). Similarly, we refer to the corresponding datasets using the DF, F2F, FS, and NT labels in the sequel.

Evaluation metrics. For fair comparisons with other methods, we use the Accuracy score (ACC) to evaluate different methods. In addition, to evaluate the effectiveness of each component of FDPT, we also calculate the Area Under the Receiver Operating Characteristic Curve (AUC) in our ablation study.

Table 1. Ablation Study of FDPT.

ID	Stream	Patch Random	Patch Sequence	Entire Face Forgery		Partial Face Forgery	
		Transformation	Transformation	ACC	AUC	ACC	AUC
1	CNN	×		94.10	0.9352	93.23	0.9277
2		✓		98.25	0.9904	99.00	0.9932
3	CNN-GRU		×	98.17	0.9864	98.44	0.9805
4			✓	99.00	0.9941	99.17	0.9921
5	Fusion	×	×	95.03	0.9416	95.83	0.9524
6		×	✓	95.35	0.9456	96.50	0.9673
7		✓	×	98.53	0.9887	99.17	0.9940
8	FDPT	✓	✓	**99.87**	**0.9996**	**99.83**	**0.9989**

Implementation In our experiments, the input size of the CNN stream is 448 × 448 × 3. Then, we train the CNN stream using the stochastic gradient descent (SGD) method, where the learning rate is 10^{-3} and the number of epochs is 20. For the CNN-GRU stream, the input size is 256 × 256 × 3 and we train the CNN-GRU stream using SGD with the learning rate being 10^{-2}. We stop the training stage of the CNN-GRU stream at the 50th epoch. For more details, the code is available at `https://github.com/xihe7/PatchT/`.

During the training stage, we evaluate different parameters n in patch transformation and find that the results of $n = 4$ perform better. Thus, we set $n = 4$ in our remaining experiments.

4.2 Ablation Study

In order to evaluate the effectiveness of each component of FDPT, we conduct experiments on the entire face forgery and partial face forgery datasets separately and summarize results in Table 1.

Note that ID 1 and 2 in Table 1 represent the CNN stream without and with patch random transformation respectively, while ID 3 and 4 mean the CNN-GRU stream without and with patch sequence transformation respectively. We also apply the fusion strategy on different components to attempt to improve the detection results. The fusion results are shown in ID 5−7. Specifically, ID 5 refers to the fusion of two streams without any patch transformation. ID 6 and 7 respectively indicate only one of the patch transformation methods used in the weighted fusion two-stream network.

We observe that ACC and AUC are significantly improved due to patch random transformation; more specifically, the detection ACC increases by 4% and AUC is improved by 0.06 on the entire face forgery dataset. In addition, the ACC increases by more than 5% and AUC increases by 0.06 on FaceForensics++. Therefore, the effect of local artifacts on forgery detection is noticeable.

We also observe that ACC and AUC scores have slight improvement after using the Hilbert curve to organize patches. Although the improvement of quantitative results is relatively small, it is a meaningful improvement for face forgery

Table 2. Accuracy results (%) on entire face forgery and partial face forgery dataset.

Methods	Entire face forgery			Partial face forgery				
	StyleGAN	PGGAN	Mix	DF	FS	F2F	NT	FF++
ResNet [17]	91.65	97.01	94.10	95.53	92.77	93.02	96.92	93.23
Xception [10]	98.00	96.49	98.52	99.27	98.87	98.17	98.00	98.30
MesoNet [4]	91.90	98.70	94.17	92.19	93.75	90.62	88.97	86.67
MesoInception4 [4]	95.00	98.40	96.67	90.62	92.19	89.27	87.52	90.17
CNNp (p=0.1) [30]	98.85	99.70	99.63	82.07	96.77	97.13	77.50	89.88
CNNp (p=0.5) [30]	98.00	99.40	99.78	81.33	91.63	89.13	80.77	85.40
PatchW [8]	**100.0**	**100.0**	-	99.27	96.56	97.66	92.23	-
FDPT	99.85	99.80	**99.87**	**99.83**	**99.83**	**99.00**	**98.67**	**99.53**

detection. These results suggest that using the Hilbert sequence is conductive to improving detection results.

Furthermore, after the application of the fusion strategy, we observed that the results of ID $5 - 6$ are not as good as one of the two streams. In the two-stream network, when the performance of one stream is poor, the effect of simple fusion is not obvious. On the contrary (compared ID 7 with ID 2 and 3), when the network effect of both two streams is excellent, the final fusion performance is improved. Further as shown in ID 8, both ACC and AUC scores of FDPT are higher than its variants (*i.e.*, ID $1 - 7$). This suggests that the two features are complementary to each other. We can draw a conclusion that local artifacts play a key role in improving detection results, and the global representation is excellent in detection.

4.3 Comparison with Existing Methods

We train and evaluate existing methods including full MesoInception4 [4], MesoNet [4], ResNet [17], Xception [10], and a classifier proposed in [30] (CNNp) on the same datasets as the datasets FDPTuses. In addition, we also compare FDPT with the latest patch-based method [8] (PatchW).

Note that ResNet and Xception are advanced classification networks. Xception draws on the idea of depth-wise separable convolution and combines with the idea of ResNet. It is the leading classification network at present. On the other hand, MesoInception4, MesoNet, and CNNp of [30] are open-source face forgery detection algorithms. In particular, CNNp is one of the most recent works and is trained to detect CNN artifacts via blurring and compression augmentations. PatchW is the latest face forgery detection using patches. To better compare with it, we directly take the experimental results in [8].

4.3.1 Comparison results on entire face forgery dataset

We divide the fake face testing set into three types: PGGAN, StyleGAN, and their mixture. The results are shown in the left of Table 2. We observe that the accuracy (*i.e.*, ACC) of FDPT in three cases are 99.85%, 99.80%, 99.87%, respectively, with a noticeable improvement compared against the current methods and excellent classification networks (except PatchW). PatchW achieves almost 100% accuracy on known datasets; however, it is more likely to suffer from over-fitting. The significant performance gains mainly benefit from the two sets of complementary discriminative information learnt from patches, which contributes to FDPT's capability of learning more local details of face images.

4.3.2 Comparison results on partial face forgery dataset

Furthermore, we evaluate FDPT on different face manipulation techniques. We mix four face manipulation techniques of FaceForensics++ for training and evaluating together (*i.e.*, DF, FS, F2F, NT). They are all fake face video datasets, and we extract frames as fake face images in the experiments. We then train the model and evaluate it in four types of face manipulated dataset respectively. When training, in order to balance the proportion of real and fake datasets, the number of frames extracted from original videos is four times that of each face manipulation video. We summarize the results in the right of Table 2.

We observe that FDPT achieves a high detection accuracy and performs much better than other methods. FDPT achieves an accuracy of nearly 100% in all testing sets. Note that due to the two-stream structure, our FDPT model extracts multiple frames from each video, and each frame is detected by the CNN stream and CNN-GRU stream. The two streams complementarily make corrections to the error and make the final results perform better.

4.3.3 Analysis

By comparing the results in Section 4.3.1 and Section 4.3.2, we observe that the performance of both CNNp and PatchW degrade significantly on fake faces of FaceForensics++. Note that CNNp focuses on CNN-synthesized images and detects CNN artifacts to distinguish real and fake. However, the FaceForensics++ dataset lacks sufficient CNN artifacts, which is completely different from the entire face forgery dataset. Therefore, the accuracy results of the classifier methods drop significantly. In terms of PatchW, it assumes that all patches cropped from a fake image are considered as fake. Therefore, it has excellent detection accuracy in the task of entire face forgery detection. However, the detection results drop on the FaceForensics++ dataset due to the incorrectness which may be introduced by the assumption mentioned above.

The experimental results suggest that FDPT can be applied to all fake face image datasets, such as StyleGAN, PGGAN, and FaceForensics++, and consistently achieves state-of-the-art performances. Note that CNNp [30] and PatchW [8] are the latest face forgery detection methods. Both can only perform well on a specific type of fake face image. The results suggests that FDPT can achieve good universality and is suitable for various types of fake faces.

4.4 Generalizability

Forgery generation algorithms have been constantly evolving. Therefore, it is crucial to explore forgery detection methods that can achieve great generalizability;

Table 3. Generalization ability evaluation on entire face forgery dataset. Each model is trained on one dataset and evaluated on another unseen dataset.

Training set	StyleGAN		PGGAN	
Testing set	StyleGAN	PGGAN	PGGAN	StyleGAN
ResNet [17]	91.65	62.18	97.01	52.37
Xception [10]	98.00	65.87	96.49	74.33
MesoInception4 [4]	95.00	76.23	98.40	71.27
CNNp (p=0.1) [30]	99.85	86.92	99.70	85.27
CNNp (p=0.5) [30]	98.00	85.28	99.40	56.34
PatchW [8]	-	-	**100.0**	95.85
FDPT	**99.85**	**93.95**	99.80	**96.85**

in other words, the detection model trained with one face forgery dataset can be generalized to images generated by other new forgery manipulated techniques. We next investigate the generalizability performance of FDPT.

We first evaluate the generalizability on the entire face forgery dataset and summarize the results in Table 3. We train each model with one face forgery dataset and evaluate it on another one (unseen). Because there are many similarities between StyleGAN and PGGAN, it is relatively easy to implement generalization between them. We observe from Table 3 that many methods are prone to over-fitting and perform poorly on the unseen dataset. PatchW performs better than some other methods. The reason is that it utilizes small patches to ignore global differences between real from fake images and focus on shared generator artifacts. Our method FDPT achieves an accuracy of nearly 100% on the seen dataset, and is superior to all other methods on the unseen dataset. Even though our accuracy is a little less than PatchW on PGGAN (seen), we have significantly improved the generalizability on StyleGAN (unseen).

We then evaluate the generalizability of FDPT across four different generators of FaceForensics++. We train on each of the four manipulations and evaluate on the remaining three datasets. We summarize in Table 4 the results in terms of ACC with respect to each type of manipulated video. We observe that ACC of most methods is up to 99% on seen manipulation dataset (in gray); however, it drops drastically for unseen manipulations (in black). This is because the model learns the specific artifacts quickly and suffers from over-fitting. Therefore, it performs well on a given dataset and has poor generalizability on unseen datasets.

As shown in Table 4, our approach FDPT has better generalizability than PatchW in most cases, and performs better than other methods in all cases. Specifically, training on NT and F2F images can still achieve satisfactory generalizability on remaining datasets, and generalization to FS images is the hardest. PatchW is the latest patch-based forgery detection method, which focuses on local patches. The assumption PatchW used will bring errors when training on partial face forgery dataset; therefore, the generalizability of PatchW on FaceForensics++ is not as good as that on the entire face forgery dataset (as shown in Table 3).

Table 4. Generalizability on FaceForensics++. Each model is trained on one dataset and evaluated on the remaining datasets. ACC on the testing set corresponding to training images is colored in gray.

	Train on DF				Train on FS			
	DF	FS	F2F	NT	DF	FS	F2F	NT
ResNet [17]	95.53	52.43	53.15	52.42	58.83	92.77	53.16	51.04
Xception [10]	99.27	47.12	53.57	58.15	54.26	98.87	53.42	51.28
MesoInception4 [4]	94.32	51.34	60.17	58.27	51.64	96.19	55.32	49.46
CNNp (p=0.1) [30]	92.64	51.23	57.66	59.29	55.24	96.72	61.32	52.88
CNNp (p=0.5) [30]	91.46	55.99	56.06	54.02	57.29	97.65	60.04	51.16
PatchW [8]	99.14	58.74	**71.74**	74.99	**61.77**	97.13	62.00	53.44
FDPT	99.84	**61.05**	68.32	**75.63**	55.58	98.46	**73.43**	**53.81**
	Train on F2F				Train on NT			
	DF	FS	F2F	NT	DF	FS	F2F	NT
ResNet [17]	54.32	53.08	93.02	52.86	65.76	50.14	55.23	89.56
Xception [10]	66.08	53.15	96.17	55.07	69.67	48.55	56.79	93.60
MesoInception4 [4]	64.43	55.16	94.37	54.42	63.72	55.83	62.25	86.87
CNNp (p=0.1) [30]	66.24	59.04	97.83	62.97	69.27	49.88	67.04	88.50
CNNp (p=0.5) [30]	66.86	64.52	93.42	62.17	67.08	51.63	69.45	90.88
PatchW [8]	84.39	63.10	97.66	79.72	70.32	52.37	65.04	86.93
FDPT	**95.53**	**67.91**	98.15	**82.42**	**98.78**	**65.71**	**96.30**	98.92

Compared with PatchW, FDPT achieves higher generalizability. More specifically, FDPT achieves face forgery detection from more general evidences available from both local subtle artifacts and global spatial features. It is clear that the improved generalizability comes from the design of FDPT, namely, detecting discriminative information from local patch space instead of paying attention to the global features of specific manipulation artifacts.

4.5 Impacts of Image Quality

Images and videos in practical scenarios may be of lower quality (*e.g.*, due to compression), and many methods with good performance on high-quality images may suffer from low image quality.

Note that different quality is available in FaceForensics++. More specifically, FaceForensics++ provides the original output video dataset (RAW). Additionally, FaceForensics++ provides two different compression datasets: low-quality videos (LQ) and high-quality videos (HQ). HQ is produced with a light compression which is almost visually lossless (*i.e.*, constant rate quantization parameter equal to 23), while LQ produced with the quantization parameter being 40 [27].

We evaluate FDPT on FaceForensics++ with different image quality. The models are trained and evaluated on the HQ and LQ datasets for each of the four face manipulation scenarios. We summarize the results in Table 5.

We observe that FDPT outperforms other methods. First, FDPT performs well on the HQ datasets. More specifically, FDPT achieves 98.33%, 98.17%,

Table 5. Accuracy results (%) of FDPT on FaceForensics++ with different quality.

	HQ (High Quality)				LQ (Low Quality)			
	DF	FS	F2F	NT	DF	FS	F2F	NT
ResNet [17]	97.33	98.50	97.67	86.17	88.89	81.95	82.17	69.50
Xception [10]	97.17	96.33	95.67	88.50	90.57	82.35	83.67	73.83
MesoInception4 [4]	91.42	87.78	88.13	68.33	83.18	77.67	76.83	60.94
CNNp (p=0.1) [30]	96.29	93.58	94.66	75.15	90.95	86.53	81.62	64.27
CNNp (p=0.5) [30]	96.58	94.03	93.17	86.25	91.13	84.32	80.87	61.33
FDPT	**98.33**	**98.17**	**98.00**	**94.17**	**91.17**	**88.33**	**88.67**	**81.50**

98.00%, and 94.17% accuracy on DF, FS, F2F, and NT, respectively. The accuracy of FDPT on the DF, FS and F2F datasets is close to 100.0%. This suggests that FDPT can still perform excellent detection even when the light compression degrades the image quality. We also observe that the performance of FDPT drops on LQ dataset; more specifically, FDPT achieves 91.17%, 88.33%, 88.67%, and 81.50% accuracy on DF, FS, F2F, and NT, respectively. Although FDPT suffers from heavily compressed images, it can still achieve a high detection accuracy.

Note that compared with videos generated by DF, FS and F2F, fake videos generated by NT is a great challenge to detection models, due to its generated faces without noticeable forgery artifacts. Therefore, the accuracy results on the NT datasets are not as good as the results on the other three datasets. But, the accuracy of FDPT on NT is more than 90% in HQ and 80% in LQ, and FDPT still plays an excellent detection effect. This is consistent with the research results in [26] which proposed a forgery detection method specifically optimized for compressed videos.

5 Conclusion

In this paper, we propose FDPT, a general two-stream face forgery detection network based on patch transformation, to achieve higher generalizability. Specifically, FDPT consists of a CNN stream and a CNN-GRU stream. The first CNN stream enhances the capture of local subtle artifacts and avoids introducing the pseudo labels used in other methods. Then, the second CNN-GRU stream captures global spatial features between patches to strength the representation of the image. Finally, the fusion of these two streams improves the performance and generalization of our proposed models. The extensive experiments have shown that our model achieves state-of-the-art results on two different face forgery datasets. Moreover, FDPT remains effective when applied on unseen forgery datasets and achieves superior performance in the generalizability experiments.

Acknowledgments This work is supported in part by the Natural Science Foundation of China (NSFC) under Grant U19B2036.

References

1. DeepFakes (2019), https://www.github.com/deepfakes/faceswap
2. FaceApp (2019), https://faceapp.com/app
3. FaceSwap (2019), https://www.github.com/MarekKowalski/FaceSwap
4. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: Mesonet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–7 (2018)
5. Bappy, J.H., Simons, C., Nataraj, L., Manjunath, B., Roy-Chowdhury, A.K.: Hybrid lstm and encoder–decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing **28**(7), 3286–3300 (2019)
6. Berthelot, D., Schumm, T., Metz, L.: BEGAN: Boundary Equilibrium Generative Adversarial Networks. arXiv e-prints arXiv:1703.10717 (Mar 2017)
7. Bianchi, T., De Rosa, A., Piva, A.: Improved dct coefficient analysis for forgery localization in jpeg images. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2444–2447 (2011)
8. Chai, L., Bau, D., Lim, S.N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: European Conference on Computer Vision. pp. 103–120. Springer (2020)
9. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8789–8797 (June 2018)
10. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1251–1258 (July 2017)
11. Cole, S.: Ai-assisted fake porn is here and we're all fucked. Motherboard Tech by Vice (December 2017)
12. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security. pp. 159–164 (2017)
13. Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5781–5790 (June 2020)
14. Du, M., Pentyala, S., Li, Y., Hu, X.: Towards generalizable deepfake detection with locality-aware autoencoder. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 325–334 (2020)
15. Durall, R., Keuper, M., Pfreundt, F.J., Keuper, J.: Unmasking DeepFakes with simple Features. arXiv e-prints arXiv:1911.00686 (Nov 2019)
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27**, 2672–2680 (2014)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016)
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: Proceedings of International Conference on Learning Representations (ICLR) (2018)

19. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4401–4410 (June 2019)
20. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems NeurIPS 2018. pp. 10236–10245 (2018)
21. Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B.: Face x-ray for more general face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5001–5010 (June 2020)
22. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 3730–3738 (December 2015)
23. Mayer, O., Stamm, M.C.: Exposing fake images with forensic similarity graphs. IEEE Journal of Selected Topics in Signal Processing **14**(5), 1049–1064 (2020)
24. Mayer, O., Stamm, M.C.: Forensic similarity for digital images. IEEE Transactions on Information Forensics and Security **15**, 1331–1346 (2020)
25. Nataraj, L., Mohammed, T.M., Manjunath, B., Chandrasekaran, S., Flenner, A., Bappy, J.H., Roy-Chowdhury, A.K.: Detecting gan generated fake images using co-occurrence matrices. Electronic Imaging **2019**(5), 532–1–532–7 (2019)
26. Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J.: Thinking in frequency: Face forgery detection by mining frequency-aware clues. In: European Conference on Computer Vision. pp. 86–103. Springer (2020)
27. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Niessner, M.: Faceforensics++: Learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1–11 (October 2019)
28. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics (TOG) **38**(4), 1–12 (2019)
29. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2387–2395 (June 2016)
30. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8695–8704 (June 2020)
31. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: 2019 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (2019)
32. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1831–1839 (2017)