FingerprintNet: Synthesized Fingerprints for Generated Image Detection

Yonghyun Jeong[1], Doyeon Kim[2], Youngmin Ro[3],
Pyounggeon Kim[4,5], and Jongwon Choi[4⋆]

[1] Clova, NAVER
[2] LINE Plus
[3] Department of Artificial Intelligence, University of Seoul, Seoul, Korea
[4] Department of Advanced Imaging, Chung-Ang University, Seoul, Korea
[5] Samsung SDS

yonghyun.jeong@navercorp.com, doyeon.k@linecorp.com,
youngmin.ro@uos.ac.kr, {trytty,choijw}@cau.ac.kr

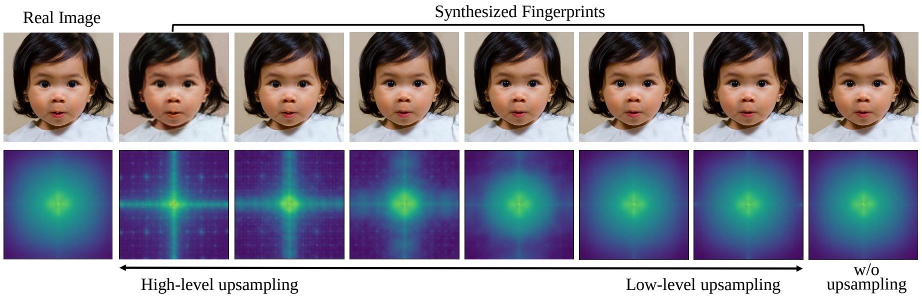


Fig. 1: **The synthesized fingerprints varying by the level of upsampling process.** Using the real images from FFHQ [32] as shown at the upper-left corner, the autoencoders can reconstruct images with various levels of upsampling processes, as shown in the columns of two to eight from the left. Their average 2D spectra in the second row show diverse synthesized fingerprints varying by the level of upsampling.

Abstract. While recent advances in generative models benefit the society, the generated images can be abused for malicious purposes, like fraud, defamation, and false news. To prevent such cases, vigorous research is conducted on distinguishing the generated images from the real ones, but challenges still remain with detecting the unseen generated images outside of the training settings. To overcome this problem, we analyze the distinctive characteristic of the generated images called 'fingerprints,' and propose a new framework to reproduce diverse types of fingerprints generated by various generative models. By training the model with the real images only, our framework can avoid data dependency on particular generative models and enhance generalization. With the mathematical derivation that the fingerprint is emphasized at the frequency domain, we design a generated image detector for effective

[⋆] Corresponding author.

training of the fingerprints. Our framework outperforms the prior state-of-the-art detectors, even though only real images are used for training. We also provide new benchmark datasets to demonstrate the model's robustness using the images of the latest anti-artifact generative models for reducing the spectral discrepancies.

1 Introduction

Based on the recent enhancement of the generative models, such as Generative Adversarial Networks (GAN) [19], it has become easy to obtain high-quality synthesized images [32, 33]. Many recent generative models can even transform the target images to include the specific properties of the users' choices [9,10,46, 59,60]. However, with technological improvement, the risk of maliciously abusing such images also rises, such as fraud, defamation, and fake news [36, 38, 44]. To prevent such cases, it is important to distinguish between the real images and the generated images [48].

Many recently generated image detectors have advanced to find the distinguishable features resulting during the image generation process [48]. For example, the checkerboard traces discovered in the frequency-level generated images are called the *'fingerprints*,' which are created during the upsampling estimation of the generator [6, 14, 15, 18]. Unfortunately, the appearance of the frequency-level fingerprints varies by the generative models and also by the object categories. Thus, when tested with the generated images of the unseen GAN models or object categories, the generated image detectors inevitably suffer from a performance decline [20]. In addition, recent studies have advanced to reduce the aliasing effect that occurs during the upsampling process of CNN in order to generate more realistic images. Such effort makes it challenging for the detectors to distinguish the fingerprints in generated images.

To overcome the issues, we suggest *FingerprintNet* composed of a fingerprint generator and a generated image detector. The overall framework utilizes the real images only and ignores the generated images by specific GAN models for training. Instead, the fingerprint generator reconstructs the real images to insert the general fingerprints to cover various GAN models. To synthesize the general fingerprints, we employ three mechanisms for the fingerprint generator, which include a random layer selection, a multi-kernel deconvolution layer, and a feature blender. Since the number of upsampling operations affects the appearance of the fingerprints as shown in Fig. 1, we control the number of upsampling operations of the fingerprint generator by applying the random layer selection. The multi-kernel deconvolution layer and the feature blender are employed to handle the diverse fingerprints due to various kernel sizes and the diverse amplitude of the fingerprints, respectively.

By using the real images and the images reconstructed from the fingerprint generator, we train the generated image detector to distinguish the generated images from the real ones. We mathematically derive the reasons why the fingerprint can be easily detected in the frequency-level domain, and accordingly, the input of our GAN detector is the magnitude spectrum to increase the detection performance. Our method is tested with various real-world scenarios to validate state-of-the-art generalization ability of our model in detecting even the unseen GAN models and object categories. Especially, our self-supervised detector shows a similar performance as the supervised detector when detecting the images generated from the recent generative model, such as Fréchet Inception Distance (FID) [21] and the anti-aliasing GAN models [7,29].

We can summarize our contributions as follows:

- Unlike the previous GAN detectors dependent on the specific GAN models, our model utilizes the self-supervised training method to obtain generalized detection ability and avoid data dependency.
- We propose a network that can generate various fingerprints, and a new way to train the detector by adjusting the amplitude of various fingerprints or perturbations.
- We provide a comprehensive analysis including visualizations and derivations on the artificial fingerprints observed in the frequency domain.
- We offer an extended benchmark dataset including the images generated by the latest anti-aliasing GAN models, which validate the state-of-the-art performance of our framework even for the unseen GAN models.

2 Related Work

We explore the previous literature on GAN image detection and the recent methods on GAN image creation, which have evolved to be more challenging to detect.

2.1 Generated Image Detection

The pixel-level characteristics in the GAN-based generated images can be used to identify the generated images. The identifiers can be referred to as the 'artifacts,' which are created due to upsampling process of the generator in GANs [6,14,20]. Some studies analyze the inconsistencies in blocking artifacts from JPEG compression [50,53], or demosaicing artifacts created by a color filter array [13,17]. Other image-based detection methods include an adaptable autoencoder-based neural network architecture for new target domains [12] and cross-model manipulation detection, such as JPEG and blur [52]. Recently, [49] proposed LRNet for detecting deepfakes based on temporal modeling.

Many generated image detectors focus on the unique patterns in the frequency spectra. [35] analyzed the artifacts in the spatial, frequency domain with the variance of the prediction residue, while [23] suggested Fast Fourier Transform [11] to distinguish image manipulations, such as JPEG compression. Also, a study by [43] employed frequency-based, GAN-specific detection using the artificial fingerprints, while [3] proposed a manipulation localization using the frequency domain correlation to find the forged areas. [18] analyzed the GAN-based artifacts using Discrete Cosine Transform [1], and [56] suggested studying the artifacts induced by the up-sampler of GANs. Also, [14,15] suggested exploiting the spectral distortions via Azimuthal integration. Recently, [27] suggested using bilateral high-pass filters for generalized detection, and [28] utilized the frequency-level perturbations for robust deepfake detection. Also, [58] proposed a new multi-attentional deepfake detection network, while [41] presented a spatial-phase shallow learning method for detecting artifacts of face forgeries.

Similar to our study, [26, 57] utilized the autoencoder to reconstruct the generated images. However, they ignored the difference of fingerprints among the various GAN models, while our study employs the additional mechanism to obtain the generality for the unseen GAN models.

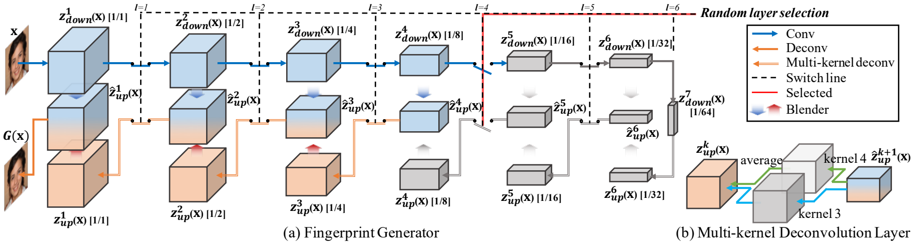
2.2 Advancement in Generative Models

Recently, generative models have become capable of creating images without the synthesized traces thanks to anti-aliasing, which refers to reducing the effect of artifacts in the generated images. Since it has now become more challenging to distinguish the generated images, it is important to analyze the latest generative models to upgrade the current detecting technologies. One of the popular anti-aliasing methods is to apply blur after deconvolution [30, 32, 33] and employ interpolation instead of the deconvolution layer [4, 10]. Recently, applying kaiser filter to the activation function is newly proposed for anti-aliasing by Karras et al. [31]. Generative models besides GANs have also advanced to generate high-quality images. An example is DDPM [22], which uses diffusion probabilistic models based on denoising score matching. Recently, ILVR [8] proposed a method to guide and condition the generative process of DDPM. Another example of high-quality generative models is NVAE [51], a deep hierarchical VAE using depth-wise separable convolutions and batch normalization. Some studies [5, 29] focused on reducing the spectral discrepancies in the spatial and spectral domains to obtain the anti-artifact characteristics. For example, SSD-GAN [7] enhanced GAN models to alleviate the loss of spectral information to generate the exact details of real images.

Since new manipulation methods quickly emerge, it is impractical to constantly update the detector's training in a supervised way [2]. Instead, it is much more practical to improve the generalization ability of generated image detectors. To improve the issue, some studies [2, 12, 24] adopted transfer learning, which utilizes a pre-trained model for another task using less amount of data. Recently, [34] proposed a method to perform domain adaptation on deepfake detection using transfer learning. However, transfer learning requires the pre-obtained knowledge on which GAN model is used for image synthesis, which makes it difficult to utilize for generalization of generated image detectors.

3 Fingerprint Generator

The purpose of the fingerprint generator is to mimic various kinds of fingerprints based on the reconstruction of the real images. The previous literature [14, 52]


$\mathbf{x}$
$G(\mathbf{x})$
$z^1_{down}(\mathbf{X})$ [1/1]
$z^2_{down}(\mathbf{X})$ [1/2]
$z^3_{down}(\mathbf{X})$ [1/4]
$z^4_{down}(\mathbf{X})$ [1/8]
$z^5_{down}(\mathbf{X})$ [1/16]
$z^6_{down}(\mathbf{X})$ [1/32]
$z^7_{down}(\mathbf{X})$ [1/64]
$\hat{z}^1_{up}(\mathbf{X})$
$\hat{z}^2_{up}(\mathbf{X})$
$\hat{z}^3_{up}(\mathbf{X})$
$\hat{z}^4_{up}(\mathbf{X})$
$\hat{z}^5_{up}(\mathbf{X})$
$\hat{z}^6_{up}(\mathbf{X})$
$z^1_{up}(\mathbf{X})$ [1/1]
$z^2_{up}(\mathbf{X})$ [1/2]
$z^3_{up}(\mathbf{X})$ [1/4]
$z^4_{up}(\mathbf{X})$ [1/8]
$z^5_{up}(\mathbf{X})$ [1/16]
$z^6_{up}(\mathbf{X})$ [1/32]
l=1
l=2
l=3
l=4
l=5
l=6
Random layer selection
Conv
Deconv
Multi-kernel deconv
Switch line
Selected
Blender
$z^k_{up}(\mathbf{X})$
average
kernel 4
$\hat{z}^{k+1}_{up}(\mathbf{X})$
kernel 3

(a) Fingerprint Generator

(b) Multi-kernel Deconvolution Layer

Fig. 2: **The overall architecture of FingerprintNet.** (a) depicts an example in which l is selected to 4 by *the random layer selection* requiring three times of upsampling. (b) shows details of the multi-kernel deconvolution layer.

mentioned that the fingerprints are created in generated images due to the upsampling process of the generative models. Based on the findings, we have developed the fingerprint generator using the autoencoder, which is designed to contain a number of upsampling process. The number of upsampling processes and the kernel size only affect the frequency of the fingerprints. Thus, to generate diverse kinds of fingerprints, we design the fingerprint generator to include the additional modules, such as blending of features, random selection of layers, as well as the multi-kernel deconvolution layers.

3.1 Overall Architecture

Fig. 2 shows the overall architecture of the fingerprint generator, which is composed of 7 blocks each containing the upsampling and downsampling convolution layers. First, the input image is turned into a small resolution by the convolution layers consecutively applied for image compression. The first layer with stride 1 is exempt, but all the other layers are required to compress the images with the stride of 2. Then, the resolution of the feature map becomes 1/64 of the original input size at the last layer. The compressed feature maps from k-th convolution layer can be defined as $z^k_{down}(\mathbf{x})$ for a given input image $\mathbf{x}$. Also, every convolution layer is paired with a ReLU activation function, which is applied after the convolution.

Then, in order to restore the feature map's resolution, we consecutively apply the deconvolution layers to the compressed features. Except for stride 1's last deconvolution layer to reconstruct the original image, every deconvolution layer enlarges the feature map's resolution twice as big as stride 2. The deconvolution layers consist of a variety of layers as follows: in the order of a transposed convolution, a batch normalization, a blur kernel for anti-aliasing, and a ReLU activation function layers. The k-th deconvolution layer's feature map is defined as $z^{k-1}_{up}(\mathbf{x})$. For easy understanding, we designate the same index of k to the same size of feature maps of $z^k_{down}(\mathbf{x})$ and $z^k_{up}(\mathbf{x})$. The fingerprint generator's output image is defined as $G(\mathbf{x})$ for the input of $\mathbf{x}$.

3.2 Training Loss

The fingerprint generator's training set is composed of only the real images. Also, additional to the conventional reconstruction loss of the autoencoder, the fingerprint generator's training loss takes the similarity losses to decrease the disparities between the feature maps of the corresponding deconvolution and convolution layers. Therefore, the fingerprint generator's training loss can be represented as:

$$\mathcal{L}(\mathbf{X}_r, G) = \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_r}\Big[||\mathbf{x} - G(\mathbf{x})||_2 + \sum_{k=1}^{6} ||z^k_{down}(\mathbf{x}) - z^k_{up}(\mathbf{x})||_2\Big], \tag{1}$$

where $\mathbf{X}_r$ indicates the training set composed of the real images. Based on the additional loss term using the latent feature maps, we can take $z^k_{down}(\mathbf{x})$ as the artifact-free version of the feature map of $z^k_{up}(\mathbf{x})$. These characteristics of the feature maps in the fingerprint generator are utilized for the feature blender, which is explained in Sec. 3.5.

3.3 Random Layer Selection

To handle the different numbers of upsampling operations in various GAN models, we employ the random layer selection in the fingerprint generator. At every training iteration, the module of the random layer selection randomly selects one value l from $\{1, 2, 3, 4, 5, 6\}$ according to the uniform distribution. Then, instead of using the entire layers, $z^l_{down}(\mathbf{x})$ is fed into l-th deconvolution layer to estimate $z^l_{up}(\mathbf{x})$. We set the feature maps from the remaining convolution and deconvolution layers with the indices larger than l as zero. Since l-th convolution and deconvolution layers have the equivalent resolutions and channel sizes, we can use the same weight parameters of the layers even after the random layer selection. As a result, while the original architecture of the fingerprint generator remains, the number of upsampling operations can vary to generalize the appearance of fingerprints in the reconstructed image.

3.4 Multi-kernel Deconvolution Layer

To consider the difference of fingerprints generated by various kernel sizes, we employ multiple kernels in the respective deconvolution layer. Among the various kernel sizes, we focus on the difference between the even and odd sizes of the kernels. Due to the constant stride size of 2 for each deconvolution layer, when estimating z^k_{up}, kernels of even sizes overlap by an even number of pixels, whereas kernels of odd sizes overlap by an odd number of pixels. Thus, instead of employing numerous kernel sizes, we use only two kernels where the sizes are set to 3 and 4, respectively. Especially, the two kernel sizes of 3 and 4 are conventionally used in most of the GAN models [9,10,30,32,60]. Then, one deconvolution layer contains two kernels, which are respectively applied to the input feature maps in parallel. Finally, z^k_{up} is obtained by estimating the average of the two feature maps resulting from the two kernels.

3.5 Feature Blender

The feature blender is employed to consider the various amplitude of fingerprints from the GAN models. Since the amplitude of fingerprints can be dependent on the input images, the feature blender augments the training samples by blending $z^k_{down}(\mathbf{x})$ and $z^k_{up}(\mathbf{x})$. According to our training loss (Eq. 1), the feature maps of the corresponding indices (*i.e.* $z^k_{down}(\mathbf{x})$ and $z^k_{up}(\mathbf{x})$) are trained to be similar to each other. Then, due to the absence of upsampling operations to estimate $z^k_{down}(\mathbf{x})$, $z^k_{down}(\mathbf{x})$ can be seen as the artifact-free feature map that is similar to $z^k_{up}(\mathbf{x})$. Thus, by blending the two feature maps, we can reduce the effect of fingerprints of $z^k_{up}(\mathbf{x})$ even while preserving its semantic information.

By using the feature blender, k-th deconvolution layer is fed by the blended feature map of $\hat{z}^k_{up}(\mathbf{x})$ instead of $z^k_{up}(\mathbf{x})$ that is the original feature map from the leading deconvolution layer. The blended feature map is obtained as follows:

$$\hat{z}^k_{up}(\mathbf{x}) = \mu_k z^k_{down}(\mathbf{x}) + (1 - \mu_k) z^k_{up}(\mathbf{x}), \tag{2}$$

where μ_k is a value randomly sampled from a Beta distribution of $\alpha = 1$ and $\beta = 1$. The value of μ_k is sampled repeatedly at every deconvolution layer and every training iteration. Thus, the fingerprint generator can generate the various amplitudes of fingerprints only by the unified model.

3.6 Fingerprint Generation

After the training of the fingerprint generator, we build the synthetic dataset containing the generated images from the fingerprint generator. During the dataset generation, we fix the indices l of the random layer selection by 1, 2, and 6. Thus, the generated images of our synthetic dataset contain various types of fingerprints. When $l = 2$ or $l = 6$, the fingerprints appear at $G(\mathbf{x})$ due to the upsampling operations, which can be used as an important characteristic to distinguish the generated images. Meanwhile, the generated images with $l = 1$ have had no upsampling operation and thus support robustness on the anti-aliasing GAN models. To improve robustness to various GAN models, the multi-kernel deconvolution layer and the feature blender remain in the dataset generation. Even though multiple images can be generated through the randomness of the feature blender, when one real image is given, we generate only three images respectively for one index l of the random layer selection. Thus, in the synthetic dataset, the quantity of generated images is three times that of real images.

4 Generated Image Detector

To classify the generated images from the real images, we utilize the additional CNN model, which is called the generated image detector. Before explaining the details of the generated image detector, we first derive mathematically the reason why the fingerprint of the generated images becomes distinctive in the frequency-level domain as referred by many studies [14, 18]. Based on the derivation, we also utilize the frequency spectrum as the input of the generated image detector. To further improve the robustness of the generated image detector, we employ the mechanism of mixed batch during its training.

4.1 Effect of Frequency-level Input

In this paper, we derive mathematically the reason why the fingerprints become distinctive in the frequency spectrum. A number of studies have confirmed that the images generated by GAN models contain the fingerprints appearing as the unique patterns in the frequency domain and utilizing those fingerprints can be the key to robust detection of the generated images [14, 18]. As shown in Fig. 1, the artificial fingerprints generated by the deconvolution layer are easily discovered in the 2D spectra. To ease the derivation, we consider the 1-D sequence in the following derivations.

As shown in Fig. 1, the fingerprints appear quite impulse train in the frequency spectrum. Thus, when a_m and T represent a scale factor for m-th impulse sequence and the period between the impulse sequences, respectively, we can represent the frequency-level fingerprints by the weighted impulse train as:

$$\mathcal{F}\{\mathbf{g}\}[k] = \sum_{m=-M}^{m=M} a_m \delta[k - mT], \tag{3}$$

where $m \in \{-M, ..., M\}$, $\mathbf{g}$ represents the pixel-level fingerprints, and $\mathcal{F}$ and $\delta[k]$ are the Fourier transformation function and an unit impulse sequence of the frequency component k, respectively.

Then, to acquire the pixel-level artifacts, we estimate the inverse Fourier transformation of the frequency-level fingerprints as follows:

$$\mathbf{g}[n] = \frac{2}{N} \sum_{m=0}^{M} |a_m| \cos\left(\frac{2\pi m T}{N} n + \alpha\right), \tag{4}$$

where N is the length of entire sequence and $\alpha = \arctan\big(\mathtt{Im}\{a_m\}/\mathtt{Re}\{a_m\}\big)$. The detailed derivation is given in Appendix A.

From the derivation, we can obtain two interesting characteristics of the pixel-domain artifacts. First, the fingerprints in the frequency domain are easier to discover because of their composition in the impulse train format, unlike the pixel-level artifacts based on the smooth trigonometric functions. Second, since M is smaller than $N/2$, we can find that the magnitude of the fingerprint in pixel-level domain cannot be larger than that in frequency-level domain according to the inequality of $g[n] < 2M|a_m|/N$ derived from Eq. 4. Thus, when we transform the input images into the frequency-level domain, the fingerprints can be emphasized to be detected easily. Therefore, we also employ the Fourier transform for the input of the detector.

4.2 Architecture of Detector

The next step after training the artificial fingerprint generator is the training of the generated image detector to discern between the generated images and the real images. As illustrated in Fig. 1 and the derivation in Sec. 4.1, it is effective to utilize the frequency-level analysis to investigate the artificial fingerprints. Therefore, we employ Fast Fourier Transform (FFT) [11] to transform the generated image $\hat{x} \in \hat{X} = \{G(x) | \forall x \in X\}$ of the artificial fingerprint generator into a 2D spectrum.

Our detector is based on ResNet-50 [39] for a fair comparison with the previous research [14,18,52]. In order to train the detector, we procure the generated images by reconstructing the real images from the training dataset. The training of the generated image detector can be challenging due to the unbalancing issue arising from the three generated images from one real image. To solve the issue, we have changed the sampling probability to extract the real images three times as much of the reconstructed images in a mini-batch.

4.3 Training Method with Mixed Batch

The generated image detector's training dataset can be divided into two categories: the generated images and real images. We sample an equal number of generated and real images to make one mini-batch. Then, we mix the sampled images instead of utilizing them directly to lessen data reliance on the category of real images, which is defined as *mixed images*. Also, the mixed images may minimize the noisy information in the generated images, improving the detector's tolerance against high-quality images from contemporary GAN models.

Every sample from the mini-batch is replaced with mixed samples, as stated by $\tilde{\mathbf{S}}$. $\tilde{\mathbf{Y}}$ stands for the labels that belong to the samples of $\tilde{\mathbf{S}}$. We assign 1 for real, and 0 for generated images. First, two images are randomly chosen from a mini-batch $\mathbf{S} = \{X_r, X_g\}$, which are indicated by $\mathbf{s}_i$ and $\mathbf{s}_j$, when we denote the sets of real images by $\mathbf{X}_r$ and generated images by $\mathbf{X}_g$. The mixed sample $\tilde{\mathbf{s}}_{(i,j)}$ and its label $\tilde{y}_{i,j}$ are retrieved by as follows:

$$\tilde{\mathbf{s}}_{(i,j)} = \lambda \mathbf{s}_i + (1-\lambda)\mathbf{s}_j, \quad \tilde{y}_{(i,j)} \quad = y_i y_j, \tag{5}$$

where λ is a mixing scale factor randomly selected from a Beta distribution with $a = 1$ and $b = 1$, and y_i and y_j are labels for $\mathbf{s}_i$ and $\mathbf{s}_j$, respectively. Only when the two real images are blended, we regard the mixed image to be the real image. Then, for each $\tilde{\mathbf{s}}_{(i,j)}$ and $\tilde{\mathbf{y}}_{(i,j)}$, $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{Y}}$ are established. The mixing mechanism may seem similar to Mixup method [55], but the notable difference of our work is the designing method of the augmented labels. While Mixup utilizes the augmented labels by integrating the original labels with the scales of the mixed samples, our feature blender considers the samples mixed with any fake images as the perfect fake images.

Then, we train the generated image detector (C) with a softmax cross-entropy loss, which can be represented as follows:

$$\mathcal{L}_C(\tilde{\mathbf{S}}) = \mathbb{E}_{(\tilde{\mathbf{s}},\tilde{y})\sim(\tilde{\mathbf{S}},\tilde{\mathbf{Y}})}[CE(\tilde{C}(\mathbf{s}), \tilde{y})], \tag{6}$$

where the softmax cross-entropy loss between the predictions of $\hat{y}$ and its associated ground-truth y is denoted by $CE(\hat{y}, y)$. Also, the training datasets are additionally supplemented with augmentation using JPEG compression and blur as provided in [52].

5 Experimental Results

5.1 Dataset

Through experiments, we compare the performance of each network based on the same data. Since the training settings have a strong impact on the analysis of the generated image detector, we adopt the same training settings as ProGAN [30] and utilize the real horse images of LSUN [54]. In contrast, the comparing models are trained with the 20 categories of ProGAN and the 20 categories of LSUN, which were used to train ProGAN. Also, for evaluation, we utilize the benchmark dataset [52] used for assessment of the generated image detector. The benchmark dataset includes several well-known unconditional GAN models including ProGAN [30], StyleGAN [32], and StyleGAN2 [33], and also a conditional GAN model, such as BigGAN [4]. We also employ the image-to-image translation models for testing, including CycleGAN [60], StarGAN [9], and GauGAN [45]. We utilize various GANs with human faces and various objects, and the real images used to train the GANs, including CelebA-HQ [37], CelebA [42], COCO [40], LSUN [54], and ImageNet [47].

Additionally, for evaluations, we utilize the recent GAN models that can generate images with spectral distributions similar to the real images, as well as state-of-the-art score-based generative models and variational autoencoders (VAE). For training, LSUN [54] and FFHQ [32] are used. For evaluations, we utilize the generative models in spatial and spectral domains including SSD-GAN [7], and SpectralGAN [29]. Also, we include the most recent score-based unconditional GAN, DDPM [22] and its conditional model, ILVR [8], as well as the most advanced unconditional VAE, NVAE [51], and state-of-the-art faceswap-based model, FICGAN [25].

5.2 Evaluation Metrics

For performance comparison, we employ the accuracy and average precision [16], which are the metrics commonly used in this field of study. To compare the generalization performance, we follow the suggestion of Wang [52] to use JPEG compression, which is known as the most effective method to test the generalization performance. Also, for the frequency-level analysis, we compare with Frank [18], Durall [14], and Jeong [27]. To evaluate cross-category performance, we compare with a self-supervised model [57].

5.3 Generalization Performance of Our Detector

To show the generalization ability of our detector, we perform two types of evaluations, which include the cross-category performance and the cross-model performance.

Table 1: **Cross-model performance with ablation study.**

Model	# of class	StyleGAN [32]		StyleGAN2 [33]		BigGAN [4]		CycleGAN [60]		StarGAN [9]		GauGAN [45]		Mean		Min	
	(real/fake)	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
Wang [52]	(1,1)	51.6	73.9	52.2	77.8	52.1	69.5	71.4	90.1	58.0	83.7	60.0	92.8	57.6	81.3	51.6	69.5
Durall [14]		64.1	58.6	69.3	62.9	55.4	52.9	69.6	62.8	95.4	91.5	57.5	54.0	68.6	63.8	55.4	52.9
Frank [18]		68.5	80.7	60.8	77.3	72.1	63.0	57.6	56.6	80.1	76.3	74.0	95.5	68.9	74.9	57.6	56.6
Jeong [27]		66.9	72.1	64.7	73.8	80.2	83.9	66.4	82.6	90.4	99.4	82.8	96.2	75.2	84.7	64.7	72.1
Wang [52]	(2,2)	52.8	82.8	75.7	96.6	51.6	70.5	58.6	81.5	51.2	74.3	53.6	86.6	57.3	82.1	51.2	70.5
Durall [14]		63.5	58.1	68.7	62.4	56.4	53.5	63.5	58.2	89.8	83.1	56.5	53.5	66.4	61.5	56.4	53.5
Frank [18]		70.8	83.8	61.2	75.6	74.9	76.2	74.8	76.8	91.7	97.5	89.2	98.4	77.1	84.7	61.2	75.6
Jeong [27]		71.6	74.1	77.0	81.1	82.6	80.6	86.0	86.6	93.8	80.8	69.6	90.8	80.1	82.3	69.6	74.1
Wang [52]	(4,4)	63.8	91.4	76.4	97.5	52.9	73.3	72.7	88.6	63.8	90.8	63.9	92.2	65.6	89.0	52.9	73.3
Durall [14]		63.9	58.4	69.0	62.7	58.5	54.7	69.6	63.1	99.0	98.1	57.0	53.8	69.5	65.1	57.0	53.8
Frank [18]		72.2	82.1	64.2	80.1	68.9	82.4	53.7	66.2	89.1	99.2	65.3	90.3	68.9	83.4	53.7	66.2
Jeong [27]		76.9	75.1	76.2	74.7	84.9	81.7	81.9	78.9	94.4	94.4	65.5	94.0	80.0	83.1	65.5	74.7
Wang [52]	(20,20)	71.4	96.3	67.5	93.4	60.9	83.3	83.8	94.3	84.6	93.6	79.3	98.1	74.6	93.2	60.9	83.3
Durall [14]		64.7	59.0	69.2	62.9	59.4	55.3	66.9	60.9	98.5	97.1	57.2	53.9	69.3	64.9	57.2	53.9
Frank [18]		81.8	91.7	71.4	93.0	76.0	87.8	62.8	77.3	96.9	99.4	73.9	93.1	77.1	90.4	62.8	77.3
Jeong [27]		73.0	83.9	62.7	75.9	78.1	94.8	60.5	85.6	100.0	100.0	68.7	97.4	73.8	89.6	60.5	75. 9
w Mix up	(1,0)	69.0	81.4	68.2	80.6	79.2	94.1	62.7	84.2	98.8	100.0	69.5	89.1	74.6	88.2	62.7	80.6
w/o Similar loss		71.3	81.3	76.6	87.6	76.9	89.6	59.4	95.8	99.1	99.3	65.7	96.8	74.8	91.7	59.4	81.3
w/o Rand. select.		56.4	53.2	57.9	77.3	54.2	69.6	51.9	41.5	86.4	89.6	53.1	75.9	60.0	67.9	51.9	41.5
w/o Multi. kernel		68.5	82.0	71.5	88.8	67.2	91.8	62.6	82.1	98.3	99.7	59.9	78.2	71.3	87.1	59.9	78.2
w/o Mixed batch		69.0	81.4	68.2	80.6	79.2	94.1	62.7	84.2	98.8	100.0	69.5	89.1	74.6	88.2	62.7	80.6
w/o Feat. Blender		78.6	89.7	73.5	88.7	73.9	86.3	63.0	88.8	98.9	99.8	61.7	91.4	74.9	90.8	61.7	**86.3**
w/o FFT		92.1	97.4	89.1	95.9	66.8	65.7	64.0	74.7	99.3	100.0	58.1	63.8	78.2	82.9	58.1	63.8
Ours		74.1	85.3	89.5	96.1	85.0	94.8	71.2	96.9	99.9	100.0	75.9	90.9	**82.6**	**94.0**	**71.2**	85.3

Table 2: **Comparison result with self-supervised manner.**

Model	Train category						Mean
	Apple	Horse	Orange	Summer	Winter	Zebra	
AutoGAN [57]	76.1	97.4	67.7	97.2	68.1	78.6	80.9
SelfDetector [26]	78.7	95.3	78.3	90.8	80.8	97.7	86.9
Ours	96.1	95.8	88.4	95.0	91.1	96.3	**93.8**

Cross-model Performance.

Table 1 shows the results of the first experiment to test the cross-model performance of the generated image detectors. We compare with the previous studies used for the comparison of cross-model performance: Wang [52], Frank [18], Durall [14], and Jeong [27]. Each of them is trained using 1 to 20 categories generated by ProGAN [30], and tested with the generated images of seven other generative models. In contrast, our self-supervised generated image detector is trained with real *horse* images only. Even with the seriously limited setting where no generated images of GAN models are used, our generated image detector achieves the highest accuracy and average precision.

To show the component-wise effectiveness of our framework, we perform the ablation studies in the cross-model experiments. As shown in the bottom section of Table 1, the averaged performance dramatically drops when only one of the components is missing, which verifies the importance of the respective component to cover the various types of generative models. For the first row of our ablation tests (*w Mixup*), we use the Mixup method [55] to replace our training loss, which supports the effectiveness of our novel training loss to recognize the subtle artifacts and improve detection accuracy. Especially, when the random selection module is removed, the amount of performance decline is substantial,

Table 3: **Robustness to anti-artifact GANs and SOTA models.**

Model	# of class (real/fake)	Test Models													
		Anti-artifact GANs						State-of-the-art generative models							
		SSD-GAN [7]		SpectralGAN [29]		Mean		DDPM [22]		ILVR [8]		NVAE [51]		FICGAN [25]		Mean	
		Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
Wang [52]	(1,1)	50.3	95.3	50.0	95.6	50.2	95.5	49.3	31.8	49.3	32.0	49.7	33.3	49.3	32.6	49.4	32.4
Durall [14]		53.1	51.7	73.2	68.0	63.2	59.9	49.9	49.9	55.6	53.0	56.1	53.2	54.7	52.4	54.1	52.1
Frank [18]		96.2	99.6	96.2	99.7	96.2	99.7	68.7	82.6	70.8	84.3	78.0	89.2	76.9	88.9	73.6	86.3
Jeong [27]		89.5	95.0	89.0	89.4	89.3	92.2	78.7	86.5	79.8	87.8	81.2	86.4	89.4	96.2	82.3	89.2
Wang [52]	(20,20)	75.8	84.6	87.9	89.5	81.9	87.1	68.4	79.2	75.8	84.6	71.0	80.9	76.6	86.5	73.0	82.8
Durall [14]		68.7	62.4	44.5	48.3	56.6	55.4	57.5	54.1	57.6	54.1	57.7	54.2	57.0	53.7	57.5	54.0
Frank [18]		96.2	100.0	96.4	100.0	96.3	**100.0**	88.4	93.2	86.5	93.6	92.4	96.2	82.7	92.2	**87.5**	93.8
Jeong [27]		84.2	99.9	84.2	99.5	84.2	99.7	83.5	93.4	83.0	92.4	82.0	91.5	83.6	94.1	83.0	92.9
Ours	(1,0)	96.4	99.2	98.5	99.9	**97.5**	99.6	78.7	92.3	84.0	95.4	91.4	97.1	92.6	98.5	86.7	**95.8**

which indicates the importance of considering the various numbers of upsampling operations to obtain diverse fingerprints. Based on the discovery, we can conclude that it is necessary to diversify the number of upsampling operations for diversity in generated fingerprints.

Cross-category Performance. We conduct a cross-category experiment to compare accuracy in the same test settings as [57]. Using the generated images of the same GAN model, we train the generated image detectors with only one object category and test with the entire object categories to evaluate the generalization performance. Table 2 shows the test results using the 6 classes (apple, horse, orange, summer, winter, and zebra) of the generated image detectors trained with each category of CycleGAN [60]. Compared to the existing model trained in a self-supervised manner [26, 57], our model shows superior performance in generalized detection.

5.4 Generalization for Recent Generative Models

According to [7,29], the spectral distributions of images are known to vary by the last de-convolution layer, and it can decline the performance of generated image detectors. Based on that, we assess the model's robustness on the synthesized images of anti-artifact generative methods to reduce the spectral discrepancies. For training of each generative model, the real horse images of LSUN [54] are utilized, as in Sec. 5.3. The left section of Table 3 shows the performance of each detector when evaluated with the generated images of the anti-artifact models. Our model and Frank [18] show the most superior performance compared to others. Jeong [27] shows declined performance due to its high-pass filter, since the high-frequency components are modified in the generated images of the anti-artifact models. Also, Durall [14] also suffers from declined performance due to the reduced spectral discrepancies in frequency distributions of images.

Technological advancement in generative models has not only affected GANs but also the score-based diffusion probabilistic models and variational autoencoders. Thus, we additionally evaluate the performance of state-of-the-art generative methods, including DDPM [22], ILVR [8], NVAE [51], and FICGAN [25].

Table 4: **Color manipulation performance.**

Model	Original		Hue		Brightness		Saturation		Gamma		Contrast		Mean	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
Wang [52]	99.9	100.0	73.9	81.3	61.8	74.7	74.3	84.4	70.2	83.2	66.6	79.7	74.5	83.9
Frank [18]	95.2	96.5	85.5	97.2	84.2	97.2	91.2	98.0	85.4	97.4	84.3	96.7	87.6	97.2
Durall [14]	86.2	93.4	86.2	81.9	85.9	81.9	86.2	81.9	85.1	80.8	85.2	81.2	85.8	83.5
Jeong [27]	97.0	98.1	92.0	97.8	92.0	97.9	91.9	96.7	91.7	96.8	92.4	98.1	92.8	97.6
Ours	97.4	99.8	97.1	99.8	89.4	96.6	94.1	99.2	96.9	99.9	89.6	97.7	**94.1**	**98.8**

DDPM is the most well-known diffusion probabilistic model, and ILVR is a conditioning method for DDPM. Also, NVAE is the most recent unconditional variational autoencoder for high fidelity synthesized images, while FICGAN is a face-swapping method for high-quality deepfake images. The right section of the Table 3 shows the performance of each model evaluated with the images generated by state-of-the-art generative models. Our detector achieves stable performance even with the face-swap model, FICGAN. Since other models trained in a supervised manner focus on the distributions of GAN training, they suffer from a decline in performance when tested with non-GAN generative models with different distributions, such as DDPM and VAE.

5.5 Color manipulation performance

We conduct an experiment to evaluate the detector's robustness on color manipulated images, using the same settings of the color manipulation experiments of Jeong [27]. First, we resize the images from 1024×1024 to 256×256, then modify colors for assessment. Manipulations in hue, brightness, saturation, gamma, and contrast modify the overall distribution of images to make challenging conditions for detectors to work [27]. The hue factor is the amount of shift in the hue channel by 0.2, while brightness, saturation, gamma, and contrast are adjusted by 1.3, respectively. Table 4 indicates the variance in detecting performance when images are manipulated and the characteristics of the artifacts have changed. For a fair comparison, we apply the supervised learning to train the detector based on ProGAN [30] face and FFHQ [32] as in [27], and do not apply the center crop. The experimental results validate that the frequency-based methods [14, 18, 27] including ours are more robust to color manipulations compared to image-based method [52].

5.6 Visualization

Fig. 3 shows the resemblance between the reconstructed average 2D spectra by adjusting the level of the upsampling process and those generated by the actual GAN models. By adjusting the level of downsampling in autoencoders, we can observe the close resemblance among the reconstructed patterns generated by each GAN model in FFT. Also, we can confirm that the transposed convolution-based GANs, including StyleGAN, StyleGAN2, CycleGAN, and StarGAN, gen-

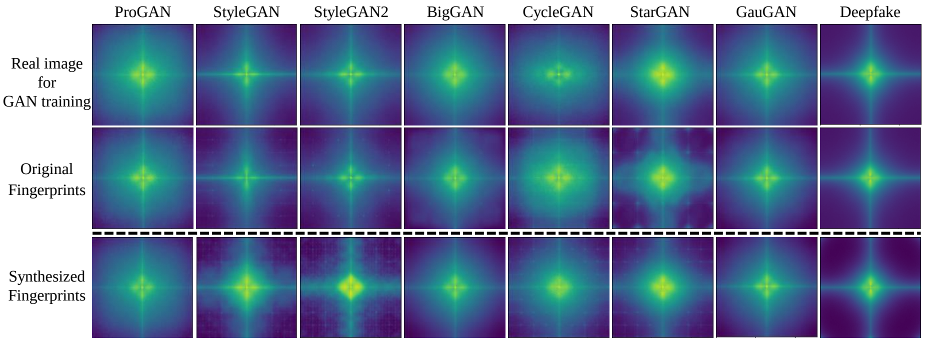


Fig. 3: **The averaged spectra of the real images, images from GAN models, and images from fingerprint generator.** The first row shows the averaged spectra of the real images used for training the GAN models, while the middle row shows those of the generated images from the GAN models. The last row shows the averaged spectra where we can obtain the highest resemblance between the spectra of the generated images and the synthesized fingerprints.

erate more distinct fingerprints, which are close to the spectrum of the reconstructed data with a high level of upsampling. From the visualization, we can confirm that the fingerprints from our fingerprint generator can be effective for the training of the generated image detector. We provide every visualization result in Appendix B.

6 Conclusion

We propose a novel framework composed of a fingerprint generator and a generated image detector for robust generalization. First, we analyze the diverse types of fingerprints in generated images and develop a fingerprint generator, which can synthesize and insert the fingerprints on real images for high-quality training data. Based on the analysis, we newly introduce a training method using real images only for generalized detection and validate its efficacy through robust performance of our model. Surpassing others trained in a supervised manner, our model achieves impressive performance in zero-shot learning, even when tested with unseen categories and GAN models. Also, we include the most recent anti-artifact generative models for evaluation and verify our model's consistent performance. We hope that the suggested framework can be enhanced in the future to manage the unexpected developments of new generative models by using the extra modules to address the additional properties of their fingerprints.

Acknowledgments. It was supported by Samsung SDS and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program(Chung-Ang University); 2021-0-01778, Development of Human Image Synthesis and Discrimination Technology Below the Perceptual Threshold; 2021-0-02067, Next Generation AI for Multi-purpose Video Search).

References

1. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (1974)
2. Aneja, S., Nießner, M.: Generalized zero and few-shot transfer for facial forgery detection. arXiv preprint arXiv:2006.11863 (2020)
3. 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)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=B1xsqj09Fm`
5. Chandrasegaran, K., Tran, N.T., Cheung, N.M.: A closer look at fourier spectrum discrepancies for cnn-generated images detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
6. Chen, S., Yao, T., Chen, Y., Ding, S., Li, J., Ji, R.: Local relation learning for face forgery detection. arXiv preprint arXiv:2105.02577 (2021)
7. Chen, Y., Li, G., Jin, C., Liu, S., Li, T.: Ssd-gan: Measuring the realness in the spatial and spectral domains. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
8. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: Ilvr: Conditioning method for denoising diffusion probabilistic models. In: IEEE International Conference on Computer Vision (2021)
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: IEEE Conference on Computer Vision and Pattern Recognition (2018)
10. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
11. Cooley, J.W., Lewis, P.A., Welch, P.D.: The fast fourier transform and its applications. IEEE Transactions on Education (1969)
12. Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., Verdoliva, L.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv (2018)
13. Dirik, A.E., Memon, N.: Image tamper detection based on demosaicing artifacts. In: 2009 16th IEEE International Conference on Image Processing. pp. 1497–1500 (2009)
14. Durall, R., Keuper, M., Keuper, J.: Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions. In: IEEE Conference on Computer Vision and Pattern Recognition. Seattle, WA, United States (2020)
15. Durall, R., Keuper, M., Pfreundt, F.J., Keuper, J.: Unmasking deepfakes with simple features. arXiv preprint arXiv:1911.00686 (2019)
16. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**, 303–338 (2010)
17. Ferrara, P., Bianchi, T., De Rosa, A., Piva, A.: Image forgery localization via fine-grained analysis of cfa artifacts. IEEE Transactions on Information Forensics and Security **7**(5), 1566–1577 (2012)
18. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: International Conference on Machine Learning. pp. 3247–3258. PMLR (2020)

19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
20. Gragnaniello, D., Cozzolino, D., Marra, F., Poggi, G., Verdoliva, L.: Are gan generated images easy to detect? a critical analysis of the state-of-the-art. arXiv preprint arXiv:2104.02617 (2021)
21. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (2017)
22. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Neural Information Processing Systems (NeurIPS) (2020)
23. Huang, D.Y., Huang, C.N., Hu, W.C., Chou, C.H.: Robustness of copy-move forgery detection under high jpeg compression artifacts. Multimedia Tools and Applications **76**(1), 1509–1530 (2017)
24. Jeon, H., Bang, Y.O., Kim, J., Woo, S.: T-gd: Transferable gan-generated images detection framework. In: International Conference on Machine Learning. pp. 4746–4761. PMLR (2020)
25. Jeong, Y., Choi, J., Kim, S., Ro, Y., Oh, T.H., Kim, D., Ha, H., Yoon, S.: Ficgan: Facial identity controllable gan for de-identification. arXiv preprint arXiv:2110.00740 (2021)
26. Jeong, Y., Kim, D., Kim, P., Ro, Y., Choi, J.: Self-supervised gan detector. arXiv preprint arXiv:2111.06575 (2021)
27. Jeong, Y., Kim, D., Min, S., Joe, S., Gwon, Y., Choi, J.: Bihpf: Bilateral high-pass filters for robust deepfake detection. arXiv preprint arXiv:2109.00911 (2021)
28. Jeong, Y., Kim, D., Ro, Y., Choi, J.: Frepgan: Robust deepfake detection using frequency-level perturbations. arXiv preprint arXiv:2202.03347 (2022)
29. Jung, S., Keuper, M.: Spectral distribution aware image generation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
30. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), `https://openreview.net/forum?id=Hk99zCeAb`
31. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. Neural Information Processing Systems (NeurIPS) (2021)
32. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
33. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. CoRR **abs/1912.04958** (2019)
34. Kim, M., Tariq, S., Woo, S.S.: Fretal: Generalizing deepfake detection using knowledge distillation and representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1001–1012 (2021)
35. Kirchner, M.: Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In: ACM workshop on Multimedia and security. pp. 11–20 (2008)
36. Kwon, P., You, J., Nam, G., Park, S., Chae, G.: Kodf: A large-scale korean deepfake detection dataset. arXiv preprint arXiv:2103.10094 (2021)
37. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

38. Lee, S., Tariq, S., Shin, Y., Woo, S.S.: Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. Applied Soft Computing **105**, 107256 (2021)
39. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (2019)
40. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
41. Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., Yu, N.: Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 772–781 (2021)
42. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: International Conference on Computer Vision (December 2015)
43. Marra, F., Gragnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: IEEE Conference on Multimedia Information Processing and Retrieval. pp. 506–511. IEEE (2019)
44. Nguyen, T.T., Nguyen, C.M., Nguyen, D.T., Nguyen, D.T., Nahavandi, S.: Deep learning for deepfakes creation and detection. arXiv preprint arXiv:1909.11573 (2019)
45. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2337–2346 (2019)
46. Pidhorskyi, S., Adjeroh, D.A., Doretto, G.: Adversarial latent autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14104–14113 (2020)
47. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
48. Sun, K., Liu, H., Ye, Q., Liu, J., Gao, Y., Shao, L., Ji, R.: Domain general face forgery detection by learning to weight (2021)
49. Sun, Z., Han, Y., Hua, Z., Ruan, N., Jia, W.: Improving the efficiency and robustness of deepfakes detection through precise geometric features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3609–3618 (2021)
50. Tralic, D., Petrovic, J., Grgic, S.: Jpeg image tampering detection using blocking artifacts. In: International Conference on Systems, Signals and Image Processing. pp. 5–8. IEEE (2012)
51. Vahdat, A., Kautz, J.: NVAE: A deep hierarchical variational autoencoder. In: Neural Information Processing Systems (NeurIPS) (2020)
52. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot...for now. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
53. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: IEEE International Conference on Multimedia and Expo. pp. 12–15. Ieee (2007)
54. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)

55. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization (2018)
56. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: IEEE International Workshop on Information Forensics and Security. pp. 1–6 (2019)
57. 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. IEEE (2019)
58. Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., Yu, N.: Multi-attentional deepfake detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2185–2194 (2021)
59. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: European conference on computer vision. pp. 592–608. Springer (2020)
60. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (2017)