BiHPF: Bilateral High-Pass Filters for Robust Deepfake Detection

Yonghyun Jeong[1], Doyeon Kim[1], Seungjai Min[1], Seongho Joe[1], Youngjune Gwon[1], Jongwon Choi[2*]
[1]Samsung SDS, Seoul, Korea
yhyun.jeong, dy31.kim, seungjai.min, drizzle.cho, gyj.gwon@samsung.com
[2]Dept. of Advanced Imaging, Chung-Ang University, Seoul, Korea
choijw@cau.ac.kr

Abstract

The advancement in numerous generative models has a two-fold effect: a simple and easy generation of realistic synthesized images, but also an increased risk of malicious abuse of those images. Thus, it is important to develop a generalized detector for synthesized images of any GAN model or object category, including those unseen during the training phase. However, the conventional methods heavily depend on the training settings, which cause a dramatic decline in performance when tested with unknown domains. To resolve the issue and obtain a generalized detection ability, we propose Bilateral High-Pass Filters (BiHPF), which amplify the effect of the frequency-level artifacts that are generally found in the synthesized images of generative models. Also, to find the properties of the general frequency-level artifacts, we develop an additional method to adversarially extract the artifact compression map. Numerous experimental results validate that our method outperforms other state-of-the-art methods, even when tested with unseen domains.

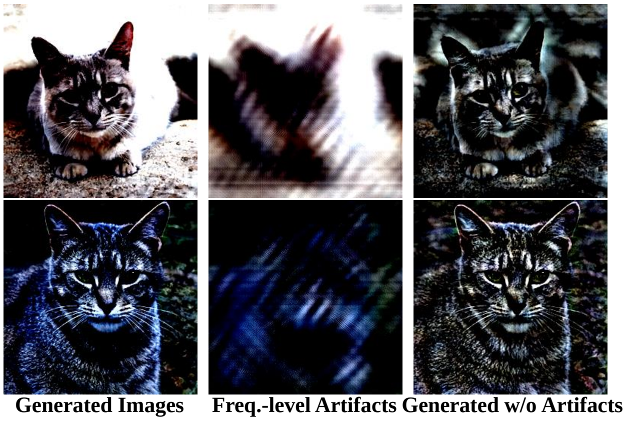


Figure 1: **Visualization of the frequency-level artifacts discovered in synthesized images.** The artifacts are extracted by the proposed classification network adversarially trained for a frequency-level compression map. Based on the analysis of the averaged artifacts, we develop a novel mechanism using Bilateral High-Pass Filters (BiHPF), which improve the cross-domain performance for detecting the synthesized images.

1. Introduction

Recently, the advancement of generative models including Generative Adversarial Networks (GAN) [22] has allowed the easy generation of realistic synthesized images. Unfortunately, these generated images can be abused with malicious purposes and ultimately bring detrimental consequences in societal, political, and economic aspects, such as fraud, defamation, and fake news [18, 19, 26, 27, 34]. Thus, it is highly important to develop a robust detection model to protect our society [19, 39].

Many studies have been conducted for detecting the generated images by GAN models. Various literature including [11, 12, 14, 41] have tackled the entire image synthesis of diverse categories, while others focus on manipulations on the human face only [1, 8, 28, 29, 32, 33, 42]. However, most of the prior literature suffers from a dramatic decline in performance when tested with unknown domains outside of the training data, due to their domain-specific detection. Since it is easy to switch the target domain of the generated images with malicious purposes, it is important to achieve robust detection even in unseen domains [39]. This capability can be defined as *cross-domain performance*. In this paper, the domains include various properties, such as the categories of the generated subjects, the color manipulations, and GAN models. As the range of such domains rapidly expands with technological advancement, we believe it is more important to focus our attention to the generalized detection of the synthesized images by GAN models, rather than focusing on the facial manipulations only.

[*]Corresponding author.

It has been confirmed by several previous studies [14,41, 45] that the synthetic images of generative models contain unique artifacts caused by the upsampling of GAN pipeline. We find that these artifacts in the frequency spectrum can become a key factor in developing a robust detector. Thus, we develop an adversarial framework to train the 'artifact compression map,' which is multiplied into the frequency spectrum to reduce the artifacts. As shown in the second column of Fig. 1, the intensity of the artifact compression map represents the degree of artifacts contained in the frequency spectrum. By analyzing the trained compression map, we obtain two practical insights. First, the artifacts have large magnitudes in the high-frequency components, as discovered in our numerous experiments in the frequency spectrum. Second, the artifacts are located in the surrounding background of the image rather than the central region, as observed in the pixel-level image transformed from the frequency-level compression map. Through experiments, we notice the two important discoveries are generally observed in various settings, and utilizing those discoveries can prevent domain-specific detection and enhance cross-domain detecting performance.

Thus, we design a simple mechanism called Bilateral High-Pass Filters (BiHPF) to emphasize the discovered properties of the artifacts. BiHPF consists of two High-Pass Filters (HPF): the frequency-level HPF for amplifying the magnitudes of the artifacts in the high-frequency components, and the pixel-level HPF for emphasizing the pixel values in the surrounding background in the pixel domain. Then, the classification model utilizes the BiHPF-processed magnitude spectrum map as the input of the network. To validate the overall performance of the proposed algorithm, we consider various domains including the unseen categories of diverse subjects, color settings, and GAN models. Numerous experiments confirm that our proposed algorithm significantly improves the detecting performance.

2. Related Work

The detection methods for CNN-based manipulations and synthesized images by GAN models can be categorized into two major schemes based on the input data: pixel-based detection and frequency-based detection.

Pixel-based detection. Some of the previous studies [5,9,13,23,35,36,40,43] target image pixels as the input data for the detection of image forgery. The earlier studies focus on specific conditions, such as compression and lighting. Some analyze the inconsistencies in blocking artifacts generated during JPEG compression [40,43], while others focus on the 3D lighting to assess the inconsistencies in lighting conditions of the subjects detect manipulations [5,23,35,36]. To distinguish more types of tampering operations, [9,13] suggest analyzing the demosaicing artifacts generated by color filter array processing in tampered images; however, this method is inapplicable to resized images, since the artifacts disappear during resizing.

Recently, with the rise of deepfakes, most studies focus on the temporal properties, such as facial features [1, 15, 32, 33], incoherent head poses [42], and lack of eye-blinking [29]. Various studies [10, 30, 37] provide large-scale datasets and apply diverse methods of image forensics for face manipulations. To broaden the scope of detection, [3] design a specific convolutional layer to learn prediction error filters for a generalized detection model; however, it shows a decline in performance when numerous post-processing methods were employed in the manipulated regions. Thus, recent studies have worked on developing a generalized model with improved performance as in [7], which introduces an adaptable autoencoder-based neural network architecture to new target domains using a few training samples. Also, [41] employs RGB images as training inputs for the classifiers to distinguish cross-model manipulations by post-processing operations as blurring and JPEG compression.

Frequency-based detection. Some other studies [2, 11,12,14,17,25,31,45] employ the frequency spectrum as training input for the classifier to distinguish between the real and fake images. Previous studies, such as [25], suggest a detection method based on the artifacts in the spatial, frequency domain through the variance of the prediction residue. To enlarge the detection scope, [17] employ FFT and SVD to distinguish copy-move manipulations in images under JPEG compression and Gaussian noise and blurring attacks. Also, [31] suggests a GAN-specific detection method in frequency-domain based on artificial fingerprints on generated images. [2] propose a manipulation localization architecture that utilizes spatial maps and frequency domain correlation to examine the distinct features of forged areas by employing an encoder and LSTM network. Recently, [14] have conducted a comprehensive analysis of the artifacts generated by GANs in the frequency space using Discrete Cosine Transfer (DCT). Also, [45] propose a classifier based on the artifacts induced by the up-sampler of GANs, but additionally exploit a number of categories, including the human face, horse, landscape, satellite image, and painting. [11,12] exploit the spectral distortions via Azimuthal integration for identifying manipulated images.

Our approach also employs the frequency spectrum as the input data but differs from the previous studies in two major aspects: first, we employ the high pass filter at the low-frequency spectrum to be able to solely concentrate on the high-frequency spectrum; second, Laplacian of Gaussian (LoG) is applied to the magnitude for our model to focus on the background of the image to identify the general characteristics of the synthesized images for cross-domain detection performance.

3. Analysis of Frequency-level Artifacts

Before developing a new framework to improve the cross-domain performance, we first analyze the frequency-level artifacts discovered in the synthesized images by GAN models, for the detector to distinguish between the ‘real images’ and ‘fake (synthesized) images.’ Unfortunately, since the previous methods only focus on detecting the artifacts using the labeled dataset, it has been impossible to extract and analyze the information of the frequency-level artifacts. Thus, we develop a novel model to extract the *Artifact Compression Map* (ACM) that shows the properties of the artifacts in the frequency spectrum.

3.1. Artifact Compression Map

We first derive the frequency-level map of the generated images by the summation of the frequency-level contents and the frequency-level artifacts. Thus,

$$|\mathbf{Z}_{fake}| \equiv |\mathcal{F}\{\mathbf{X}_{fake}\}| = |\mathbf{Z}_{content}| + |\mathbf{U}|\,, \qquad (1)$$

where $|\bullet|$ results in the element-wise magnitudes of the values in the input matrix, $\mathbf{X}_{fake}$ indicates the fake image, and $\mathbf{Z}_{content}$ and $\mathbf{U}$ represent the frequency-level content map and the frequency-level artifacts, respectively. Since the real image is obtained without any frequency-level artifacts, we can define the frequency-level real image by $|\mathbf{Z}_{real}| \equiv |\mathcal{F}\{\mathbf{X}_{real}\}| = |\mathbf{Z}_{content}|$ where $\mathbf{X}_{real}$ is the real image.

According to the previous studies [11, 12], the difference in the frequency spectrum between the real and fake images can be discovered in the specific frequency components. We define a trainable ACM by $\mathbf{W}_c \in \mathbb{R}^{w\times h}$ where the values range from zero to one. When the frequency component contains the frequency-level artifacts, the value of ACM should be zero to compress the artifacts, while all the other values are one. Then, the magnitude of $\mathbf{U}$ is removed by the element-wise multiplication of $\mathbf{W}_c$ for $|\mathbf{Z}_{fake}|$ as follows:

$$\mathbf{W}_c \odot |\mathbf{Z}_{fake}| = \mathbf{W}_c \odot |\mathbf{Z}_{content}|\,. \qquad (2)$$

The resulting $\mathbf{W}_c \odot |\mathbf{Z}_{content}|$ is similarly obtained from the real image (i.e. $\mathbf{W}_c \odot |\mathbf{Z}_{real}| = \mathbf{W}_c \odot |\mathbf{Z}_{content}| \approx$). Thus, when $\mathbf{W}_c \odot |\mathbf{Z}_{fake}| \approx \mathbf{W}_c \odot |\mathbf{Z}_{real}|$, $\mathbf{W}_c$ selectively compresses the frequency components of the artifacts, so we can recognize the artifacts by analyzing $\mathbf{W}_c$.

3.2. Extraction of Artifact Compression Map

We extract the ACM by using the fake image classification network with an add-on module adversarially updating the trainable $\mathbf{W}_c$. In contrast to the conventional classification network, the goal of our proposed network is to acquire $\mathbf{W}_c$, rather than the detection of fake images.

As shown in Fig. 2, the overall architecture of our network is similar to the conventional deep neural network, and we use ResNet-50 [15] as the basic architecture. The discriminator predicts the real and fake by using the pixel-level original images, and the add-on module compresses the frequency components of input images by the element-wise multiplication of $\mathbf{W}_c$. The add-on module has a simple process of operation, which contains a Fourier transformer $\mathcal{F}$, an inverse Fourier transformer $\mathcal{F}^{-1}$, and a trainable weight map $\mathbf{W}_c^o$. The resolution of $\mathbf{W}_c^o$ is equivalent to that of the input image ($w \times h$), while the number of its channels is two (i.e. $\mathbf{W}_c^o \in \mathtt{R}^{w\times h\times 2}$). From the two channels of $\mathbf{W}_c^o$, $\mathbf{W}_c$ is estimated by the softmax with the temperature scaling [16] as follows:

$$\mathbf{W}_c(\omega_1, \omega_2) = \frac{e^{\left(T_f \mathbf{W}_c^{o1}(\omega_1,\omega_2)\right)}}{e^{(T_f \mathbf{W}_c^{o1}(\omega_1,\omega_2))} + e^{(T_f \mathbf{W}_c^{o2}(\omega_1,\omega_2))}}, \qquad (3)$$

where T_f is a temperature scaling parameter, $\omega_1 \in \{1, \ldots, w\}$, $\omega_2 \in \{1, \ldots, h\}$, and $\mathbf{W}_c^{o1}$ and $\mathbf{W}_c^{o2}$ represent the first and second channel of $\mathbf{W}_c^o$, respectively. We represent the operation of the add-on module as follows:

$$\begin{aligned}&\widehat{\mathbf{X}}(x, y) = \\ &\quad \mathcal{F}^{-1}\{\mathbf{W}_c(\omega_1, \omega_2) \odot \mathcal{F}\{\mathbf{X}(x, y)\}(\omega_1, \omega_2)\}(x, y),\end{aligned} \qquad (4)$$

where $x \in \{1, \ldots, w\}$, $y \in \{1, \ldots, h\}$, $\mathbf{X} \in \mathtt{R}^{w\times h}$ is the original input data, $\widehat{\mathbf{X}} \in \mathtt{R}^{w\times h}$ is the result from the add-on module, and $\odot$ indicates the element-wise multiplication. Thus, in the add-on module, the compression map scales the frequency components of the input images in the frequency domain. When the input image contains RGB color channels, one $\mathbf{W}_c$ is shared across all channels. $\mathbf{W}_c^{o1}$ and $\mathbf{W}_c^{o2}$ are initialized by w^o and $-w^o$, respectively, where w^o is a user-defined hyperparameter. We set w^o by a large value to let the initial values of $\mathbf{W}_c$ be close to 1. Thus, at the initial phase, the compressed image after the add-on module is almost equivalent to the original image, because $\mathbf{W}_c$ compresses none of the frequency components.

In the training phase, the classification network is trained by the mini-batch gradient descent where every iteration contains two inner updates. At the first update, the classification network is trained simultaneously by the original input images and the compressed images obtained from the add-on module. With the two types of images, every weight parameter except $\mathbf{W}_c^o$ is updated by the classification loss $\mathcal{L}_c$ to classify the real and fake images as:

$$\mathcal{L}_c = \sum_{i=1}^{N_b} CE(g(\mathbf{X}_i), \mathbf{y}_i) + \sum_{i=1}^{N_b} CE(g(\widehat{\mathbf{X}}_i), \mathbf{y}_i), \qquad (5)$$

where CE is the cross-entropy loss, N_b is the size of mini-batch, $g(\mathbf{X})$ means the results of $\mathbf{X}$ predicted by the ResNet-50, $\mathbf{X}_i$ and $\mathbf{y}_i$ are the i-th pair of data and label in the sampled mini-batch, and $\widehat{\mathbf{X}}_i$ is the compressed image

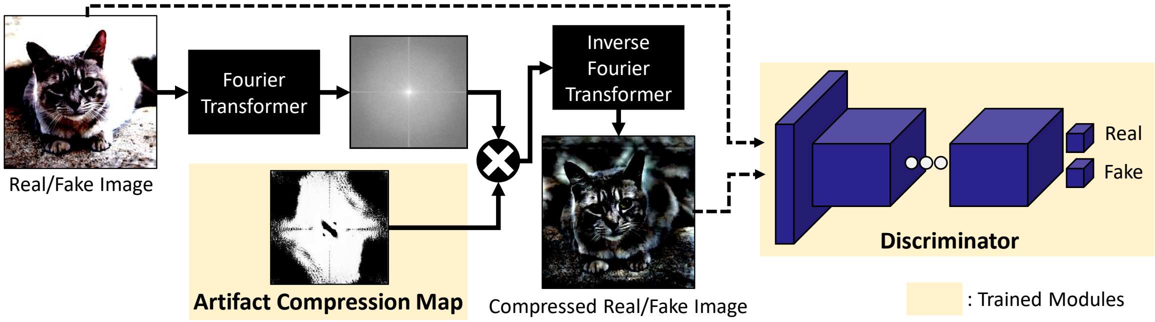


Figure 2: **Network architecture for artifact compression map.** Based on the frequency-level compression map and a conventional classification network, the Artifact Compression Map (ACM) is estimated by an adversarial learning scheme.

Table 1: Comparison of prediction accuracy by *Horse*.

Prediction	*Horse*	*Cat*	*Car*	*Church*	All-categories
$\mathbf{X}$	89.1	**86.4**	**68.5**	**55.3**	**74.9**
$\widehat{\mathbf{X}}$	**96.0**	71.1	54.4	52.9	68.6

from $\mathbf{X}_i$. Since fake image detection is a binary classification task, $\mathbf{y}_i \in \{0, 1\}$ where 0 and 1 represent the real and fake label, respectively. Thus, the first term considers the predictions for the original images, while the second term tries to correctly predict the compressed images.

At the second update, only the parameters of $\mathbf{W}_c^o$ in the add-on module are updated. We update $\mathbf{W}_c^o$ according to the definition of $\mathbf{W}_c$ that confuses the classifier to mistakenly label fake images as real, by compressing the frequency components of artifacts. Thus, we only consider the fake data of the sampled batch in the loss, while the labels are inverted by 0 indicating the real image as follows:

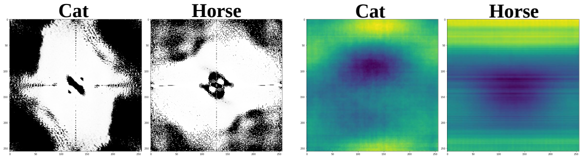
$$\mathcal{L}_{adv} = \sum_{i=1}^{N_f} CE(g(\widehat{\mathbf{X}}_{f(i)}), 0), \qquad (6)$$

where $f(i)$ is the index of pair of i-th fake image in the sampled batch and N_f is the number of fake data in the sampled batch.

3.3. Analysis of Compression Map and Artifacts

For the analysis, we extract the ACM by using the fake datasets obtained by ProGAN [20]. The fake datasets contain multiple categorized fake datasets with different classes of target subjects from each other. To show the effectiveness of the detected artifacts across the subject classes, we train the proposed network by one of the categorized datasets, which is validated by other datasets of different categories. The datasets are explained in detail in Section 5.1.

Using the trained network, we can predict the test images by two schemes: the first scheme is the conventional prediction method using the original images, and the second scheme is the compressed prediction method using the compressed images from the add-on module. According to Eq. 1, the first scheme would consider both the artifacts and the content information, while the second scheme cannot utilize the artifacts by the compression. As shown in Table 1, the first scheme ($\mathbf{X}$) shows better cross-domain performance than the second compressed scheme ($\widehat{\mathbf{X}}$), which verifies that the compressed artifacts can be found across the unseen domains. Thus, it can be confirmed that the compressed artifacts generally appear in fake images, which can be the key factor to improve the cross-domain performance.



(a) The frequency-level compression maps. The dark regions represent the frequency components of the artifacts.

(b) The average maps of pixel-level artifacts. The bright pixels represent the artifacts in the background.

Figure 3: **Analysis of the compression map on cat and horse.** The maps illustrate that the artifacts are found in the high frequency-level components and the pixel-level background regions.

To speculate the properties of the compressed artifacts, we analyze the trained compression map respectively by the frequency-level and pixel-level. Fig. 3(a) shows that the compression maps trained by *cat* and *horse* classes. The dark regions of the compression maps represent the compressed frequency components of the artifacts. In the trained compression map, even though the small regions of the low-frequency components in the center are also compressed, most compressed regions are located at the high-frequency components. From the analysis, we confirm the compressed artifacts are located at the high-frequency components.

To analyze the compression map at the pixel-level, we subtract the compressed image from the original image to obtain the artifact image. Fig. 3(b) shows that the average image of the artifact images obtained from the entire training images. Interestingly, the artifacts mainly appear in the surrounding background regions in bright colors, while the central regions where subjects are commonly located appear dark. It can be concluded that the compressed artifacts generally appear in the background region of the fake images.

4. Bilateral High-Pass Filters

From the trained compression map, we confirm that the cross-domain artifacts mainly appear in the high-frequency components and the background region of the pixel-level images. Based on the discoveries, we propose the Bilateral High-Pass Filters (BiHPF) that emphasize the effect of the artifacts in fake images. BiHPF contains two High-Pass Filters (HPF) including the *pixel-level HPF* and the *frequency-level HPF*. The pixel-level HPF highlights the artifacts appearing near the backgrounds, and the frequency-level HPF emphasizes the high-frequency components.

First, the input image is transformed into the magnitude spectrum of the frequency map through the 2D Fourier transform. If the input image has multiple channels (i.e. 3 RGB channels), we first transform the colored image into grayscale to reduce the class-specific information [32]. Then, we obtain the magnitude spectrum map from the frequency-level map and shift the coordinates to relocate the origin at the center. Here, BiHPF is applied to the magnitude spectrum map, in the consecutive order of pixel-level HPF to frequency-level HPF. After applying the HPFs, we use the filtered magnitude spectrum map as the input of the deep neural network for detection. The backbone architecture of our detection model is based on ResNet-50 [15], which is pre-trained by ImageNet [38]. Although the pre-trained model is optimized to the pixel values, we empirically find that the pre-trained model demonstrates better performance even for the frequency-level inputs compared to a model developed from scratch.

4.1. Pixel-level High-pass Filter

The pixel-level HPF works to compress the central regions of the image in the pixel-domain and to emphasize the effect of the artifacts in the background regions. Although various options are available to compress the central region, we propose a method utilizing the frequency-level Laplacian of Gaussian (LoG) filter. Since the LoG filter is applied to the frequency domain, the artifacts in the frequency-level are effectively focused compared to the method employing a weighting window in the pixel-level map. Also, the LoG filter requires only one tuning hyperparameter as the variance of LoG kernel, which simplifies the tuning sequence. To verify the equivalency of the LoG filter in the frequency magnitude spectrum and the weighting window in the pixel image, we need to utilize several properties of the Fourier transform. For a simple description, we consider the 1D LoG kernel, which can be extended to 2D LoG kernels by independently deriving the two dimensions. Through the properties of differentiation and linearity of the Fourier transform, the LoG kernel with the various σ^2 can be transformed as follows:

$$\begin{aligned}\mathcal{F}\{LoG(x)\}(\omega) &= \mathcal{F}\left\{-\frac{1}{\sigma^2}\left(1-\frac{x^2}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}\right\}(\omega) \\ &= -\sigma\omega^2 e^{-\frac{(\sigma\omega)^2}{2}}.\end{aligned} \tag{7}$$

Then, according to the dual properties of the Fourier transform, we can estimate the inverse Fourier transform of $LoG(\omega)$ as follows:

$$\mathcal{F}^{-1}\{LoG(\omega)\}(x) = -\frac{\sigma x^2}{2\pi}e^{-\frac{(\sigma x)^2}{2}}. \tag{8}$$

From the derivation, the weighting window is represented as $\mathcal{F}^{-1}\{LoG(\omega)\}(x)$, which amplifies the specific regions determined by σ while compressing the other regions. Thus, for a 1D vector of v, the LoG filter in the frequency spectrum map is equivalent to the element-wise multiplication of the weighting window in the pixel domain by $\mathcal{F}^{-1}\left\{\mathcal{F}\{v(x)\}(\omega) * LoG(\omega)\right\}$ derived as follows:

$$v(x) \odot \mathcal{F}^{-1}\left\{LoG(\omega)\right\}(x) = v(x) \odot \left(-\frac{\sigma x^2}{2\pi}e^{-\frac{(\sigma x)^2}{2}}\right). \tag{9}$$

4.2. Frequency-level High-pass Filter

The frequency-level HPF should compress out the low-frequency components in the magnitude spectrum map after applying the pixel-level HPF. We utilize the ideal HPF that equals 0 before the predefined cut-off frequency, and 1 otherwise. Thus, the output $\mathbf{Z}'$ after applying the frequency-level HPF can be derived as:

$$\mathbf{Z}'(\omega_1,\omega_2) = \begin{cases} 0 & \text{if } \omega_1^2+\omega_2^2 \le \omega_c^2 \\ \mathbf{Z}(\omega_1,\omega_2) & \text{otherwise,} \end{cases} \tag{10}$$

where $\omega_1 \in \{-w/2,\dots,w/2\}$, $\omega_2 \in \{-h/2,\dots,h/2\}$, w_c is the predefined cut-off frequency, and $\mathbf{Z}$ is the magnitude spectrum map filtered by the pixel-level HPF. Then, after applying the frequency-level HPF, all the low-frequency components under ω_c are entirely removed, while the high-frequency components over ω_c remain.

5. Experimental Results

To show the cross-domain performance of the proposed framework, we build three types of experiments as fol-

Table 2: Cross-category performance with face dataset

Model	Face		Cross-categories		All-categories	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
Wang (CVPR 2020)	**99.9**	**100.0**	60.9	73.6	68.7	78.9
Frank (ICML 2020)	95.2	96.5	69.1	72.3	74.3	77.1
Durall (CVPR 2020)	86.2	93.4	62.7	53.1	67.4	61.2
Ours	97.0	98.1	**72.7**	**76.1**	**77.6**	**80.5**

Table 3: Cross-category performance with *horse* only

Feat.	Classifier	Test-category		Cross-categories		All-categories	
		Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
Pixel-based	ResNet-50	72.4	68.7	61.9	58.6	64.5	61.1
Wang (CVPR 2020)	ResNet-50	50.5	66.6	50.0	58.3	50.2	60.4
Frank (ICML 2020)	ResNet-50	93.3	89.7	73.5	68.1	78.4	73.5
Durall (CVPR 2020)	SVM (rbf)	88.3	83.0	62.0	59.2	68.5	65.1
Durall (CVPR 2020)	SVM (poly)	88.8	83.9	62.0	59.1	68.7	65.3
Durall (CVPR 2020)	SVM (linear)	81.1	74.1	60.2	57.0	65.4	61.3
Durall (CVPR 2020)	Linear Reg.	79.9	73.2	60.5	57.0	65.3	61.1
Ours	ResNet-50	**94.8**	**93.5**	**73.4**	**69.0**	**78.7**	**75.2**

lows: the *cross-category*, *cross-color*, and *cross-GAN performance* evaluations. First, for *cross-category performance*, we test the model trained with only one category and then with multiple categories. Second, for *cross-color performance*, color manipulation is considered within the same category and tested with different color variations. Lastly, for *cross-GAN performance*, the model is trained with only one GAN model and tested with multiple GAN models.

5.1. Implementation

Hyperparameter. The input size is fixed by 256×256, and Adam optimizer [24] is used to train the classification network with the learning rate of 10^{-4}. The updating phase is iterated until 20 epochs. In BiHPF, the cutoff frequency ω_c is set to 40, and the variance of LoG filters σ is 0.01.

Dataset. We train our model with ProGAN [20] and test with various images and diverse GAN models, as conducted by Wang *et al.* [41]. For real images, we use *LSUN* [44] and *FFHQ* [21]. For fake images, we utilize various generated images of different GAN models, including *ProGAN*, *StyleGAN* [21], *StyleGAN2* [22], *BigGAN* [4], *CycleGAN* [46], and *StarGAN* [6]. To validate the effectiveness of BiHPF for cross-domain performance, we utilize various domains including the subject categories, color manipulations, and GAN models. The generated classes vary by the GAN models, as in Wang *et al.* [41]. Also, we consider five types of color manipulations including variations of hue, brightness, saturation, gamma, and contrast. Since the images have different resolutions, all images are resized into 256×256.

Evaluation metrics. We utilize two evaluation metrics of the Average Precision score (A.P.) and Accuracy (Acc.). The A.P. is obtained by the alternative measurement used by Wang *et al.* [41], which approximates the area under the precision-recall curve by using a few thresholds. The Acc. is the proportion of the correctly predicted test images among the entire test images.

5.2. Cross-domain Performance Evaluations

Cross-category Performance. We perform two experiments to show the cross-category performance. First, we train the detection model with the *face* category of fake images generated by *ProGAN* and evaluate it with the remaining 20 categories. The number of training images is $20,000$ and each category of the test set contains 400 images. The number of real images is always the same as the number of fake images in each dataset. The experimental results are given in Table 2. The performance in the columns of *Face* is obtained only by the test set of *face* category, while the columns of *Cross-categories* and *All-categories* show the average performance of the unseen four categories and all categories in the testset, respectively. Especially for the *cross-categories,* we can confirm that the proposed algorithm shows state-of-the-art performance when the same category is considered in the test phase. Interestingly, although trained only with the *face* category, BiHPF maintains high performance when tested with other categories.

Second, the *horse* category of *ProGAN* is employed as the training data, and the testset contains four categories generated by the same GAN model (ProGAN) as follows: *horse*, *car*, *cat*, and *church*. To further evaluate the model with more datasets compared to Wang *et al.* [41], we obtain four additional categories from ProGAN, each containing 1,000 real and 1,000 fake images. In Table 3, the three columns of *Test-category*, *Cross-categories*, and *All-categories* represent the test results in the corresponding category, the other categories, and all four categories, respectively. The *pixel-based detector* utilizes the original images for the fake image detection, while its training method and architecture are equivalent to the proposed framework. From the results, our approach successfully improves the cross-category performance compared to the previous studies. In particular, our model outperforms others in generalized detection based on the simple operations of BiHPF.

Cross-color Performance. We conduct experiments to validate the robustness of the detection models in color manipulations. If images are partially manipulated with variations in hue, brightness, saturation, gamma, and contrast, the image artifacts contain different characteristics compared to those in the entirely synthesized images. The hue factor is the amount of shift in 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. Impressively, our model is more robust and resistant to diverse manipulations compared to others. Especially, Wang *et al.* [41] show a performance drop, since it uses RGB rather than grayscale.

Table 4: Comparison of cross-color performance.

Model	Original		Hue		Brightness		Saturation		Gamma		Contrast		Mean		Min	
	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 (CVPR 2020)	99.9	100.0	73.9	81.3	61.8	74.7	74.3	84.4	70.2	83.2	66.6	79.7	69.4	80.7	61.8	74.7
Frank (ICML 2020)	95.2	96.5	85.5	97.2	84.2	97.2	91.2	98.0	85.4	97.4	84.3	96.7	86.1	97.3	84.2	**96.7**
Durall (CVPR 2020)	86.2	93.4	86.2	81.9	85.9	81.9	86.2	81.9	85.1	80.8	85.2	81.2	85.7	81.5	85.1	80.8
Our	97.0	98.1	92.0	97.8	92.0	97.9	91.9	96.7	91.7	96.8	92.4	98.1	**92.0**	**97.5**	**91.7**	**96.7**

Table 5: Comparison of cross-model performance.

Model	Training settings		Test Models																	
	Input	# class	ProGAN		StyleGAN		StyleGAN2		BigGAN		CycleGAN		StarGAN		Mean		Min			
			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 (CVPR 2020)	Pixel	1	50.4	63.8	50.4	79.3	68.2	94.7	50.2	61.3	50.0	52.9	50.0	48.2	53.2	66.7	50.0	48.2		
Frank (ICML 2020)	Freq	1	78.9	77.9	69.4	64.8	67.4	64.0	62.3	58.6	67.4	65.4	60.5	59.5	67.7	65.0	60.5	58.6		
Durall (CVPR 2020)	Freq	1	85.1	79.5	59.2	55.2	70.4	63.8	57.0	53.9	66.7	61.4	99.8	99.6	73.0	68.9	57.0	53.9		
Our	Freq	1	82.5	81.4	68.0	62.8	68.8	63.6	67.0	62.5	75.5	74.2	90.1	90.1	**75.3**	**72.4**	**67.0**	**62.5**		
Wang (CVPR 2020)	Pixel	2	64.6	92.7	52.8	82.8	75.7	96.6	51.6	70.5	58.6	81.5	51.2	74.3	59.1	**83.1**	51.2	70.5		
Frank (ICML 2020)	Freq	2	85.7	81.3	73.1	68.5	75.0	70.9	76.9	70.8	86.5	80.8	85.0	77.0	80.4	74.9	**73.1**	68.5		
Durall (CVPR 2020)	Freq	2	79.0	73.9	63.6	58.8	67.3	62.1	69.5	62.9	65.4	60.8	99.4	99.4	74.0	69.7	63.6	58.8		
Our	Freq	2	87.4	87.4	71.6	74.1	77.0	81.1	82.6	80.6	86.0	86.6	93.8	80.8	**83.1**	81.8	71.6	**74.1**		
Wang (CVPR 2020)	Pixel	4	91.4	99.4	63.8	91.4	76.4	97.5	52.9	73.3	72.7	88.6	63.8	90.8	70.2	**90.2**	52.9	73.3		
Frank (ICML 2020)	Freq	4	90.3	85.2	74.5	72.0	73.1	71.4	88.7	86.0	75.5	71.2	99.5	99.5	83.6	80.9	73.1	71.2		
Durall (CVPR 2020)	Freq	4	81.1	74.4	54.4	52.6	66.8	62.0	60.1	56.3	69.0	64.0	98.1	98.1	69.7	66.6	54.4	52.6		
Our	Freq	4	90.7	86.2	76.9	75.1	76.2	74.7	84.9	81.7	81.9	78.9	94.4	94.4	**84.2**	81.8	**76.2**	**74.7**		

Cross-model Performance. In addition to the superior cross-category and cross-color performances, we discover that our model also works generally across various GAN models. To show the property, we use the experimental settings of Wang *et al.* [41], which are designed to show the generality of the fake image detector trained by a specific GAN model to identify other GAN models. We use three types of training settings, which contain 1-class, 2-class, and 4-class settings. As represented in Table 5, our proposed approach shows improved performance on the various GAN models, even though the detection algorithm is trained only with the fake images generated by ProGAN. In contrast to other models, our detector shows a decline in performance for StyleGAN and StyleGAN2. This happens due to the similarity of artifacts from them and ProGAN, which results in the overfitting issue of the previous methods. From the result, we can validate that the proposed approach can be expanded to consider the generality of the artifacts across various GAN models.

Table 6: Component-wise ablation tests.

Feat.	Classifier	$\mathcal{L}$	$\mathcal{F}$	Test-category		Cross-categories	
				Acc.	A.P.	Acc.	A.P.
Durall (CVPR2020)	SVM (rbf)			92.8	88.4	70.2	66.5
Durall (CVPR2020)	SVM (linear)			86.0	78.9	68.0	63.4
Durall (CVPR2020)	Logistic Reg.			83.9	76.7	67.7	63.1
Ours	ResNet-50			98.4	97.9	68.7	67.7
Durall (CVPR2020)	SVM (rbf)	✓		91.1	86.1	70.4	66.4
Durall (CVPR2020)	SVM (linear)	✓		84.3	76.4	68.2	63.5
Durall (CVPR2020)	Logistic Reg.	✓		82.0	74.4	67.6	62.9
Ours	ResNet-50	✓		98.9	**98.6**	73.1	72.5
Durall et al. (CVPR 2020)	SVM (rbf)		✓	92.6	88.2	71.7	67.6
Durall et al. (CVPR 2020)	SVM (linear)		✓	86.3	79.0	70.0	64.9
Durall et al. (CVPR 2020)	Logistic Reg.		✓	84.7	77.5	71.2	65.8
Ours	ResNet-50		✓	99.0	98.2	77.7	75.4
Durall et al. (CVPR 2020)	SVM (rbf)	✓	✓	91.4	86.3	69.6	64.9
Durall et al. (CVPR 2020)	SVM (linear)	✓	✓	84.8	76.9	69.1	64.2
Durall et al. (CVPR 2020)	Logistic Reg.	✓	✓	83.3	75.6	70.1	64.9
Ours	ResNet-50	✓	✓	**98.9**	98.5	**79.8**	**77.7**

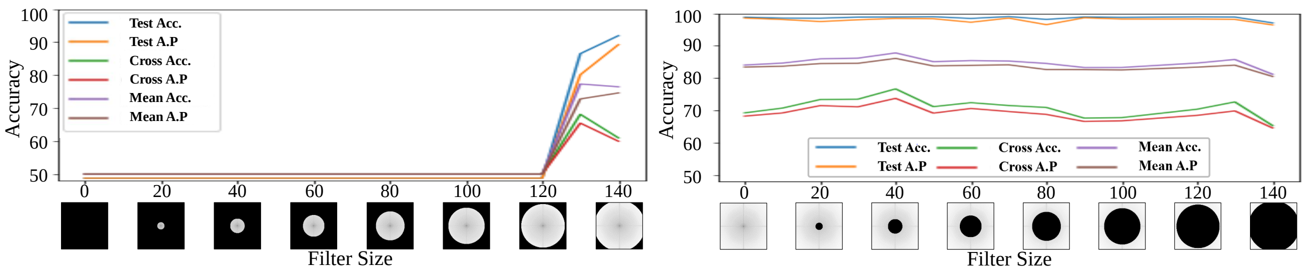
5.3. Ablation Study

We validate the performance of each component and hyperparameter of BiHPF through experiments for ablation study. In this section, we employ the *horse* dataset of *StyleGAN2* for training to show the model's stable performance even when trained with a different dataset other than *ProGAN*. Also, we conduct cross-category experiments to further validate the model's performance.

5.3.1 Validity of BiHPF.

To show that the effect of BiHPF is not limited to our framework, we employ BiHPF to other algorithms for fake image detection [11]. In this experiment, we also compare the effect of the pixel-level HPF ($\mathcal{L}$) and the frequency-level HPF ($\mathcal{F}$) by removing either one or both of them from our mechanism. Table 6 shows the results that our mechanism can improve the cross-domain performance compared to the other algorithms for fake image detection. Thus, by using the proposed mechanism as pre-processing, every single one of the previous studies can preserve its performance when the given subject is out of the training settings. In addition, when utilizing both of the HPFs in BiHPF, our framework shows superior cross-domain performance than the framework where one or both of the HPFs are missing. The frequency-level HPF shows greater performance improvement than the pixel-level HPF, signifying the importance of dismissing the low-frequency components to emphasize the effect of the artifacts.

More specifically, the artifacts in the foreground or cen-



(a) Low-pass filter

(b) High-pass filter

Figure 4: Ablation test with various w_c

Table 7: Ablation test with various σ.

σ	Test-category		Cross-categories		All-categories	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
0.0001	**99.0**	98.4	79.3	76.8	84.2	82.2
0.001	98.2	97.0	78.0	75.7	83.1	81.0
0.01	98.9	**98.5**	**79.8**	**77.7**	**84.6**	**82.9**
0.1	98.8	97.8	76.5	74.0	82.1	79.9
0.5	98.9	98.4	74.9	73.7	80.9	79.9

ter of the image contain domain-specific features, whereas the artifacts in the background or edge of the image contain general information. Thus, we can confirm that employing the LoG filter is effective at improving the cross-domain performance. Since Durall *et al.* [11] does not use the spatial information, it shows poor performance in the cross-domain test.

5.3.2 Hyperparameters of BiHPF.

The frequency-level and pixel-level HPFs of BiHPF are respectively controlled by only one hyperparameter for each: the cut-off frequency (w_c) and the variance of LoG (σ^2). First, to show the sensitivity of the performance with σ values, we perform additional experiments by changing the value of σ as represented in Table 7. The performance is evaluated by the validation set different from the test set, and the best performance overall is when $\sigma = 0.01$. Interestingly, while the test domain performance fluctuates with the various values of σ, the cross-domain performance consistently declines as σ moves far from the peak of $\sigma = 0.01$. Thus, we can reconfirm that the value of σ influences the effect of the artifacts in fake images.

Second, we validate the necessity of the frequency-level HPF by replacing it with the frequency-level low-pass filter. As shown in Fig. 4 (a), the performance declines dramatically when the high-frequency components are removed by the frequency-level low-pass filter, which validates the importance of the high-frequency components for fake image detection. In contrast, as shown in Fig. 4 (b), the performance maintains even with the various values of w_c. Thus, the performance is robust to w_c, which indicates that the fake images can be distinguished even with limited high-frequency components in the Fourier domain.

6. Conclusion

We present a novel mechanism called BiHPF for robust detection of synthesized images across various categories, color manipulations, and GAN models. Since the previous state-of-the-art models heavily depend on the training settings, their performance can decline when tested with unseen data during the training phase. In contrast to those models, our model achieves robust generalization with state-of-the-art performance, when tested even with unseen data outside of the training settings. Our new model has strong practical implications in three major aspects: first, our model achieves superior performance in both facial and non-facial categories, unlike most other models with limited detection performance in the face category only; second, our new model shows the outstanding cross-domain performance when tested with various categories; finally, we develop a simple but robust model adversarially extracting the frequency-level artifacts, which are successfully utilized for the detection of synthesized images of GAN models. We expect our new framework to be utilized in the prevention and detection of malicious abuse of synthesized images to protect our society. [1]

Acknowledgements

This work was supported by Samsung SDS and the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the National Innovation convergence complex R&D program (P0015336_Autonomous Driving for Last Mile Service near Central Government Complex).

[1] https://github.com/SamsungSDS-Team9/BiHPF

References

[1] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR Workshops*, 2019.

[2] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *TIP*, 2019.

[3] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *ACM Workshop on Information Hiding and Multimedia Security*, 2016.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Tiago Carvalho, Hany Farid, and Eric R Kee. Exposing photo manipulation from user-guided 3d lighting analysis. In *MWSF*, 2015.

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[7] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv*, 2018.

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

[9] A. E. Dirik and N. Memon. Image tamper detection based on demosaicing artifacts. In *ICIP*, 2009.

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

[11] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions. In *CVPR*, 2020.

[12] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features. *arXiv*, 2019.

[13] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Trans. Inf. Forensics Secur*, 2012.

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

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015.

[17] Deng-Yuan Huang, Ching-Ning Huang, Wu-Chih Hu, and Chih-Hung Chou. Robustness of copy-move forgery detection under high jpeg compression artifacts. *Multimedia Tools and Applications*, 2017.

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

[19] Yonghyun Jeong, Jongwon Choi, Doyeon Kim, Sehyeon Park, Minki Hong, Changhyun Park, Seungjai Min, and Youngjune Gwon. Dofnet: Depth of field difference learning for detecting image forgery. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *CVPR*, 2020.

[23] Eric Kee and Hany Farid. Exposing digital forgeries from 3-d lighting environments. In *WIFS*, 2010.

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

[25] Matthias Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *MMSec*, 2008.

[26] Patrick Kwon, Jaeseong You, Gyuhyeon Nam, Sungwoo Park, and Gyeongsu Chae. Kodf: A large-scale korean deepfake detection dataset. *arXiv preprint arXiv:2103.10094*, 2021.

[27] Sangyup Lee, Shahroz Tariq, Youjin Shin, and Simon S Woo. Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. *Applied Soft Computing*, 105:107256, 2021.

[28] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020.

[29] Y. Li, M. Chang, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *WIFS*, 2018.

[30] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *CVPR*, 2020.

[31] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *CMIPR*, 2019.

[32] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IWACVW*, 2019.

[33] Daniel Mas Montserrat, Hanxiang Hao, SK Yarlagadda, Sriram Baireddy, Ruiting Shao, János Horváth, Emily Bartusiak, Justin Yang, David Güera, Fengqing Zhu, et al. Deepfakes detection with automatic face weighting. *arXiv*, 2020.

[34] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection. *arXiv*, 2019.

[35] Bo Peng, Wei Wang, Jing Dong, and Tieniu Tan. Improved 3d lighting environment estimation for image forgery detection. In *WIFS*, 2015.

[36] Bo Peng, Wei Wang, Jing Dong, and Tieniu Tan. Optimized 3d lighting environment estimation for image forgery detection. *IEEE Trans. Inf. Forensics Secur*, 2016.

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

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[39] Ke Sun, Hong Liu, Qixiang Ye, Jianzhuang Liu, Yue Gao, Ling Shao, and Rongrong Ji. Domain general face forgery detection by learning to weight. 2021.

[40] Dijana Tralic, Juraj Petrovic, and Sonja Grgic. Jpeg image tampering detection using blocking artifacts. In *IWSSIP*, 2012.

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

[42] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, 2019.

[43] Shuiming Ye, Qibin Sun, and Ee-Chien Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *ICME*, 2007.

[44] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv*, 2015.

[45] X. Zhang, S. Karaman, and S. Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, 2019.

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.