

Reducing the Content Bias for AI-generated Image Detection

Seoyeon Gye* Junwon Ko* Hyounguk Shon* Minchan Kwon Junmo Kim

School of Electrical Engineering, KAIST, South Korea

{sawyun, kojunewon, hyounguk.shon, kmc0207, junmo.kim}@kaist.ac.kr

Abstract

Identifying AI-generated content is critical for the safe and ethical use of generative AI. Recent research has focused on developing detectors that generalize to unknown generators, with popular methods relying either on high-level features or low-level fingerprints. However, these methods have clear limitations: biased towards unseen content, or vulnerable to common image degradations, such as JPEG compression. To address these issues, we propose a novel approach, SFLD, which incorporates PatchShuffle to integrate high-level semantic and low-level textural information. SFLD applies PatchShuffle at multiple levels, improving robustness and generalization across various generative models. Additionally, current benchmarks face challenges such as low image quality, insufficient content preservation, and limited class diversity. In response, we introduce TwinSynths, a new benchmark generation methodology that constructs visually near-identical pairs of real and synthetic images to ensure high quality and content preservation. Our extensive experiments and analysis show that SFLD outperforms existing methods on detecting a wide variety of fake images sourced from GANs, diffusion models, and TwinSynths, demonstrating the state-of-the-art performance and generalization capabilities to novel generative models.

1. Introduction

The rapid advancement of AI image generation technologies has brought significant achievements but also growing social concern, as these technologies are increasingly misused for the creation of fake news, malicious defamation, and other forms of digital deception. In response, AI-generated image detection is receiving more attention. There is a wide variety of generative models, along with commercial models with unknown internal architectures. This highlights the need for a generalized detector

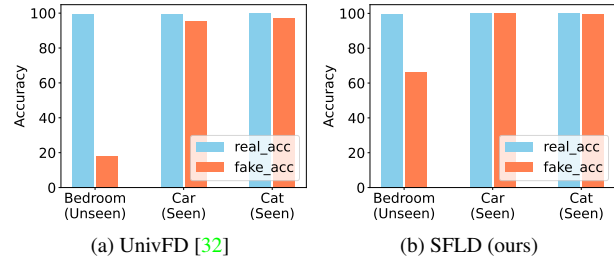


Figure 1. Class-wise detection accuracy for StyleGAN-1 {bedroom, car, cat} class categories. The bedroom class does not appear at training, while car and cat does. UnivFD [32] catastrophically fails to identify synthetic bedroom images, which hints at model bias towards high-level image content.

capable of distinguishing between real and fake images, regardless of the generative model structure.

In this context, early research focused on identifying the characteristic fingerprints of generated images. Recent work, NPR [46] shows that pixel-level features, induced by the upsampling layers commonly found in current generative models, can serve as cues for detection. However, there are clear practical limitations to relying on low-level fingerprints. First, the approach is vulnerable to simple image degradations, such as JPEG compression or blurring, which are common in real-world online environments [49]. Additionally, the model may become biased toward the specific *fakeness* seen at training in cases where generalization to novel generators is not sufficiently considered [32, 56]. For instance, a detector trained on GAN-generated images may learn the characteristics of GANs as the fake features, while mistakenly perceiving images generated by diffusion models as real. This bias limits the detector’s generalizability across different types of generative models.

To tackle these limitations, UnivFD [32] utilizes a robust, pre-trained image encoder. This image embedding is task-agnostic, enabling it to capture high-level semantic information from images. However, we found that UnivFD exhibits a bias towards the observed content in the training images, learning another specific *fakeness*. Fig. 1 shows that Uni-

*Equal contribution.

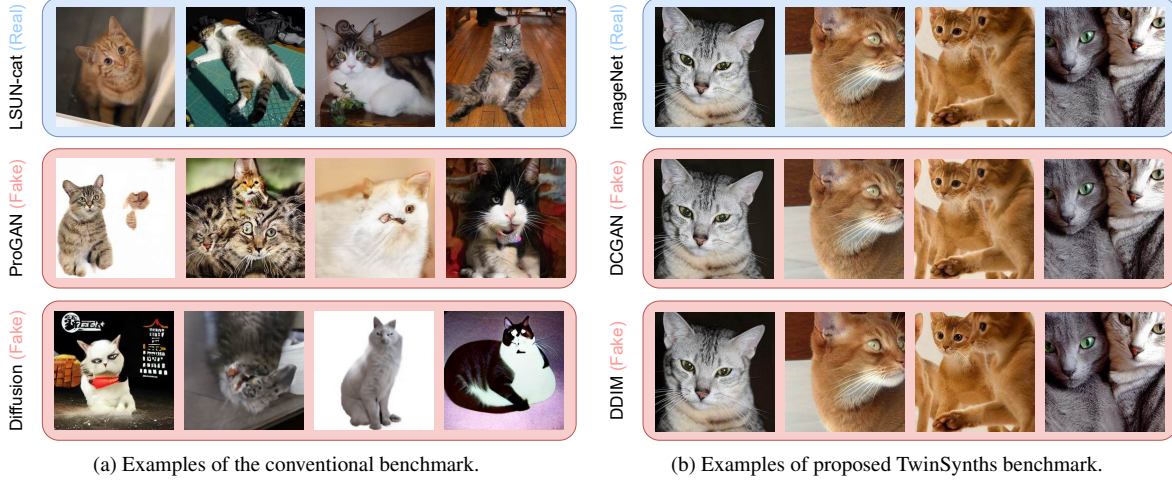


Figure 2. Comparison of benchmarks. (a) Real images and fake GAN images are sampled from the test ProGAN set in the ForenSynths [49]. Fake diffusion images are sampled from benchmark of Ojha *et al.* [32], each from LDM, GLIDE and DALL-E dataset. (b) Real images are sampled from ImageNet dataset, and corresponding fake images are generated by each model.

vFD misclassifies most GAN-generated images of a novel class (StyleGAN-*bedroom*) as *real*. The *bedroom* class is absent from the training set, which may lead the detector to mistakenly classify most images as real, demonstrating the detector’s reliance on seen content during training.

We propose a novel technique called **PatchShuffle**, which is the core of our fake image detection model, **SFLD** (pronounced “shuffled”). PatchShuffle divides the image into non-overlapping patches and randomly shuffles them. This procedure disrupts the high-level semantic structure of the image while preserving low-level textural information. This allows the detection model to better focus on both context and texture. SFLD utilizes an ensemble of classifiers at multiple levels of PatchShuffle, leveraging hierarchical information across various patch sizes. This approach ensures that the model leverages both the semantic and textural aspects of the image to improve fake image detection. The results demonstrate that SFLD achieves superior performance with enhanced robustness and better generalization.

Furthermore, we observe that previous benchmarks have three limitations: **(1) low image quality**. The previous benchmarks contain a significant portion of low-quality images that lag behind the capabilities of current generative models. As a result, the practical usefulness of these benchmarks is significantly reduced. **(2) lack of content preservation**. Some subsets—particularly foundation generative models—lack access to the training data used for the checkpoints. Consequently, the content of the generated and real images often differs significantly, making it difficult to determine whether a detector focuses on real/fake discriminative features or other irrelevant features. **(3) limited class diversity**. Existing benchmarks primarily focus on expanding the variety of generative models without considering the

generated class diversity and scalability among generative models. As shown in Fig. 1, this makes it difficult to identify detection bias towards certain classes, as well as hard to represent the in-the-wild performance of the detector due to limited class diversity.

To address these challenges, we propose a new benchmark generation methodology and corresponding benchmark, **TwinSynths**. It consists of synthetic images that are visually near-identical to paired real images for practical and fair evaluations. TwinSynths constructs image pairs that preserve both quality and content while retaining the architectural characteristics of each generative model. Also, TwinSynths enables flexible class expansion by generating synthetic images tailored to the real image. Using this benchmark, we evaluate the performance of our proposed SFLD method as well as existing detection models.

Our main contributions are summarized as follows:

- We propose SFLD, a novel AI-generated image detection method that integrates semantic and texture artifacts on generated images, achieving state-of-the-art performance.
- We propose a new approach on benchmarks and the subset of generated images that can ensure the quality and content of generated images.
- We validate our method through extensive experiments and analysis that support our hypothesis.

2. Method

2.1. Patch Shuffling Fake Detection

Backbone. We utilize the visual encoder of CLIP ViT-L/14 [13, 36] to leverage the pre-trained feature space. This choice is based on Ojha *et al.* [32], which showed

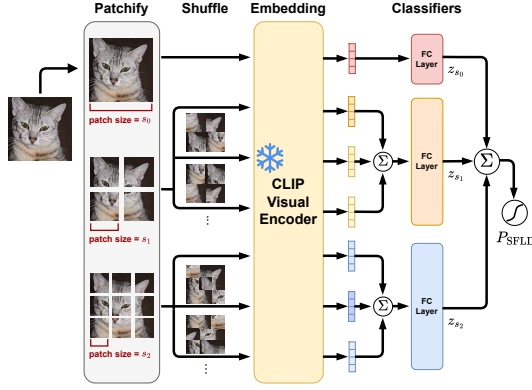


Figure 3. Architecture of the proposed fake image detector (SFLD). z_{s_i} refers to the logit score generated from an input image processed via $s_i \times s_i$ patch size. Σ indicates weighted sum.

that it outperforms other models such as CLIP:ResNet-50, ImageNet:ResNet-50, and ImageNet:ViT-B/16 in distinguishing real from fake images. The results indicated that both the architecture and the pre-training data are crucial. Based on this insight, we chose the ViT model for our backbone. As shown in Fig. 3, we extract CLIP features and train a fully connected layer to classify real and fake images.

PatchShuffle. To effectively integrate both semantic and textural features, PatchShuffle disrupts the global structure of an image while preserving local features. In the PatchShuffle process, the input images are divided into non-overlapping patches of size $s \times s$ and then randomly shuffled. This operation produces a new shuffled image x_s .

For a given s , the logit score of the shuffled image is,

$$z_s = \psi(f(x_s)), \quad (1)$$

where $f(\cdot)$ represents a pre-trained CLIP encoder and $\psi(\cdot)$ is a single fully connected layer appended to f .

There are classifiers for each patch size of shuffled images to leverage local structure information hierarchically within the image. We selected patch sizes of 28, 56, and 224 for the proposed SFLD. As shown in Fig. 3, s_0 is 224, s_1 is 56 and s_2 is 28. These configurations are studied in detail in Sec. 4.5. For each patch size s_j , the classifier ψ_{s_j} is trained independently. Notably, UnivFD takes a center-cropped 224×224 image as input to the CLIP encoder. Therefore, when using a patch size of 224 in PatchShuffle, it effectively corresponds to the same setting as UnivFD [32].

We employ binary cross-entropy loss for each classifier:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \sigma(z_{s_j}) + (1 - y_i) \log (1 - \sigma(z_{s_j}))] \quad (2)$$

where N is the number of data and $y_i \in \{0, 1\}$ is the label whether an input x_i is real ($y_i = 0$) or fake ($y_i = 1$).

SFLD. SFLD combines multiple classifiers trained on shuffled images with different patch sizes. By varying the patch size, SFLD incorporates models that focus on various levels of structural features, ranging from fine-grained local details to more global patterns.

During testing, $N_{views} = 10$ shuffled views are generated for each patch size. The logits from these views are averaged and processed by the corresponding classifier. The final probability $P_{SFLD}(y|x)$ is computed by averaging the logits across patch sizes and applying the sigmoid function:

$$P_{SFLD}(y|x) = \sigma \left(\frac{1}{k} \sum_{j=1}^k \psi_{s_j} \left(\frac{1}{N_{views}} \sum_{i=1}^{N_{views}} f(x_{s_j}^i) \right) \right), \quad (3)$$

where k is the number of patch sizes used in the ensemble (e.g., $k = 3$ in our configuration).

Binary classification is done using a threshold of 0.5 on P_{SFLD} . Although the fusion method is simple and not tuned for each test class, its simplicity enables strong generalization across diverse fake image sources. By combining classifiers trained on different patch sizes, SFLD achieves a robust and general detection performance. Algorithm 1 shows the full workflow of SFLD, especially the fusion of multiple classifiers during inference.

2.2. TwinSynths

In Sec. 1, we pointed out three shortcomings in the previous benchmarks: low image quality, lack of content preservation, and limited class diversity. This issue must be addressed to allow a comprehensive comparison of detectors. Therefore, we propose a novel dataset creation methodology and *TwinSynths* benchmark, consisting of GAN- and diffusion-based generated images that are paired with visually-identical real counterparts. To create a practical benchmark for evaluating generated image detectors, it is essential to ensure the generation of high-quality images that preserve the original content. To achieve this, the image generation process should ideally sample a distribution that closely resembles a real distribution. From this perspective, the image generation or sampling process can be interpreted as effectively fitting the generator to a single real image. Through this approach, we construct image pairs that preserve the content of the images while reflecting the architectural traits of the generative models. Additionally, this methodology allows for the expansion of target classes in the benchmark by generating paired images for any real image. Fig. 2b are some examples of *TwinSynths*. We can see that the content of the paired real image is faithfully reproduced and the quality of the generated image is guaranteed.

TwinSynths-GAN benchmark. The GAN-based subsets in the previous benchmark have disparate training configurations, especially the class of training images, resulting in a discrepancy between the generated and the real images.

In order to generate a high quality image that preserves the content of the paired real image while leveraging the training methodology of GANs, we trained the generator from scratch using a single real image. The MSE loss was provided to the generator to generate an image that is identical to the original image. For reproduction, the latent vector for the generator input is maintained at a fixed value. We created 8,000 generated images from 80 selected ImageNet [41] classes, which is much larger than previous benchmarks. We selected 40 classes following the *ProGAN* subset in ForenSynths [49], while the other 40 classes were chosen arbitrarily. We utilized DCGAN [35] architecture.

TwinSynths-DM benchmark. In comparison to GAN-based subsets, diffusion-based subsets in conventional benchmarks were generated with off-the-shelf pretrained models, having much severer content discrepancy between real and generated images. In order to generate a high quality image that preserves the content of paired real image while leveraging the inference process of diffusion models, we used DDIM inversion [44] to generate image that is similar to the real image. We apply a DDIM forward process to the real image to make it noisy and perform text-conditioned DDIM denoising process using the prompt template ‘a photo of {class name}’. For the prompts, we used the class names from ImageNet. This process makes TwinSynths-DM preserve the similarity with the paired real images. We used the same image classes used to create TwinSynths-GAN. We utilized the pretrained decoder and scheduler of [44].

3. Experiments

3.1. Settings

Datasets. Following the conventions of AI-generated image detection, all detectors were trained using the ForenSynths train set [49]. This train set consists of real images used to train ProGAN [18] and ProGAN-generated images. We evaluate the performance of SFLD on several benchmarks, including conventional benchmarks, TwinSynths, and low-level vision/perceptual loss benchmarks. For more detailed descriptions of the datasets and configurations used, please refer to Appendix B.

Baseline methods. We compare the performance of the proposed SFLD with existing AI-generated image detection methods. It includes CNNSpot [49], FreDect [14], GramNet [25], Fusing [17], LNP [23], LGrad [45], UnivFD [32], and NPR [46]. We conducted evaluations on the detection methods with our test dataset. The evaluation is done by the official models [32, 49], re-implemented models [14, 17, 23, 25, 45] by Zhong *et al.* [54], or trained model with the official codes using 20-classes train set [46].

Evaluation metrics. We assess the performances of the detection models by average precision score (AP) and

classification accuracy (Acc.), following previous works [32, 46, 49]. The AP metric is not dependent on the threshold value, whereas the Acc. is calculated with a fixed threshold of 0.5 across all generation models.

3.2. Results on Conventional Benchmark

Tabs. 1 and 2 shows the detection performance on conventional benchmarks in AP and Acc. All baselines are trained on only the ProGAN train dataset consisting of 20 classes. Higher performance is colored darker. SFLD demonstrates robust and generalized performance across various generators in the benchmark. Note that SFLD achieves above 90.0 AP on every unseen generator. SFLD has an average of 98.43 AP, outperforming the best-performing baseline, UnivFD, by up to 2.14 in average. While for some tasks NPR has shown outperforming AP values in some generators, it has shown relatively low performance on some generators. In this regard, we found that NPR is sensitive to some image degradation or different post-processing methods in different generative models, which limits its practicality. Refer to Sec. 4.3 for further comparison of robustness on image degradation.

SFLD also exhibits state-of-the-art performance in classification accuracy. It performs particularly well on challenging datasets like DeepFake and ADM. On DeepFake, it improves accuracy from 74.6% to 84.2% (+9.6), and on ADM, from 79.5% to 86.0% (+6.5). These gains highlight its superior generalization in difficult scenarios.

3.3. Analysis on TwinSynths

Tab. 3 illustrates the detection performance on TwinSynths in AP. The results demonstrate that SFLD is effective in TwinSynths while some detectors have shown a significant drop in performance. Note that the TwinSynths focused on three key aspects: image quality, content preservation, and class diversity. This suggests that the high performance on conventional benchmarks may not guarantee the detector’s performance in real-world scenarios.

The results of TwinSynths allow an indirect analysis of the factors that the detectors focus on. For convenience, we now define high-level features and low-level features. high-level features are semantic information and their artifacts originate from distribution disparity between real images and generated images. low-level features are texture information and their artifacts stem from the generator traces and image quality of generated images. The TwinSynths-GAN preserves the content of the real image with minimal alteration, as the images are generated from a single real image. This results in UnivFD, which captures high-level feature artifacts on the entire image, resulting in poor performance on the TwinSynths-GAN subset. In contrast, NPR, which captures high-frequency artifacts in neighboring pixels, demonstrates better performance than UnivFD on the

Method	Pro GAN	Style GAN	Style GAN2	Big GAN	Cycle GAN	Star GAN	Gau GAN	Deep fake	DALL E	Glide 100_10	Glide 100_27	Glide 50_27	ADM	LDM 100	LDM 200	LDM 200_cfg	Avg.
CNNSpot [49]	100	99.8	99.5	86.0	94.9	99.0	90.8	84.5	72.9	82.5	80.1	84.7	78.3	71.5	70.3	73.6	85.53
FreDect [14]	100	96.3	72.7	93.9	88.8	99.4	84.5	71.9	95.0	52.2	53.9	55.0	57.3	93.1	92.7	90.4	81.07
GramNet [25]	100	88.2	100	62.7	74.2	100	55.0	93.5	98.8	99.7	99.3	99.1	79.8	99.8	99.8	99.8	90.61
Fusing [17]	100	97.5	100	89.1	95.5	99.8	87.7	69.3	77.1	83.6	81.3	86.2	82.6	75.5	76.2	77.9	86.20
LNP [23]	100	92.5	100	90.2	93.9	100	77.9	73.7	94.9	92.1	88.5	89.5	85.5	93.9	93.6	93.7	91.24
LGrad [45]	100	84.2	99.9	87.9	94.4	100	91.7	64.3	95.6	97.1	94.8	96.3	74.9	96.3	96.2	96.5	91.88
UnivFD [32]	100	97.2	98.0	99.3	99.8	99.4	100	81.8	97.7	95.5	95.8	96.0	88.3	99.4	99.4	93.2	96.29
NPR [46]	100	99.4	99.9	87.4	90.0	100	76.7	82.7	99.2	100	99.8	99.9	84.2	100	99.9	99.9	94.94
SFLD (224+28)	100	99.8	99.9	99.9	100	100	100	91.5	99.1	96.7	97.0	97.5	94.5	99.3	99.3	94.2	98.03
SFLD (224+56)	100	99.8	99.9	99.8	100	100	100	90.9	99.2	98.2	98.4	98.7	94.4	99.6	99.6	95.8	98.39
SFLD	100	99.9	99.9	99.9	100	100	100	93.3	99.3	97.6	97.9	98.4	95.4	99.3	99.3	95.0	98.43

Table 1. Generalization performance on the conventional benchmark reported in AP. SFLD (224+28) indicates the ensemble of the classifier with patch sizes 224 and 28. And SFLD indicates the ensemble of the three classifiers with patch sizes 224, 56, and 28.

Method	Pro GAN	Style GAN	Style GAN2	Big GAN	Cycle GAN	Star GAN	Gau GAN	Deep fake	DALL E	Glide 100_10	Glide 100_27	Glide 50_27	ADM	LDM 100	LDM 200	LDM 200_cfg	Avg.
CNNSpot [49]	100	90.2	86.9	71.2	87.6	94.6	81.4	50.7	57.7	62.4	61.3	64.4	62.5	54.9	54.8	56.0	71.02
FreDect [14]	99.4	80.3	56.1	82.7	81.6	94.5	81.0	62.5	81.6	49.7	52.2	53.4	57.8	79.3	79.0	76.7	72.97
GramNet [25]	100	50.8	100	67.9	72.8	100	57.4	62.0	87.8	95.6	93.4	91.8	79.5	98.7	98.5	98.5	84.65
Fusing [17]	100	71.0	97.1	76.7	85.7	97.2	76.1	53.0	56.1	60.9	59.7	61.6	62.4	53.8	54.5	56.0	70.10
LNP [23]	99.8	78.1	99.6	81.1	82.1	99.9	71.7	56.1	83.5	80.3	76.7	78.0	67.2	80.6	79.6	81.7	80.98
LGrad [45]	99.7	71.4	96.0	80.3	86.6	98.4	80.3	51.9	86.0	90.4	87.1	90.0	68.1	87.9	87.4	87.8	84.30
UnivFD [32]	100	84.4	75.7	95.2	98.7	95.9	99.7	67.7	87.5	78.1	78.7	79.2	70.0	95.2	94.6	74.2	85.89
NPR [46]	100	95.4	96.9	82.9	90.0	99.9	79.8	74.6	83.0	97.9	96.6	97.1	74.3	98.0	97.9	97.7	91.38
SFLD (224+28)	100	95.8	89.0	97.2	99.1	99.3	97.8	80.1	94.6	87.0	87.1	88.9	83.9	95.6	95.5	80.8	91.94
SFLD (224+56)	100	90.6	86.5	97.8	99.5	99.0	98.9	82.7	94.0	89.2	89.2	90.9	81.0	97.0	96.6	80.1	92.05
SFLD	100	96.7	91.9	96.5	99.2	99.4	96.0	84.2	95.2	90.6	90.7	92.5	86.0	95.6	95.7	82.9	93.30

Table 2. Generalization performance on the conventional benchmark reported in accuracy.

Method	Twin-GAN	Twin-DM	Avg.
CNNSpot	62.92	46.93	54.93
FreDect	54.57	55.64	55.11
GramNet	71.98	36.10	54.04
Fusing	61.80	48.62	55.21
LGrad	59.51	34.25	46.88
UnivFD [32]	58.09	74.38	66.24
NPR [46]	78.19	35.76	56.98
PatchShuffle (28)	73.56	65.52	69.54
PatchShuffle (56)	75.90	60.73	68.32
SFLD (224+28)	70.43	75.80	73.12
SFLD (224+56)	70.16	72.44	71.30
SFLD	73.82	72.05	72.94

Table 3. Performance comparisons on TwinSynths. Values indicate AP score. *DM* refers to diffusion model.

TwinSynths-GAN subset. On the other hand, the generated images in TwinSynths-DM contain low-level discriminative features introduced by the DDIM decoder, which incorporates additional fully connected layers and post-processing steps following the upsampling blocks. We can see that NPR exhibits lower performance, whereas UnivFD demonstrates higher performance. Nevertheless, SFLD demonstrates superior and robust performance on both bench-

Tasks	SITD	SAN	CRN	IMLE
UnivFD [32]	65.9	81.2	96.4	98.4
NPR [46]	55.2	60.0	50.0	50.0
SFLD	64.8	86.4	96.6	98.7

Table 4. Low-level vision and perceptual benchmarks. Values indicate AP scores.

marks, indicating its ability to capture both low-level feature artifacts and high-level feature artifacts. Notably, no existing detector has ever exhibited such a high level of performance on both benchmarks.

3.4. Low-level Vision and Perceptual Benchmark

Tab. 4 shows the detection performance on different benchmarks from ForenSynths [49]. Low-level vision models, including SITD and SAN, preserve high-level features of real images. Perceptual models (CRN and IMLE) color semantically segmented images to match real images, preserving semantic information. Notably, while NPR was able to detect some super-resolution images from SAN, it failed to perform well in other image-to-image translation tasks. This indicates that detectors specialized in identifying low-

Tasks	UnivFD [32]		SFLD (ours)	
	Real	Fake	Real	Fake
ProGAN	99.9	100	100	100
StyleGAN	99.4	69.4	99.4	93.9
StyleGAN2	99.8	51.5	100	83.8
BigGAN	98.1	92.2	93.2	99.8
CycleGAN	98.9	98.4	98.3	100
StarGAN	93.6	98.1	98.9	99.9
GauGAN	99.3	100	92.0	100
Deepfake	94.8	40.6	85.2	83.2
DALLE	99.1	75.8	96.2	94.1
ADM	97.2	42.8	95.3	76.6
Glide_100_10	99.1	57.0	96.2	85.0
Glide_100_27	99.1	58.2	96.2	85.2
Glide_50_27	99.1	59.3	96.2	88.7
LDM_100	99.1	91.2	96.2	94.9
LDM_200	99.1	90.0	96.2	95.2
LDM_200_cfg	99.1	49.2	96.2	69.6
Avg.	98.4	73.4	96.0	90.6

Table 5. Classification accuracy on real and fake sets on Foren-Synths [49] and diffusion sets in Ojha *et al.* [32].

level feature artifacts from ProGAN struggle to generalize to images generated from different vision tasks. Conversely, a detector that focuses on high-level feature artifacts demonstrates strong performance on these benchmarks. SFLD integrates semantic and structural information from different patch sizes to show superior performance on low-level vision and perceptual benchmarks.

4. Discussion

4.1. Detailed Comparison with UnivFD

This section presents a comprehensive comparison of SFLD against UnivFD. Tab. 5 shows the classification accuracy of the prediction results of real and fake images on each generator in the conventional benchmark. It is evident that SFLD exhibits superior performance in predicting generated images. Notably, UnivFD is unable to predict fake images in some generated subsets, whereas SFLD demonstrates its strength in both real and generated images. This result supports that SFLD can capture both low-level feature artifacts and high-level feature artifacts, making the detector better generalize on novel generators.

4.2. Score Ensembling

Scatter plots. Ensembling of the detection scores of the original image and patch-shuffled images is supported by Fig. 4. In all cases, ensembling the two detectors with patch sizes 224 and 28 as an average of the two logit scores consistently improved binary separation and thus resulted in superior performance with the default threshold (as evidenced by Tabs. 1 and 2). This proves that the two detection methods work as complementary functions.

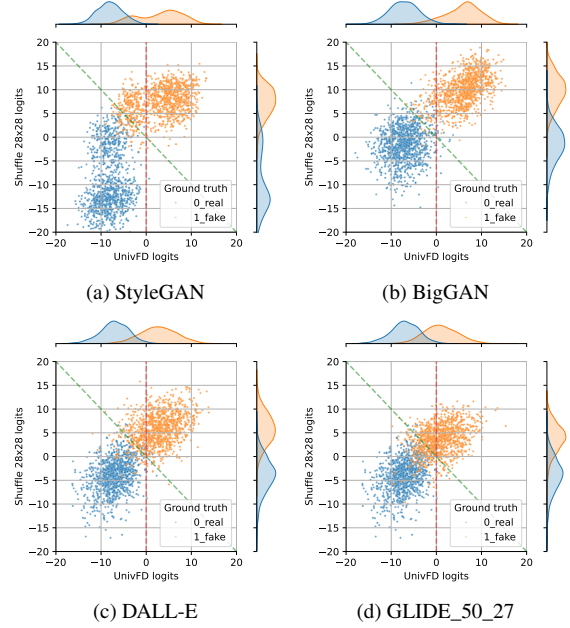


Figure 4. Scatter plots of per-sample scores. X-axis is the UnivFD logits, and Y-axis is the logit from PatchShuffle with patch size 28. The decision boundary of UnivFD (red) and SFLD (green) are shown. See Appendix A for extended results.

A closer look into failure cases. Fig. 5 visualizes some exact failure cases with StyleGAN-generated images (Fig. 4a). Fig. 5a shows a case where UnivFD fails and PatchShuffle succeeds. These images seem to cause UnivFD to fail because the high-level feature is well generated (high global structure fidelity). In contrast, PatchShuffle, which focuses on local structure, succeeds in detection. Our method with score ensembling was able to capture these examples illustrated as the green line in Fig. 4. On the other hand, Fig. 5b shows a case where PatchShuffle fails and UnivFD succeeds. These generated images have well-generated local structures like textures but have defects in global structures such as ears, eyes, and faces. However, there are very few examples corresponding to this. This analysis indicates that using both local and global information is necessary for detecting generated images.

4.3. Robustness Against Image Degradation

Applying a Gaussian blur and JPEG compression to an image is a common degradation that can naturally occur. Fig. 6 illustrates the impact of each attack on two subsets of generated images. The diffusion-subset and GAN-subset are subsets of diffusion and GAN generators, respectively, drawn from the conventional benchmark. Gaussian indicates the addition of a Gaussian blur with a standard deviation of σ . JPEG indicates the application of JPEG compression with a specified compression quality. Note that JPEG



(a) Fake image examples on the second quadrant of Fig. 4a.



(b) Fake image examples on the fourth quadrant of Fig. 4a.

Figure 5. A closer look into the failure cases from the StyleGAN-generated test images.

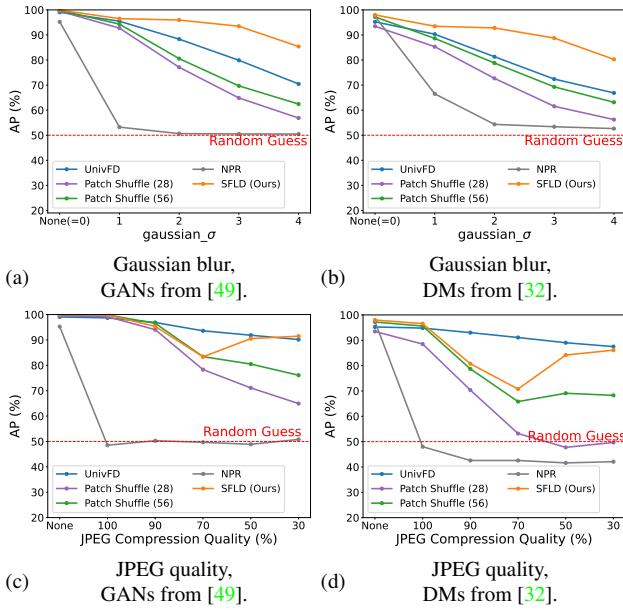


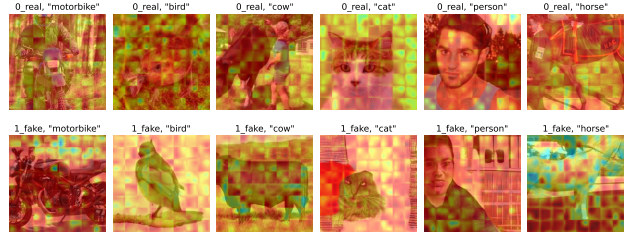
Figure 6. Robustness against simulated image degradation. Methods include Gaussian blur and JPEG compression.

compression with quality 100 does not result in the same image, as JPEG compression reduces color information and rounds coefficients, thereby losing some information.

If the model is vulnerable to image degradation, we can infer that it is influenced by the features targeted by the degradation. Specifically, Gaussian blur affects both high- and low-level features in the image, while JPEG compression primarily targets low-level features (see Fig. 13). Figs. 6a and 6b demonstrates that SFLD always shows the best performance against Gaussian blur, since it integrates both high- and low-level features through ensemble/fusion, enabling each to compensate for the information lost in the other. Figs. 6c and 6d illustrates that SFLD restores robustness against JPEG compression, supporting the fundamental principle behind our model. Additionally, Uni-



(a) UnivFD [32] examples



(b) Patch-shuffle 28x28 examples

Figure 7. Class activation maps (CAM) for UnivFD [32] and the patch-shuffled detector (ours). GradCAM [15, 43] was used to obtain the heatmaps. The ground truth real/fake labels and class labels are displayed on top of each image. Note that for Fig. 7b, the heat map is split into patches then reverse-shuffled back to the corresponding spatial location of the input image.

vFD, which focuses on capturing high-level feature artifacts is also robust against JPEG compression. However, NPR, which focuses on capturing low-level feature artifacts, is vulnerable to both Gaussian blur and JPEG compression even at JPEG compression quality 100.

4.4. Qualitative Analysis

GradCAM visualization. See Fig. 7 for image attribution heat maps generated using GradCAM [15, 43]. The examples are from the ProGAN test set. In addition, the heat maps are averaged across ten predictions to reduce the randomness from the patch permutation. The CAM of UnivFD focuses on the class-dependent salient region, whereas the patch-shuffled detector focuses on the entire image region.

Feature visualization. Because taking an average of the logits generated via a linear layer is equivalent to taking an average of the feature embeddings, we can understand the SFLD embeddings by taking the average of the embeddings over multiple shuffles. Fig. 8 visualizes the feature embeddings by projecting onto a 2D plane using UMAP [42]. We used the ProGAN test set to extract the embeddings.

Because UnivFD learns the features directly from the CLIP visual encoder, the embeddings form class-dependent clusters. This creates class-dependent decision boundary, which may introduce unintended content bias to the real-fake detector. In contrast, because PatchShuffle destroys

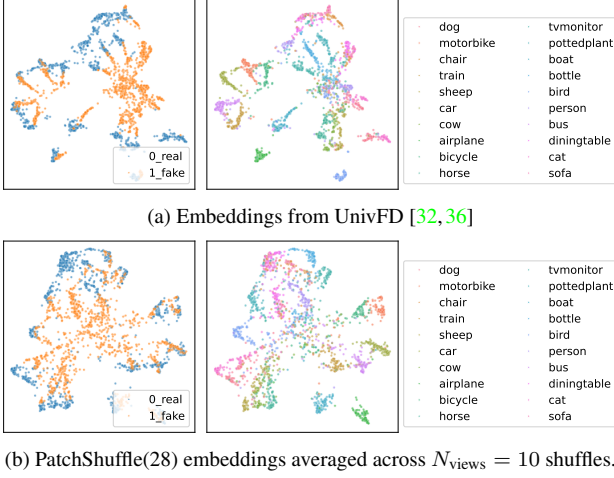


Figure 8. UMAP visualization [42] of feature embeddings. Left and right plots show the same projected embeddings colored by real/fake labels (left) and object category labels (right). Our method destroys the class information from the embeddings, thereby improving the generalization by reducing the content bias.

class-related information from the image, the corresponding embeddings show more dispersion within each class.

4.5. Effect of PatchShuffle Hyperparameters

Improving feature extraction with PatchShuffle. We suggest additional details to get better CLIP features from the shuffled images. To improve stability against the randomness introduced by PatchShuffle, we use the averaged logits of $N_{\text{views}} = 10$ randomly shuffled patch combinations for each input image during testing.

Moreover, in our problem setup, training images are fixed at 256×256 size, while test images can vary in size. Resizing test images is avoided, as image degradation due to resizing (e.g., JPEG compression or blur) has been shown to impact the detection of AI-generated images negatively [49]. Instead, recent detectors [32, 46] prefer cropping over resizing. Our backbone model without PatchShuffle also extracts CLIP features from 224×224 center-cropped images without resizing. However, we can extract information not only from the center of the image but from the entire image by taking advantage of the proposed PatchShuffle, which allows non-consecutive patchwise combinations. We divide the entire test image into non-overlapping patches of the given patch size and combine these patches into 224×224 images. This approach enables the detector to analyze information from the entire image, rather than being constrained to a single central region. See Appendix D for more details.

Patch size. The optimal patch size should be sufficiently small to disrupt the underlying image structure while preserving some high-level feature artifacts. The results for the performance difference according to patch sizes on a con-

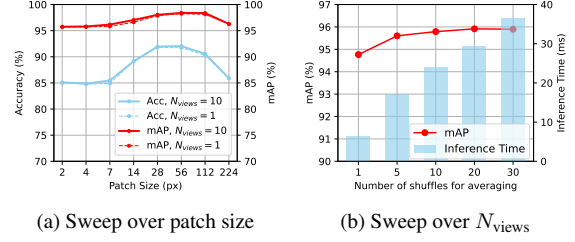


Figure 9. Best patch sizes were found at 28×28 and 56×56 . $N_{\text{views}} = 10$ showed the best balance between performance and inference cost.

ventional benchmark are presented in Fig. 9a. Each patch size model in x-axis refers to the ensemble between the corresponding PatchShuffle model and UnivFD(patch size 224). It can be observed that an too small patch size and an excessively large patch size do not assist the model in capturing useful high-level and low-level feature artifacts. Therefore, the majority of experiments in this paper utilized patch sizes 28×28 and 56×56 according to this result.

Number of shuffled views. To ensure the stability of the random patch shuffle, SFLD generates multiple versions of shuffled image from a single test image and employs the average of them as the score. As illustrated in Fig. 9b, mAP enhances with higher N_{views} . However, due to the tradeoff with inference time, we chose $N_{\text{views}} = 10$, and all results presented in this paper were obtained with this setting. The results in Fig. 9b are from the PatchShuffle model with a patch size of 28, without an ensemble with UnivFD. The inference time was measured using RTX 4090 GPU.

5. Conclusions

In this paper, we introduced SFLD, a novel method for detecting AI-generated images that effectively combines global semantic structures and textural structures to improve detection performance. By leveraging random patch shuffling and an ensemble of classifiers trained on patches of varying sizes, our approach effectively addresses the shortcomings of existing methods, such as their content bias and susceptibility to image perturbations. Also, We proposed a new quality-ensuring benchmark, TwinSynths. It is the first to consider a scenario of infinitely real-like fake images, providing a valuable resource for future research in this area. We demonstrated that SFLD outperforms SOTA methods in generalization to various generators, even in challenging scenarios simulated with TwinSynths.

Acknowledgements This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II212068, Artificial Intelligence Innovation Hub).

References

- [1] Ali Borji. Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2. In *Arxiv*, 2022. 14, 15
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv*, 2018. 12
- [3] George Cazenavette, Avneesh Sud, Thomas Leung, and Ben Usman. Fakeinversion: Learning to detect images from unseen text-to-image models by inverting stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10759–10769, 2024. 10
- [4] Lucy Chai et al. What makes fake images detectable? understanding properties that generalize. In *ECCV*, pages 103–120. Springer, 2020. 15
- [5] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. 12
- [6] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 12
- [7] Yunjei Choi et al. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 12
- [8] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 14
- [9] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019. 12
- [10] Boris Dayma et al. Dall- e mini. *Hugging-Face. com*. <https://huggingface.co/spaces/dallemini/dallemini> (accessed Sep. 29, 2022), 2021. 12
- [11] Deepfakes. Deepfakes/faceswap: Deepfakes software for all. 14
- [12] Prafulla Dhariwal et al. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 12
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2
- [14] Joel Frank et al. Leveraging frequency analysis for deep fake image recognition. In *ICML*, pages 3247–3258. PMLR, 2020. 4, 5, 14
- [15] Jacob Gildenblat and contributors. Pytorch library for cam methods. <https://github.com/jacobgil/pytorch-grad-cam>, 2021. 7, 12
- [16] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hananeh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. 15
- [17] Yan Ju et al. Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3465–3469. IEEE, 2022. 4, 5
- [18] Tero Karras et al. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 4, 12
- [19] Tero Karras et al. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 12
- [20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 12
- [21] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4220–4229, 2019. 12
- [22] Tsung-Yi Lin et al. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 12
- [23] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *European Conference on Computer Vision*, pages 95–110. Springer, 2022. 4, 5
- [24] Ziwei Liu et al. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 12
- [25] Zhengzhe Liu et al. Global texture enhancement for fake face detection in the wild. In *CVPR*, pages 8060–8069, 2020. 4, 5
- [26] Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lare²: Latent reconstruction error based method for diffusion-generated image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17006–17015, 2024. 14
- [27] Francesco Marra et al. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 506–511. IEEE, 2019. 14
- [28] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018. 14
- [29] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM workshop on information hiding and multimedia security*, pages 43–47, 2018. 14
- [30] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting gan generated fake images using co-occurrence matrices. *Electronic Imaging*, 31:1–7, 2019. 14

- [31] Alex Nichol et al. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 12
- [32] Utkarsh Ojha et al. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15
- [33] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 15
- [34] Taesung Park et al. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019. 12
- [35] Alec Radford et al. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4
- [36] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 8
- [37] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes, 2024. 14
- [38] Robin Rombach et al. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 12
- [39] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision (ICCV)*, 2019. 14
- [40] Andreas Rossler et al. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the ICCV*, pages 1–11, 2019. 12
- [41] Olga Russakovsky et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4, 12
- [42] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning. *ArXiv e-prints*, 2020. 7, 8
- [43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020. 7, 12
- [44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*. 4
- [45] Chuangchuang Tan et al. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR (CVPR)*, pages 12105–12114, June 2023. 4, 5
- [46] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28130–28139, 2024. 1, 4, 5, 8, 15
- [47] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. Gan is a friend or foe? a framework to detect various fake face images. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1296–1303, 2019. 14
- [48] Rafael Valle, Wilson Cai, and Anish Doshi. Tequilagan: How to easily identify gan samples. *arXiv preprint arXiv:1807.04919*, 2018. 14
- [49] Sheng-Yu Wang et al. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020. 1, 2, 4, 5, 6, 7, 8, 12
- [50] Zhendong Wang et al. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22445–22455, October 2023. 14
- [51] Fisher Yu et al. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 12
- [52] Ning Yu et al. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the ICCV*, pages 7556–7566, 2019. 14
- [53] Yichi Zhang and Xiaogang Xu. Diffusion noise feature: Accurate and fast generated image detection. *arXiv preprint arXiv:2312.02625*, 2023. 14
- [54] Nan Zhong, Yiran Xu, Zhenxing Qian, and Xinpeng Zhang. Rich and poor texture contrast: A simple yet effective approach for ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023. 4
- [55] Jun-Yan Zhu et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 12
- [56] Mingjian Zhu, Hanting Chen, Mouxiao Huang, Wei Li, Hailin Hu, Jie Hu, and Yunhe Wang. Gendet: Towards good generalizations for ai-generated image detection. *arXiv preprint arXiv:2312.08880*, 2023. 1, 15