IPD-Net: Detecting AI-Generated Images via Inter-Patch Dependencies

Jiahan Chen[1], Mengtin Lo[2], Hailiang Liao[3], Tianlin Huang[4]*
College of Cyber Security, Jinan University, Guangzhou, China[1,2,3]
School of Physics and Telecommunications Engineering, Yulin Normal University, Yulin, China[4]

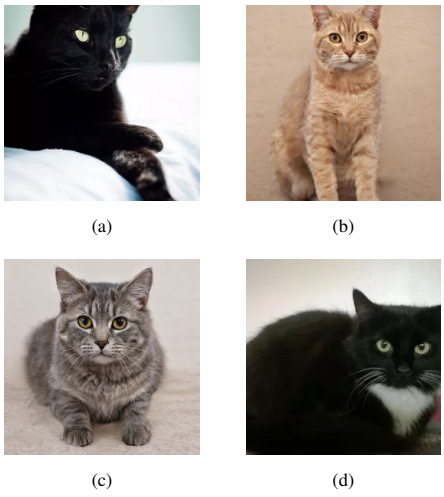
***Abstract*—With the rapid development of generative models, the fidelity of AI-generated images has almost reached a level that is difficult for humans to distinguish true from fake. The rapid development of this technology may lead to the widespread dissemination of fake content. Therefore, developing effective AI-generated image detectors has become very important. However, current detectors still have limitations in their ability to generalize detection tasks across different generative models. In this paper, we propose an efficient and simple neural network framework based on inter-patch dependencies, called IPD-Net, for detecting AI-generated images produced by various generative models. Previous research has shown that there are inconsistencies in the inter-pixel relations between the rich texture region and the poor texture region in AI-generated images. Based on this principle, our IPD-Net uses a self-attention calculation method to model the dependencies between all patches within an image. This enables our IPD-Net to self-learn how to extract appropriate inter-patch dependencies and classify them, further improving detection efficiency. We perform experimental evaluations on the CNNSpot-DS and GenImage datasets. Experimental results show that our IPD-Net outperforms several state-of-the-art baseline models on multiple metrics and has good generalization ability.**

***Keywords*—AI-generated image detection; image forensics; self-attention mechanism**

I. Introduction

In recent years, generative model technology has achieved rapid development. As shown in Fig. 1, the quality of AI-generated images is getting higher and higher. Various generative models such as VAE [1], GAN [2] and their derivative models continue to emerge. Ho et al. [3] provided rigorous mathematical derivation for the diffusion model, and then Dhariwal et al. [4] made the diffusion model gradually become the most mainstream generative model together with GAN, and promoted many derivative models. AI-generated images are becoming more and more realistic and difficult to distinguish with the naked eye, which opens up a wide range of possibilities for a variety of application scenarios. However, the development of this technology has two sides, and there have been some egregious incidents of malicious use of generative models to generate fake images. Because of this, in the face of the continuous evolution of future generative models, there is an urgent need to develop a universal detection method to distinguish AI-generated images from real ones.

A simple strategy is to use an existing multi-class CNN such as ResNet [5] for the binary classification task. However, when this method detects the generative model that is seen during training, it can recognize AI-generated images from the real images effectively, but its accuracy is significantly reduced in the detection across the generative model. CNNSpot [6] shows that with careful pre- and post-processing and data augmentation, a standard image classifier trained on a specific CNN-generated image training set can be extended to detect unseen GAN-generated image detection tasks. However, this method is found to perform well within the same family of generative models, but its generalization ability is limited when detected across different families [7]. For example, a model trained on a dataset containing images generated by ProGAN [18] (a variant of GAN) and real images, when tested on a dataset containing images generated by SD v1.4 [30] (a variant of diffusion models) and real images, shows a sharp decline in accuracy compared to detection within the same generative family. UnivFD [7] further points out that the previous method [6] relies mainly on the common features of the AI-generated images of the generative model seen during training to classify images as "fake" or "true". Therefore, they propose to use untrained features to distinguish AI-generated images from real images and use a frozen large pre-trained vision-language model for classification. This method significantly improves the generalization ability of detection models on unknown generative models. However, because real images cover a large number of categories, determining a general classification range becomes a challenge, which may affect the classification accuracy of unknown generative models.



(a) (b) (c) (d)

Fig. 1. Can you determine which are real images and which are AI-generated images? Where (a) and (d) are real images, and (b) and (c) are AI-generated images.

*Corresponding authors.

The diversity of real images makes it difficult to place them in a single category. Therefore, some detection methods try to distinguish real images from AI-generated images by finding common features among multiple generative models. However, with the continuous evolution of the field of the generative model, some early methods fail to generalize well to new models [6], [8], [9], [10]. In this paper, we propose IPD-Net, which can extract features from noise patterns of the pre-processed image and model the dependencies between all patches in the image by computing the dot product similarity between all vectors. The dependencies matrix is then classified into binary categories using a specially designed classification layer to determine whether the input image is an AI-generated image. Experimental results show that our proposed IPD-Net has a stronger generalization ability in detecting AI-generated images compared to baseline models.

In general, our main contributions are as follows:

- We propose IPD-Net, a novel neural network framework for AI-generated image detection based on inter-patch dependencies. IPD-Net can generalize to the detection of images generated by unseen generative models and has a fast inference speed.
- Unlike methods that directly segment pre-processed images into multiple patches and compute relationships between the patches, our proposed IPD-Net calculates the dot-product similarity between all vectors in the feature map using a self-attention mechanism, thereby modeling dependencies between patches in the image. In addition, we design dedicated classification layers to classify the modeled inter-patch dependencies to determine whether the image is AI-generated.
- We collect a highly diverse dataset containing images from various resolutions, operation types, and a wide range of generative models for evaluating our approach. Experimental results show that IPD-Net outperforms baseline models on multiple metrics and it has good generalization ability.

II. Related Work

In the following, we present an overview of related work in terms of both AI-generated image techniques and AI-generated image detection techniques. Additionally, we discuss the connections to our approach.

A. AI-Generated Image Techniques

In recent years, AI-generated image techniques have made great progress and caused a lot of concern. Among them, the Generative Adversarial Network (GAN) model [2] is one of the early important generative models. Its basic principle involves adversarial training between a generator and a discriminator, where the generator is responsible for generating images, and the discriminator is used to discriminate the authenticity of the images. As training progresses, the fidelity of the generated images increases and eventually reaches a very high level. The success of this technique has spawned many variants, such as [19], [20]. Ho et al. [3] brought rigorous mathematical derivation to the diffusion model, leading to its wider application in the field of image generation. The basic idea of the diffusion model is to gradually add noise to the data during the forward process and then learn to restore the original data from the noise during the reverse process. This technique has also produced many related models [30], [31], [32]. In contrast, the goal of our proposed IPD-Net is that, after training on the AI-generated image training set of a specific generative model, it can be generalized to other unknown generative models to perform AI-generated image detection tasks, thus better generalized to real-world scenarios.

B. AI-Generated Image Detection Techniques

With the rapid advances in generative techniques, modern AI-generated images have reached a level nearly indistinguishable from real images. Although generative technology has brought convenience to some industries such as AI mapping, like the two sides of a coin, this also brings potential social risks. For example, highly realistic AI-generated images could be exploited by criminals as a medium to disseminate fake information, thereby causing social problems. Therefore, the research and development of AI-generated image detection technology is particularly urgent.

Wang et al. [6] constructed a dataset containing AI-generated images from 11 different generative models based on CNN. For the construction of the training set, they trained 20 ProGAN [18] models, each trained on a different LSUN [17] object class. For each trained ProGAN model, 36K (for training) +200 (for validation) AI-generated images are generated, and the corresponding images for training ProGAN models are used as the real class, and the resulting training set and validation set contain the same number of true/fake images. Therefore, the resulting training set has a total of 720k images and the validation set has 4k images. Through careful pre-processing and data augmentation, they trained a binary classifier using a ResNet50 [5] pre-trained on ImageNet [29], and tested on a dataset of true/fake images collected from 11 different generative models. Experimental results show that even standard image classifiers trained for specific CNN generators can generalize over unseen generative model detection. Additionally, several other works [8], [9] investigated the frequency domain of GAN-generated images and leveraged the frequency domain for detection. Before the diffusion model became popular, most researchers focused on identifying GAN-generated images. However, these efforts were later found to be difficult to generalize to detecting AI-generated images from more recent generative models [7], such as diffusion models. With the rise of diffusion models, many previous detection methods have difficulty identifying this emerging model. Wang et al. [34] found that, unlike real images, images generated by diffusion models can be reconstructed through pre-trained diffusion models. So they use the error between the reconstructed image and the input image to detect AI-generated images. However, this method mainly focuses on diffusion models. Ojha et al. [7] found that although detection methods trained on ProGAN [6] generalize well when tested on the same generation model family (GAN model family), their accuracy significantly drops when tested on different generative model family (diffusion model family). Previous methods [6] mainly relied on features of seen models to classify images. Therefore, Ojha et al. [7] proposed using untrained features to distinguish AI-generated images from real ones and using a frozen large pre-trained vision-language

model for classification, thus enhancing the generalization of the detection model to the unknown generation models, but this leads to a slower inference speed. Zhong et al. [14] used the inconsistency between rich and poor texture regions in AI-generated images as a universal fingerprint. Chen et al. [35] proposed using the noise pattern of the simplest patch of the input image to identify AI-generated images. However, both approaches require selecting specific patches from a large number of patches through mathematical calculations before sending them into the neural network. For this reason, we hope that our IPD-Net to not only adapt to the detection tasks of the above generation models but also achieve ultra-fast forward inference speed to attain efficient scalability.

III. Our Method

A. Motivation

To ensure that a detector trained on a specific GAN-generated image training set can be generalized to other GAN-generated image detection, or even generalized to the detection task of images generated by different families of generative models, it is crucial to find common features of fake images. For example, Zhong et al. [14] found that AI-generated images processed with a carefully selected set of SRM filters [15] (a type of high-pass filter with fixed parameters) have inconsistencies in inter-pixel relations between the rich texture region and the poor texture region. Inspired by [14], we argue that there exists some kind of dependencies between different patches of AI-generated images processed by the SRM filter, which can be used as common features of the AI-generated images, and such dependencies are not limited to those between rich texture patches and poor texture patches. To extract these dependencies efficiently, we designed IPD-Net that enables the model to capture interdependencies from all patches. To avoid the huge computational overhead associated with actively selecting specific patches, we are inspired by Wang et al. [16], who proposed a self-attention mechanism that, in the process of computing the weight matrix, can be viewed as modeling dependencies between patches of the preprocessed image. Based on this process, we discarded the softmax operation and performed spatial transformations to avoid the step of actively selecting patches. In this way, we can obtain the dependencies between all patches we need. Next, we specifically designed a classification layer so that the obtained dependencies can be directly used for classification. This design enables the model to achieve end-to-end training, allowing it to self-learn how to extract suitable dependencies and perform classification, thus further improving detection efficiency. Through this method, our model can not only recognize AI-generated images generated by generative models seen in training data but also can be generalized to other unseen generative model image detection tasks, improving the generalization ability of the detector.

B. Inter-Patch Dependencies Extraction

As shown in Fig. 2, our network architecture is divided into two parts: Inter-Patch Dependencies Extraction and Inter-Patch Dependencies Classification. In the Inter-Patch Dependencies Extraction part, our neural network aims to extract dependencies between image patches processed by the SRM filter. Therefore, firstly the pre-processed input image needs to be processed using the SRM filters. This filter has been widely adopted in the field of fake image detection [12], [13]. To validate our IPD-Net of generality, we chose the same as the [12], [13], general SRM filter configuration. In terms of the computation of inter-patch dependencies, we are inspired by the self-attention computation method proposed by Wang et al. [16], where the computation process can be viewed as calculating a correlation score between each patch in the image and all other patches (including itself). Specifically, each patch is processed by the neural network to become a feature vector. We input the noise patterns processed by the SRM filter into the backbone to obtain a feature map of size $C \times H \times W$, where C, H, and W represent the number of channels, height, and width of the feature map, respectively. For each patch P_i, the corresponding feature vector extracted by the backbone is denoted as e_i, and its size is C. For the relationship between any two patches, such as P_i and P_j, we use their corresponding feature vectors e_i and e_j to calculate their dot product similarity to represent the dependencies between them, as described by the following equation:

$$\text{Dependency}(e_i, e_j) = f(e_i) \cdot f(e_j) \tag{1}$$

Where f represents a 1×1 convolution. We perform this operation for all feature vectors, i.e., calculating their dot product similarity with all other feature vectors, resulting in a dependencies matrix of size $(H \times W) \times H \times W$. This can be viewed as $H \times W$ two-dimensional dependencies matrices of size $H \times W$, where each point represents the dependency score between the feature vector e_i of a certain patch P_i and the feature vector of another patch. Through this step, we successfully extract the inter-patch dependencies matrix.

C. Inter-Patch Dependencies Classification

After extracting the inter-patch dependencies matrix for each input image, to enable end-to-end training of the model, we classify the inter-patch dependencies and use the classification loss to optimize the model training. However, directly flattening the dependencies matrix for linear classification would result in huge computational overhead. Convolution is also unsuitable because the dependencies matrix is not a traditional feature map, and direct convolution may destroy it. Therefore, we adopt a 2D AdaptiveAvgPool operation. For the dependencies matrix of size $(H \times W) \times H \times W$, we can regard it as the feature matrix of $C' \times H \times W$, where $C' = H \times W$, is regarded as the number of channels, and H and W can be regarded as the feature matrix height and width. At this point, any channel represents a two-dimensional matrix of dependencies between a particular feature vector and all other feature vectors (including itself). Given this, we perform a two-dimensional average pooling operation on the inter-patch dependencies matrices to scale the size to $(H \times W)$ to directly extract the average dependency scores between each patch and other patches. Subsequently, since the inter-patch dependencies matrix is processed into vectors of size $(H \times W)$, it can directly be input into the linear classification layer. Meanwhile, the pooling operation significantly reduces the number of parameters in the model, further improving the inference speed. To validate the effectiveness of our IPD-Net, our linear classification layer uses only one or two linear

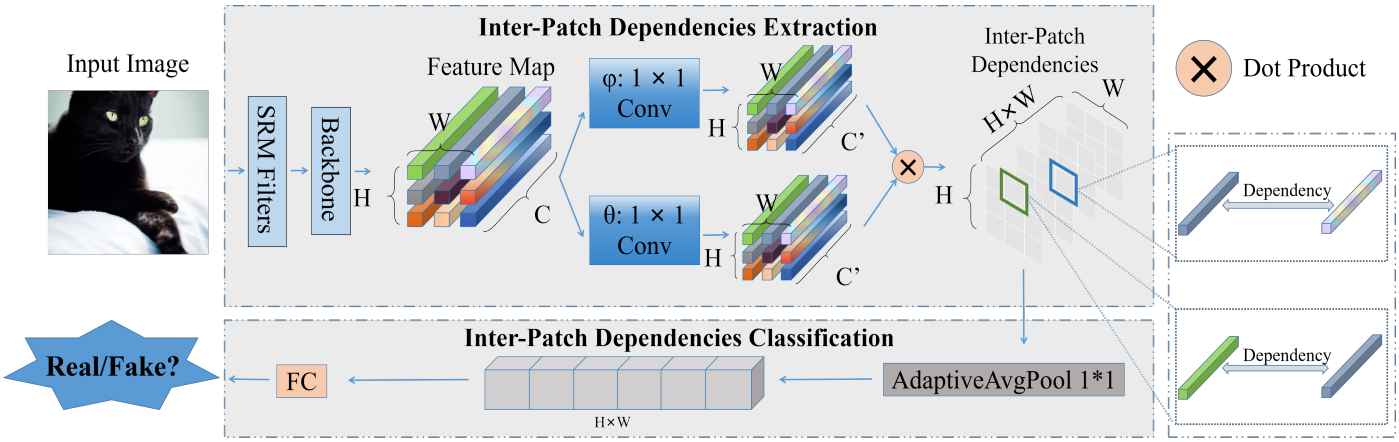


Fig. 2. Structure of IPD-Net. The preprocessed images are first processed by SRM filter and backbone to extract the dependencies between patches by a simple modified self-attention computation method. Subsequently, these dependencies are fed into a specially designed classification layer to determine whether the input image is an AI-generated image (Fake) or a real image (Real).

layers. The final output vector is processed with a sigmoid function to constrain the values to the $[0, 1]$, determining whether the input image is real or fake. We use Binary Cross Entropy Loss as the loss function, and our neural network is defined as $\mathrm{NN}(\cdot)$, formulated as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log(\mathrm{NN}(x_i)) + (1 - y_i) \log(1 - \mathrm{NN}(x_i))] \quad (2)$$

Where x_i and y_i represent the input image and its corresponding label, respectively. The goal is to minimize this total loss during the training process to improve classification accuracy. Meanwhile, our network is more friendly to computation and memory due to the average pooling operation and simple linear classification operation.

IV. Experiment

A. Datasets

To fully evaluate the effectiveness of our proposed IPD-Net, we conducted experiments using the CNNSpot-DS [6] and GenImage dataset [11]. The AI-generated image in the former is mainly composed of the image generated by the GAN model, and the AI-generated image in the latter is mainly composed of the image generated by the diffusion model. We follow the same protocol as described in the baselines [6], [7], the training set we use for training is the training set of CNNSpot-DS [6]. The AI-generated images in the training set were generated by ProGAN [18], and the training set contains a total of 720k images, which contains 360k real images and 360k AI-generated images, where the real images are from LSUN [17] dataset of 20 categories. We only use it as the training set for all subsequent training, so our training and validation are restricted to only accessing the real/fake images of one generative model, and detecting other generative models that are not seen during training.

When evaluating the detector's ability, we considered various generative models. We tested the generative models on the test set of the CNNSpot-DS following the baselines [6], [7]: ProGAN [18], StyleGAN [19], BigGAN [20], CycleGAN [21], StarGAN [22], GauGAN [23], CRN [24], IMLE [25], SAN [26], SITD [27], and DeepFakes [28]. Additionally, we tested the test set of the GenImage dataset [11], which mainly contains many AI-generated images generated by diffusion models. Zhu et al. [11] used ImageNet [29] to generate 1.3 million AI-generated images. We tested the generation models in the GenImage dataset: Midjourney*, SDV1.4 [30], SDV1.5 [30], ADM [4], GLIDE [31], Wukong†, VQDM [32], and BigGAN [20].

B. Implementation Details

All training is implemented on an NVIDIA GeForce RTX 3090 GPU and an Intel Xeon Gold 6238R CPU. Our model is implemented using PyTorch [33] and the batch size was set to 32. We optimize using Adam with a learning rate of 0.0001. For the SRM filters [15], we follow the settings from [12], [13], adopting the three commonly used kernels from the original SRM filters [15]. To model the inter-patch dependencies of a feature map processed by the backbone, assuming the input feature map is $C \times H \times W$, we use two different 1×1 convolutions to process the feature map separately, reducing the number of channels to half of the original, and obtaining two different feature maps with sizes of $\left(\frac{C}{2}\right) \times H \times W$. We reshape them to $\left(\frac{C}{2}\right) \times (H \times W)$, converting them into two-dimensional matrices. We transpose one of the feature maps and then multiply it with another feature map to obtain a product f with a size of $(H \times W) \times (H \times W)$. After that, we transpose f once and reshape its size to $(H \times W) \times H \times W$ to perform the next step of the average pooling operation. We select a non-trained ResNet50 [5] as the backbone for feature extraction. In the design of the backbone, we consider three variants: ResNet50-Layer2, ResNet50-Layer3, and ResNet50-Layer4, where Layer2, Layer3, and Layer4 denote the layers after which truncation is applied. The input image can be of any size, we apply reflect padding to add 224 pixels on all sides of the image, then crop out 224 pixels and resize it to 256 pixels. During training, after resizing, we apply

*Midjourney, https://www.midjourney.com/home/. 2022.
†Wukong, https://xihe.mindspore.cn/modelzoo/wukong. 2022.

TABLE I. EVALUATION RESULTS. AVERAGE PRECISION (AP) OF DIFFERENT TRUE/FAKE IMAGE DETECTION METHODS. WE REPORT MEAN AVERAGE PRECISION (MAP) BY AVERAGING THE AP SCORES FOR EACH GENERATIVE MODEL DETECTION METHOD

Detection method	Variant	Generative Adversarial Networks							Low level vision		Perceptual loss		GenImage [11]								Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN	Deep-fakes	SITD	SAN	CRN	IMLE	Mid-journey	SD-v1.4	SD-v1.5	ADM	GLIDE	Wu-kong	VQDM	Big-GAN	mAP
CNNSpot[6]	Aug(0.1)	**100.0**	92.42	83.20	99.60	87.79	98.18	90.69	68.64	53.84	98.78	98.67	58.32	59.14	59.42	72.27	66.86	54.83	60.21	86.75	78.40
	Aug(0.5)	99.99	95.33	88.90	98.85	97.30	96.01	68.48	85.93	56.36	**99.36**	**99.56**	51.16	54.29	54.38	67.94	66.68	51.62	67.80	94.10	78.63
Fusing[10]	-	**100.0**	97.52	95.95	98.80	96.43	99.65	65.52	**88.36**	72.58	98.50	99.21	68.06	61.08	61.13	90.31	65.85	62.82	77.72	95.65	83.95
UnivFD[7]	-	**100.0**	**99.80**	**99.27**	97.56	**99.98**	99.37	81.76	63.84	78.81	96.59	98.61	74.61	86.56	86.19	87.13	84.26	91.34	**96.65**	98.21	90.55
Ours	Layer2	**100.0**	96.50	89.00	99.76	79.40	**99.94**	89.37	80.26	91.38	90.23	91.90	86.53	94.63	94.46	94.08	99.27	92.36	91.98	99.14	92.64
	Layer3	99.99	96.51	89.99	**99.86**	84.36	99.93	91.83	79.87	93.21	87.61	92.03	**86.55**	94.41	94.11	**95.05**	99.15	**92.69**	93.32	**99.34**	**93.15**
	Layer4	99.99	97.73	88.82	99.58	78.86	99.92	**93.67**	73.47	**94.15**	85.35	84.75	85.70	**94.89**	**94.56**	94.26	**99.38**	92.35	91.86	99.12	92.02

TABLE II. EVALUATION RESULTS ACCURACY (ACC) OF DIFFERENT TRUE/FAKE IMAGE DETECTION METHODS. WE REPORT AVERAGE ACCURACY (AVG. ACC) BY AVERAGING THE ACC SCORES FOR EACH GENERATIVE MODEL DETECTION METHOD

Detection method	Variant	Generative Adversarial Networks							Low level vision		Perceptual loss		GenImage [11]								Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN	Deep-fakes	SITD	SAN	CRN	IMLE	Mid-journey	SD-v1.4	SD-v1.5	ADM	GLIDE	Wu-kong	VQDM	Big-GAN	Avg. acc
CNNSpot[6]	Aug(0.1)	99.97	85.08	70.52	88.98	78.66	92.24	57.74	61.94	49.77	80.40	80.35	53.59	53.08	53.15	60.61	56.54	51.48	53.61	77.33	68.69
	Aug(0.5)	99.97	82.28	62.12	73.72	81.82	81.81	51.26	56.38	50.22	**95.64**	**96.80**	50.43	49.97	49.98	52.58	52.78	50.08	52.53	71.25	66.40
Fusing[10]	-	**99.98**	91.67	82.82	80.06	83.54	96.97	54.41	72.77	52.28	93.93	95.85	52.02	50.56	50.53	54.20	54.46	51.03	57.32	84.90	71.54
UnivFD[7]	-	99.81	**98.33**	**95.08**	84.93	**99.47**	95.75	68.57	62.22	56.62	56.59	69.11	56.24	63.75	63.57	66.94	62.53	71.06	85.42	90.18	76.11
Ours	Layer2	**99.98**	86.57	81.02	**95.19**	68.67	**99.08**	62.11	79.11	71.73	64.73	64.42	**72.19**	**80.03**	**79.70**	84.38	95.05	**77.19**	**79.37**	**93.75**	80.75
	Layer3	99.97	86.51	81.52	94.71	72.91	98.61	63.75	77.43	71.57	70.38	70.41	69.36	75.48	74.92	82.87	93.23	74.21	77.50	93.52	80.47
	Layer4	99.94	89.87	80.12	94.74	67.82	98.42	**69.42**	**81.87**	**81.05**	63.53	63.37	70.14	79.64	78.89	**84.42**	**95.56**	76.61	78.05	93.13	**81.40**

Gaussian blur with $\sigma \sim \text{Uniform}[0, 3]$ with 10% probability, JPEG compression with quality $q \sim \text{Uniform}\{30, 31, \ldots, 100\}$ with 10% probability. In addition, we added random flip with 50% probability, and the above crop operations use random crop. During testing, our crop operations uniformly use center crop. For the classification part, we set up two linear layers for ResNet50-Layer2 and one linear layer for both ResNet50-Layer3 and ResNet50-Layer4.

For the baseline methods, we used CNNSpot [6], Fusing [10] and UnivFD [7]. For UnivFD [7], we tested using its publicly published training weights and open-source code. And for CNNSpot [6] and Fusing [10], we trained from scratch with the training dataset, following the settings in their open-source code. In evaluating our model and the baseline methods, we used Average Precision (AP) and Accuracy (ACC) to evaluate our model, which is consistent with recent related work [7]. We report the mean Average Precision (mAP) and Average Accuracy (Avg. acc) of each detector by averaging the AP scores and ACC scores obtained when each detector was tested against each test set of the generative model.

C. Evaluations

1) Combined dataset evaluation: Following the indicator setting of recent baselines [7], Table I and Table II present the average precision (AP) and accuracy (acc) of real/fake image detection by the baseline models and our IPD-Net (rows) for different generative models (columns). Following the training setting of recent baseline [6], [7], all our training is performed on the training set of CNNSpot-DS [6], the AI-generated images in the training set were all only generated by ProGAN. Therefore, models other than ProGAN can be considered as generalization domains. In the variant setting, Aug (0.1) and Aug (0.5) represent two training configurations of CNNSpot [6] open-source code, which apply JPEG compression and Gaussian blur with 10% or 50% probability, respectively. "Ours" represents our model's test results, and Layer2, Layer3, Layer4 correspond to the three backbone variants mentioned in Section IV-B, namely, ResNet50-Layer2, ResNet50-Layer3 and ResNet50-Layer4. The settings of the other two baseline methods [7], [10] are the same as those in the papers and open-source codes. Compared to the three baselines, all three variants of our proposed IPD-Net achieved better mAP and average accuracy. The mAP of our three variants improved by 1.47-2.6% over the best-performing baseline, and the average accuracy improved by 4.36-5.29% over the best baseline. Our model performs worse on Perceptual loss compared to other baseline models. We speculate that GAN models and Perceptual loss share some common features, which the baseline models may tend to fit. However, this common feature does not apply to diffusion models. In contrast, the common feature self-learned by our model is common to both the GANs model and the diffusion models, except that it is less general on Perceptual loss. Overall, our IPD-Net achieves the best performance in terms of mAP and average accuracy in combined dataset evaluation, indicating that our model has stronger generalization ability compared to the baseline models.

2) In-dataset and cross-dataset evaluation: To more effectively reflect the generalization ability of our proposed IPD-Net, we conducted in-dataset and cross-dataset evaluation. As shown in Table III, we analyzed the accuracy of the test set of CNNSpot-DS [6] and GenImage [11] datasets respectively. Among them, the accuracy of the CNNSpot-DS is an in-

TABLE III. EVALUATION RESULTS ACCURACY OF THE CNNSPOT-DS AND GENIMAGE DATASET. WE REPORT ACCURACY BY AVERAGING THE ACCURACY SCORES FOR EACH GENERATIVE MODEL DETECTION IN THE CORRESPONDING DATASET IN TABLE II FOR EACH DETECTOR

Detection method	CNNSpot-DS [6]	GenImage [11]
	ACC	ACC
CNNSpot [6]	76.88	57.42
Fusing [10]	**82.21**	56.88
UnivFD [7]	80.59	69.96
Ours_Layer4	79.33	**82.71**

dataset evaluation, and the accuracy of the GenImage dataset is a cross-dataset evaluation. We report accuracy by averaging the accuracy scores for each generative model detection in the corresponding dataset in Table II for each detector. In the in-dataset evaluation, that is the evaluation on the test set of CNNSpot-DS. Because the AI-generated images in the training set are all generated by ProGAN, the trained baseline models still have high accuracy in detecting GAN variants. The CNNSpot-DS's test set is mainly generated by a large number of GAN-generated images, so all detectors achieved high accuracy. The in-dataset accuracy between all detectors ranged from 76-82%. In the cross-dataset evaluation, that is the evaluation on the test set of the GenImage dataset. When the baseline models are faced with the GenImage dataset's test set mainly containing a large number of images generated by the diffusion model, their detection accuracy drops significantly. Among them, the cross-dataset accuracy of CNNSpot [6] and Fusion [10] is even between 50-60%. In comparison, our IPD-Net's cross-dataset accuracy is 12.75% higher than the best baseline model. Overall, the results show that the generalization ability we demonstrated in the combined dataset evaluation is effective both within and across datasets, further demonstrating that our IPD-Net has a stronger generalization ability than other baseline models.

D. Effect of Different Backbone

When we design the IPD-Net backbone network, as the number of backbone network layers decreases, in the feature map extracted by the backbone network, the area corresponding to the original image for each feature vector becomes smaller. Therefore, we speculate that as the number of layers decreases, IPD-Net can learn to extract more detailed inter-patch dependencies, thereby achieving better results. In this section, we will study what happens when our proposed IPD-Net selects different backbone networks for feature extraction. We consider the three variants mentioned in Section IV-B: (i) ResNet-Layer2, (ii) ResNet-Layer3, and (iii) ResNet-Layer4. We trained each model again using the same ProGAN real/fake image training set as in the above experiments.

To better analyze these three different variants, we provide a visual analysis of these three variants. We saved the vectors obtained by average pooling and flattening of the inter-patch dependencies, tested them on the CNNSpot-DS and GenImage dataset respectively, and drew six t-SNE diagrams, as shown in Fig. 3, where the true/fake labels were marked in red/blue respectively. From top to bottom represent three variants: (i) ResNet-Layer2, (ii) ResNet-Layer3, and (iii) ResNet-Layer4. The left column (a) represents the three variants tested on the CNNSpot-DS test set, and the right column (b) represents the

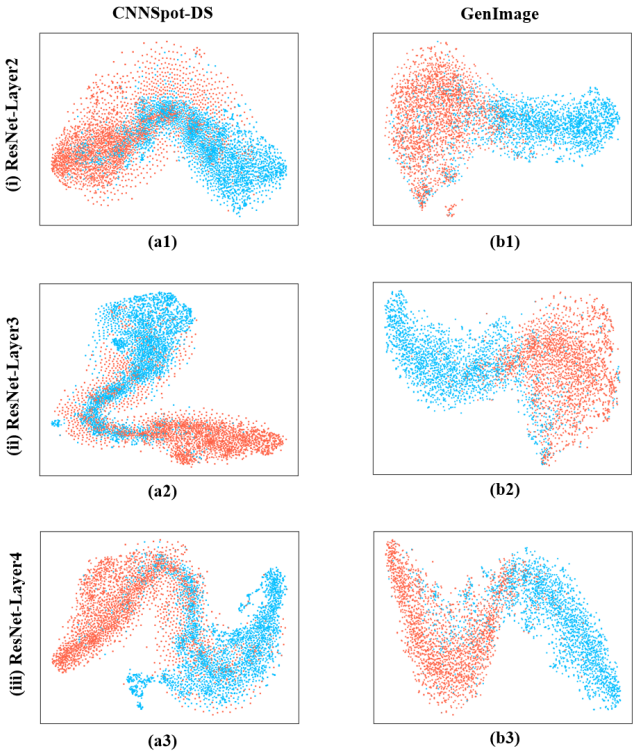


Fig. 3. Visualization results of three variants of t-SNE. True/fake labels are shown in red/blue, where the rows from top to bottom represent the three variants: (i) ResNet-Layer2, (ii) ResNet-Layer3, and (iii) ResNet-Layer4, left column (a) represents the results of the CNNSpot-DS test set, and the right column (b) represents the results of the GenImage test set.

results of the three variants tested on the GenImage test set. However, it can be seen from the t-SNE visualization results that with the increase in the number of layers, the inter-patch dependencies after average pooling and flattening can be better divided into true/fake, and the features of the same class are more concentrated, even though the difference between these three variants in mean average precision and average accuracy indicators seems to be very small. This suggests that deeper model structures may still have better results, although they are less detailed in dividing patches than shallow backbone networks.

E. Analysis of Limitations

We evaluated the robustness of our ResNet-Layer4 variant and the best-performing baseline model against jpeg compression and Gaussian blurring. Fig. 4 shows the mAP of both the ResNet-Layer4 variant and the best-performing baseline model under different post-processing configurations. Without any post-processing, the mAP of our method is significantly higher than that of the best-performing method. However, the robustness of our model to post-processing operations is significantly weaker than that of the best-performing baseline model, especially in the case of JPEG compression. We speculate that this may be because the way of extracting inter-patch dependencies in our IPD-Net is too simple, resulting in higher sensitivity to data changes and thus weaker robustness compared to the best-performing baseline model. We also

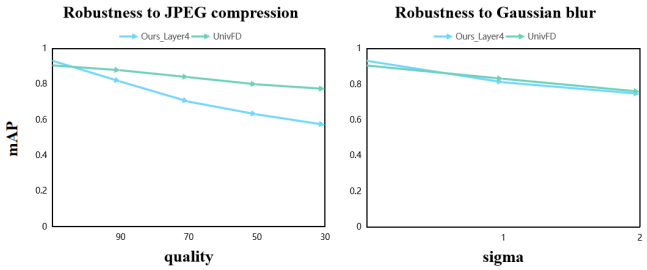


Fig. 4. Limitations analysis robustness analysis of different image post-processing operations.

analyzed the scalability issue. IPD-Net uses the self-attention calculation method proposed in [16]. To compare with the baseline models, the size of the input is only 256×256, and the training time per epoch is about 10-30 minutes slower than directly using ResNet50. However, because IPD-Net does not need to actively select specific patches, it is significantly faster than [14], [35]. If the input size increases, the calculation time of IPD-Net will significantly increase. Assume that when the height and width of the feature map to be multiplied both become n times larger, the number of dot products becomes n^4, while the number of channels remains the same. Therefore, optimizing the computation scheme for modeling the dependencies between patches is a problem for IPD-Net. For example, [36] proposed the Asymmetric Non-local Neural Network to improve Non-local Net. Therefore, reducing the number of steps in matrix multiplication, such as through dimensionality reduction or sampling before matrix multiplication, could potentially improve efficiency.

V. Conclusion

In this paper, we propose IPD-Net based on the existing inference that there is an inconsistency in the inter-pixels relation between the rich texture region and the poor texture region of AI-generated images. Firstly, we use a self-attention computation method and design a classification layer adapted to classification tasks, aiming to capture the interdependencies between patches of input images processed by the SRM filter. This enables the model to avoid the huge overhead caused by actively selecting specific patches and self-learn the common features of AI-generated images, thus improving computational efficiency. Secondly, we conduct extensive experiments on a test dataset containing 18 generative models, and the results show that our IPD-Net has high accuracy and good generalization ability. Thirdly, we conduct a comparison with several recent methods. IPD-Net outperforms the baseline models on multiple metrics. Regarding the future improvement of IPD-Net, we will focus on improving its network structure, especially the method of calculating the dependencies between patches to enhance its scalability and robustness. We hope that our work can provide some reference for future research.

Acknowledgment

This work was supported by Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011740), and Fundamental Research Funds for the Central Universities (Nos. 21624404, 23JNSYS01).

References

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, pp. 1–14, 2022.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, p. 139–144, 2020.

[3] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the Neural Information Processing Systems*, 2020, pp. 6840–6851.

[4] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Proceedings of the Neural Information Processing Systems*, 2021, pp. 8780–8794.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[6] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8695–8704.

[7] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 480–24 489.

[8] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *Proceedings of the IEEE International Workshop on Information Forensics and Security*, 2019, pp. 1–6.

[9] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proceedings of the International Conference on Machine Learning*, 2020, pp. 3247–3258.

[10] Y. Ju, S. Jia, L. Ke, H. Xue, K. Nagano, and S. Lyu, "Fusing global and local features for generalized ai-synthesized image detection," in *Proceedings of the International Conference on Image Processing*, 2022, pp. 3465–3469.

[11] M. Zhu, H. Chen, Q. YAN, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang, "Genimage: A million-scale benchmark for detecting ai-generated image," in *Proceedings of the Neural Information Processing Systems*, 2023, pp. 77 771–77 782.

[12] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 317–16 326.

[13] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[14] N. Zhong, Y. Xu, S. Li, Z. Qian, and X. Zhang, "Patchcraft: Exploring texture patch for efficient ai-generated image detection," *arXiv:2311.12397*, pp. 1–18, 2024.

[15] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[16] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[17] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv:1506.03365*, pp. 1–9, 2016.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proceedings of the International Conference on Learning Representations*, 2018, pp. 1–26.

[19] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[20] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv:1809.11096*, pp. 1–35, 2019.

[21] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[22] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[23] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[24] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1511–1520.

[25] K. Li, T. Zhang, and J. Malik, "Diverse image synthesis from semantic layouts via conditional imle," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4220–4229.

[26] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074.

[27] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3291–3300.

[28] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[31] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv:2112.10741*, pp. 1–20, 2022.

[32] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 696–10 706.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the Neural Information Processing Systems*, 2019, pp. 8024–8035.

[34] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 445–22 455.

[35] J. Chen, J. Yao, and L. Niu, "A single simple patch is all you need for ai-generated image detection," *arXiv:2402.01123*, pp. 1–10, 2024.

[36] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 593–602.