CGFormer: ViT-Based Network for Identifying Computer-Generated Images With Token Labeling

Weize Quan, Pengfei Deng, Kai Wang, and Dong-Ming Yan, *Member, IEEE*

***Abstract*— The advanced graphics rendering techniques and image generation algorithms significantly improve the visual quality of computer-generated (CG) images, and this makes it more challenging to distinguish between CG images and natural images (NIs) for a forensic detector. For the identification of CG images, human beings often need to inspect and evaluate the entire image and its local region as well. In addition, we observe that the distributions of both near and far patch-wise correlation have differences between CG images and NIs. Current mainstream methods adopt the CNN-based architecture with the classical cross entropy loss, however, there are several limitations: 1) the weakness of long-distance relationship modeling of image content due to the local receptive field of CNN; 2) the pixel sensitivity due to the convolutional computation; 3) the insufficient supervision due to the training loss on the whole image. In this paper, we propose a novel vision transformer (ViT)-based network with token labeling for CG image identification. Our network, called CGFormer, consists of patch embedding, feature modeling, and token prediction. We apply patch embedding to sequence the input image and weaken the pixel sensitivity. Stacked multi-head attention-based transformer blocks are utilized to model the patch-wise relationship and introduce a certain level of adaptability. Besides the conventional classification loss on class token of the whole image, we additionally introduce a soft cross entropy loss on patch tokens to comprehensively exploit the supervision information from local patches. Extensive experiments demonstrate that our method achieves the state-of-the-art forensic performance on six publicly available datasets in terms of classification accuracy, generalization, and robustness. Code is available at https://github.com/feipiefei/CGFormer.**

Manuscript received 13 October 2022; revised 18 July 2023; accepted 18 September 2023. Date of publication 4 October 2023; date of current version 20 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62102418 and Grant 62172415; in part by the Beijing Science and Technology Plan Project under Grant Z231100005923033; in part by the French National Research Agency under Grant ANR-15-IDEX-02, CDTools CyberAlps; and in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant 2022B1212010005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Naser Damer. *(Weize Quan and Pengfei Deng are co-first authors.) (Corresponding author: Dong-Ming Yan.)*

Weize Quan is with the MAIS and NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518055, China (e-mail: qweizework@gmail.com).

Pengfei Deng and Dong-Ming Yan are with the MAIS and NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: dengpengfei2021@ia.ac.cn; yandongming@gmail.com).

Kai Wang is with the CNRS, Grenoble INP, GIPSA-Laboratory, Université Grenoble Alpes, 38000 Grenoble, France (e-mail: kai.wang@gipsa-lab.grenoble-inp.fr).

Digital Object Identifier 10.1109/TIFS.2023.3322083

***Index Terms*— CG image forensics, transformer, token labeling, generalization, robustness.**

I. Introduction

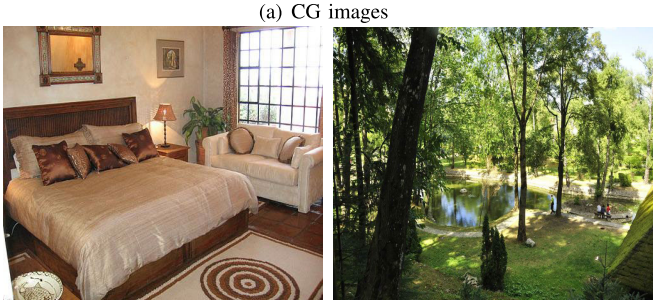
IMAGE has become a prevalent and important medium of communication due to simple acquisition process and rich content. Natural image (NI) captured by digital camera is historically reliable, *i.e.*, "seeing is beliving". Unfortunately, this reliability is threatened by the computer-generated (CG) image produced by advanced computer graphics rendering techniques and image generation algorithms. These CG images often have similar visual realism as the natural images (see Fig. 1(a) for example). Consequently, what you see is no longer trustworthy, and distinguishing CG images from NIs (*i.e.*, CG image forensic problem [1]) is a practical and valuable task in the image forensics community.

In the literature, many efforts have been made to solve this problem. Early work mainly relied on feature engineering [3] to design discriminative features, such as, geometry-based features [4] and wavelet statistic-based features [5]. These hand-crafted features are then fed to a classifier, *e.g.*, LDA (linear discrimination analysis), nonlinear SVM (support vector machine), ensemble classifier, etc. The feature design heavily depends on the human prior knowledge and some (shallow-level) observations about the data, therefore, the hand-crafted features often have a limited discrimination capability. Correspondingly, this hampers the identification performance of hand-crafted-feature-based forensic methods.

Inspired by significant performance improvement of convolutional neural works (CNNs) on image classification, where CNN combines the hierarchical feature extraction and classifier as a whole, recent research works [2], [6], [7], [8] pay more attention to the design of CNN models and achieve better forensic performance compared to traditional methods. However, these CNN-based approaches still have limitations in two aspects, *i.e.*, generalization and robustness, which are very important factors for a forensic detector working in real-world scenarios. The generalization refers to the performance of forensic models trained on "known" data and tested on "unknown" data. The robustness means achieving stable classification performance when the test data are subject to post-processing operations (either unintentional or malicious), *e.g.*, rescaling, JPEG compression, and noises.

Through carefully analyzing the existing CNN-based methods, we think that there are three reasons related to the

(a) CG images



(b) NIs

Fig. 1. CG images (a) and NIs (b). CG images are generated by the rendering tool VRay; NIs are from LSCGB dataset [2] and Corona dataset [1], respectively.

limitations mentioned in the last paragraph. The first one is the convolution computation process in CNNs. The common convolutional operation is element-wise product between pixels in a convolutional window and weights in a convolutional kernel, followed by a summation. This process fully depends on the pixels in local windows and thus is relatively sensitive to these pixel values and corresponding distributions. As a result, the trained models are possibly vulnerable to changes in the values of local pixels and their distributions, *e.g.*, changes introduced by post-processing operations such as rescaling and JPEG compression. In addition, the weights of learned convolutional kernels are fixed and applied to each input image. Therefore, CNNs to some extent have limited adaptability to the content of a single image and this may hamper the model's generalization. The second reason is the limited capability of modeling long-distance relationships of image content. CNNs progressively enlarge the receptive field with stacked convolutional layers (with dilations) and pooling operations so as to access larger regions of input images. The receptive field refers to the region of input image that is path-connected to a neuron in a certain layer [9]. However, simple stacks of multiple convolutional layers cannot effectively establish the relationships between long-distance image content, while such relationships contain important traces for CG image forensics as described in Section III-A. The third reason is that current CNN-based methods omit the importance of local image patches for the training objective. A local patch from NI/CG image has same label as the entire NI/CG image. However, existing methods mainly apply the conventional cross entropy loss over the whole image to train the model, in a similar way as the common image recognition task in computer vision community [10], [11]. Consequently, the CNN-based forensic detectors receive limited guidance to explore the relationships between the local patches and the global image. Inspired by these insights, we propose a vision transformer (ViT)-based network with simple and effective token labeling to significantly improve the generalization and robustness for CG image forensics.

Our work provides the following contributions:

- We introduce a ViT-based framework, named as CGFormer, for the CG image forensic problem. Compared to CNN-based method, this framework can lessen network's reliance on pixel values and corresponding distributions in a local window, and model the relationships of both near and far image patches. Therefore, our framework is able to achieve better generalization and robustness.
- We propose an effective token labeling method to optimize the training process. Besides the classification loss on the whole image, we add a weak constraint on the local patches, so that each patch token of our proposed CGFormer can capture more local information and the local-global relationships. This weak constraint can remarkably improve the forensic performance.
- We conduct a comprehensive study on CNN-based and ViT-based methods for the CG image forensic problem, including architecture analysis, experimental comparisons, visualization, and understanding. This provides some insights and inspires follow-up studies.
- Compared to CNN-based approaches, our ViT-based framework achieves the state-of-the-art performance on six common datasets for CG image forensics: LSCGB [2], SPL2018 [12], Autodesk [1], Artlantis [1], Corona [1], and VRay [1].

The rest of this paper is organized as follows. Section II reviews the existing works. Section III describes the motivation and technical details of the proposed method. In Section IV we present the experimental results of our method, compare it with existing state-of-the-art methods, and make efforts to understand the results with visualization tools. Section V concludes this paper and discusses directions for future work.

II. Related Work

This section briefly reviews the existing CG image forensic approaches, including computer graphics-generated image detection and generative models-generated image detection. We refer readers to recent surveys [13], [14] for more details and information about the relevant research.

A. Computer Graphics-Generated Image Detection

1) Traditional Methods: Traditional methods follow a two-stage machine learning framework consisting of feature extraction and classifier design. The previous works extract the discriminative features in spatial domain and transform domain.

a) Spatial domain: This kind of methods usually extracts the statistical differences, texture details, and geometric information as the discriminative features. Based on the fractal and differential geometry, Ng et al. [4] designed geometry-related features to identify CG images. Pan et al. [15] used the global color distribution and the texture difference as the cues

for CG image forensics. Sankar et al. [16] proposed hybrid features combining the periodic correlation, color histogram, moment and local patch statistics. More aspects are also considered, such as local image edge statistics [17], gray-scale invariant local binary patterns [18], PRNU (photo response non-uniformity noise) [19], multi-fractal spectrum [20], etc.

b) Transform domain: Through transforming the image from spatial domain to frequency domain, some hidden forensic traces are exposed to construct the feature vectors. Lyu and Farid [5], [21] combined the wavelet statistics of first four orders (mean, variance, skewness, and kurtosis) as the feature to discriminate between NIs and CG images. Chen et al. [22] conducted the discrete wavelet transform in the HSV (hue, saturation, lightness) color space, and then extracted discriminative features from the discrete Fourier transform of the wavelet coefficients. Özparlak et al. [23] extracted features from the contourlet and ridgelet transform of input image. Considering some drawbacks existing in discrete and contourlet wavelet transform, later, Wang et al. [24] proposed a quaternion wavelet transform-based feature extraction method.

2) CNN-Based Methods: The basic idea of CNN-based methods is to directly learn the mapping function from input image to output label. Technically, the automatic feature extraction and classifier are combined as a whole, which can be trained in an end-to-end manner. Based on the traditional CG image forensic model, Rahmouni et al. [25] combined convolutional layers, a specific pooling layer, and a multi-layer perceptron to construct a deep model for distinguishing NIs from CG images. By analyzing the difference between conventional image classification and CG image forensic problem, Quan et al. [6] improved CNN's forensic performance with a learnable preliminary filtering layer. They also understood the trained model with several advanced visualization tools. To remove low-frequency signals and enhance the sensor pattern noise, Yao et al. [26] used three sets of high-pass filters in the front of CNN model. He et al. [12] proposed a two-stream CNN to exploit the color and texture characteristics and then combined a RNN (recurrent neural network) to enhance the discriminative capability of forensic features. Nguyen et al. [27] modified the architecture and training method of the capsule network [28] to better model image spatial information. Instead of the uniform processing for entire input image, Bhalang et al. [29] proposed an attention-guided recursive model to progressively process the local image area. Zhang et al. [30] identified the CG images by jointly exploiting the channel and pixel correlation with a hybrid correlation module, which can also be used in other CNN models. Besides the classification accuracy, Quan et al. [1] paid more attention to the blind detection (or generalization) problem, and proposed a two-branch network with a negative-sample-based enhanced training method. He et al. [7] introduced a dual-branch CNN with an attention-based fusion model for the identification of CG images. Their network takes as input stacked original input image and its Gaussian blurred version. Very recently, Bai et al. [2] constructed a large-scale CG images benchmark (LSCGB), and proposed a texture-aware network to explore the texture difference between NIs and CG images. Yao et al. [8] designed a CG image detection network with a feature transfer module and an attention-guided fusion module, considering both the shallow content and deep semantic features.

B. Generative Models-Generated Image Detection

1) Spatial Domain: Wang et al. [31] demonstrated that a common image classifier (*i.e.*, ResNet-50 [11]), which is trained on a specific CNN generator with two operations (*i.e.*, Gaussian blur and JPEG compression), can detect fake images generated by unseen CNN-based generative models. Further, Chandrasegaran et al. [32] discovered the transferable forensic features in "universal" detectors [31] by introducing forensic feature relevance statistics. Chai et al. [33] designed a CNN detector truncated from the Xception backbone [34] with small receptive fields to focus on the local artifacts. Cozzolino et al. [35] introduced a ForensicTransfer model to improve the generalization performance. Specifically, this model is an autoencoder-based network that jointly performs embedding learning and forensic detection. There are some works about face forgery detection. Li et al. [36] detected face forgery by exposing the blending boundary. To train the detector, they introduced a training sample generation method that blends two real images. Kim et al. [37] proposed a transfer learning-based feature representation learning method to train a student model on new deepfake datasets and achieved good performance on the domain adaptation task. Shiohara and Yamasaki [38] also designed a method based on synthetic training data, namely, self-blended images. Different from [36], they generated the fake image by blending the source image and its slightly changed version.

2) Frequency Domain: Zhang et al. [39] proposed a spectrum-based classifier to identify GAN-generated images. For an RGB image, they first extracted the 3 channels of the frequency spectrum with 2D DFT (discrete Fourier transform), then normalized the spectrum as the classifier's input. Frank et al. [40] applied the discrete cosine transform (DCT) to transform input image into the frequency domain, and then carried out the identification of GAN fake images. Dzanic et al. [41] observed that there were noticeable differences between real and deep network-generated images in their high-frequency spectra, and proposed a detection method based on these characteristics. Durall et al. [42] had a similar observation and proposed a spectral regularization term, which was inserted into the GAN training objective to compensate for observed spectral distortions. Recently, Chandrasegaran et al. [43] revisited the claim that CNN-generated images shared high-frequency spectral decay attributes as reported in previous studies [41], [42]. They observed that these discrepancies could be avoided by modifying the last feature map scaling method. Similarly, Dong et al. [44] proposed a pipeline to mitigate spectra artifacts and correct the power discrepancy of GAN-based images. As a result, the spectrum-based detectors [40], [42] have a noticeable performance drop. These recent researches imply that spectral(frequency)-domain-based detectors can be vulnerable to mitigation methods; by contrast, the spatial-domain-based methods may have better robustness.

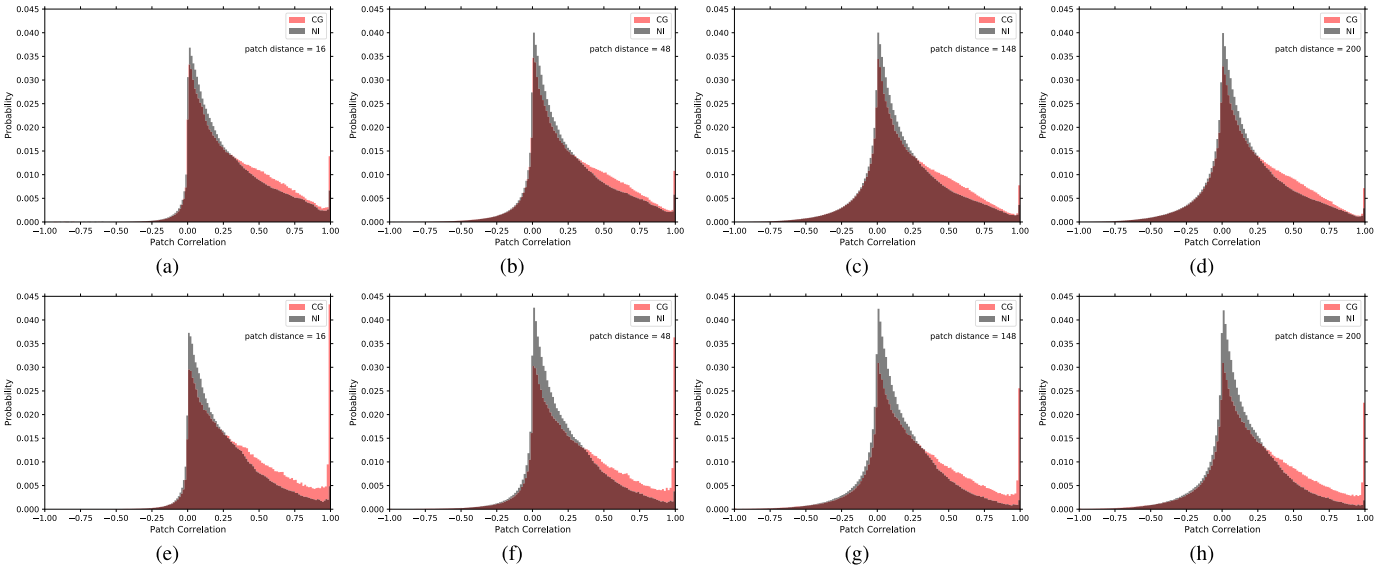


Fig. 2. The distribution of correlation of patch pairs with different distances: 16, 48, 148, and 200. In (a-d) are results calculated on LSCGB [2], and in (e-h) are results calculated on SPL2018 [12].

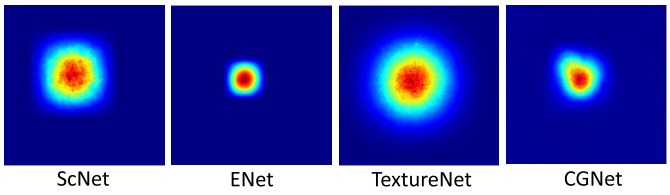


Fig. 3. Illustration of effective receptive field of state-of-the-art CNN-based CG image forensic detectors.

III. Proposed Method

A. Observation & Motivation

For the CG image forensic problem, we have several observations and have carried out some analyses about the data, the existing CNN-based methods, and human observers.

1) Correlation Analysis: The images generated by computers, based on advanced rendering techniques or the recent deep generative models, are all simulating the distribution of pixels in natural images. Due to various influence factors, *e.g.*, illumination, viewing angle, shadow, occlusion, and camera settings, the distribution of pixels is complex and diverse for natural images. Different rendering algorithms and trained deep generative neural networks may cover part of the distribution space of natural images. In this work, we use the probability distribution of the correlation of image patches with different distances as the statistic measure to illustrate the differences between NIs and CG images.

We analyze this statistical distribution on two datasets: LSCGB [2] and SPL2018 [12]. Specifically, we randomly select N natural images and N CG images from one dataset, where $N = 10,000$ for LSCGB and $N = 3,000$ for SPL2018. For each image, we randomly crop 1,000 pairs of 16×16 patches with four different distances, *i.e.*, 16, 48, 148, and 200, along the horizontal direction. To compute the patch correlation, we first rescale the pixel value to $[-1, 1]$, and then flatten each color patch of $16 \times 16 \times 3$ into a 768-dimensional vector. The patch correlation is obtained via the inner product of two vectors representing a pair of patches, and the correlation value is divided by 768 to fall into $[-1, 1]$. Fig. 2 shows the distribution of patch correlation for four different distance settings. It is observed that the empirical distributions of patch correlation exhibit notable difference between NI and CG patches for *all distances*. Specifically, more patch pairs of CG images fall into the strong correlation region, for both close patch pairs and distant patch pairs. The reason might be that computer generation algorithms have potentially certain specific modes and rules which introduce some kind of repetitive patterns to CG images, while NIs have relatively more randomness.

2) Effective Receptive Field: For the CG image forensic problem, CNN-based models are composed of multiple convolutional layers and down-sampling layers. For each convolutional layer, convolution operation is conducted via convolutional kernels with fixed and small sizes (often 3×3 or 5×5) in a sliding window manner. Although the stacked arrangement of convolutional layers can increase the receptive field, the capability of modeling long-distance relationship is limited [45] and the effective receptive field only occupies part of the entire theoretical receptive field [46]. For several advanced CNN-based CG image forensic methods, *i.e.*, ScNet [30], ENet [1], TextureNet [2], and CGNet [8], we visualize the effective receptive field of a neuron in the final feature extraction layers using the popular analysis tool proposed in [46], and the results are shown in Fig. 3. We can see that the effective receptive field of these models for a feature neuron only covers a fraction of input image. Therefore, the discriminative features extracted by CNNs have a certain level of locality.

3) Human Observers: To identify the authenticity of a given image, *i.e.*, to answer the question whether it is an NI or a CG image, human observers often need to analyze the local region and the entire image, and the relationships between them as well. Then, they combine all these pieces of information to deduce a final result according to some "prior knowledge".

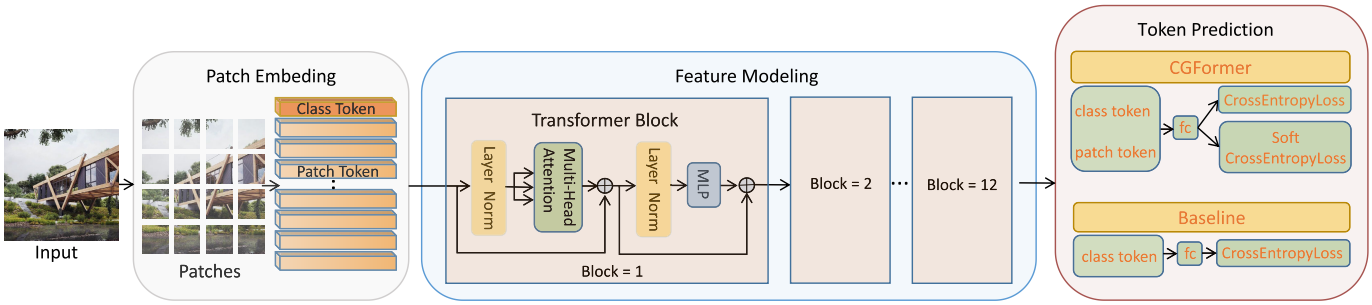


Fig. 4. The whole architecture of our CGFormer. Patch embedding contains the class token and the patch tokens. Feature modeling consists of 12 stacked transformer blocks. Token prediction efficiently exploits the supervision information by adding cross entropy loss on class token and soft cross entropy loss on patch tokens.

In addition, human observers have certain ability to correctly recognize the images that are "unseen" during their learning stage (*i.e.*, images that are somewhat different from those seen during learning). This "generalization" ability would be related to the "adaptability" of their analysis process, especially the analysis of relationships within a single image as described above.

B. Our Framework

Inspired by the above observations and analyses, we intend to design a novel CG image forensic network, which can model the patch-wise relationship with near and far distances and possess the good "adaptability" capability. Therefore, we propose a ViT-based network with token labeling, named as CGFormer. We adopt the transformer architecture to model the patch relationships and the dynamic weight strategy hidden behind the self-attention computation to enhance the "adaptability" capability of network. *Dynamic weights* refer to the so-called connection weights that are adaptively learned for each image instance [47] (details are presented later in this subsection), whereas the convolutional kernels are fixed for all image instances. Compared to the simple inner product of image space in Sec. III-A.1, multiple self-attention-based blocks have the potential to model complex relationships between patch pairs, which can enlarge the representation differences between NIs and CG images. Moreover, we introduce the token labeling to guide the network to extract more traces from local patches and enhance the relationship modeling of local patches and the global image.

Our CGFormer takes an image $\mathbf{I} \in \mathbb{R}^{224\times224\times3}$ as input and outputs a binary label y, where 0 means CG image and 1 means NI. As illustrated in Fig. 4, the whole network consists of three parts: patch embedding, feature modeling, and token prediction.

1) Patch Embedding: This part reorganizes the input image as sequential tokens, which are then used for learning the patch-wise relationship in the consequent feature modeling stage. Image $\mathbf{I}$ is first split into 16×16 patches without overlapping, and the total number of patches is M. For each patch $\mathbf{p}$, it is flattened into a vector and transformed into embedded vector via a linear projection. It is called as patch token $T_p^m \in \mathbb{R}^d$, where d is the feature dimension of patch token, and m is the patch token index. For the image recognition, a class token $T_c \in \mathbb{R}^d$ with learnable parameters is often used to collect the discriminative information for the final classification. Following the original ViT [48], we adopt a learnable positional encoding mechanism, which introduces a parametric matrix $\text{pos} \in \mathbb{R}^{M\times d}$ describing positional information of image patches. After patch embedding, we can obtain sequential tokens $T \in \mathbb{R}^{(M+1)\times d}$. The detailed formulation is

$$T = \text{cat}(T_c, T_p + \text{pos}), \tag{1}$$

where cat is the concatenation of patch tokens T_p and class token T_c along the token dimension.

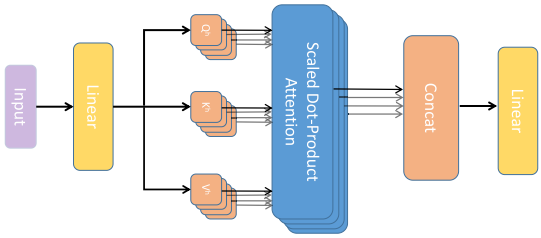


Fig. 5. Illustration of the multi-head attention module.

2) Feature Modeling: CNN-based methods extract the hierarchical features through stacked convolutional layers and down-sampling operations. In contrast, our method applies multiple transformer blocks to extract discriminative features. As shown in the middle of Fig. 4, the transformer block is based on skip connection and multi-head attention [49]. The transformer blocks take $T \in \mathbb{R}^{(M+1)\times d}$ as input and output attentive tokens $\hat{T} \in \mathbb{R}^{(M+1)\times d}$. In our work, there are B stacked transformer blocks and we take an transformer block as example to explain the practical computation process. This can be formulated as follows:

$$\begin{aligned} \tilde{T} &= \text{MHA}(\lambda(T)) + T, \\ \hat{T} &= \text{MLP}(\lambda(\tilde{T})) + \tilde{T}, \end{aligned} \tag{2}$$

where λ is the layer normalization [50], MHA represents the multi-head attention module, and MLP means a multilayer perceptron layer.

Fig. 5 shows the architecture of multi-head attention module. The input $T \in \mathbb{R}^{(M+1)\times d}$ of MHA is first processed by layer normalization and then projected into query $Q \in \mathbb{R}^{(M+1)\times d}$, key $K \in \mathbb{R}^{(M+1)\times d}$, and value $V \in \mathbb{R}^{(M+1)\times d}$,

respectively. It is written as,

$$Q = \lambda(T)\mathbf{W}^Q,\ K = \lambda(T)\mathbf{W}^K,\ V = \lambda(T)\mathbf{W}^V, \tag{3}$$

where $\mathbf{W}^Q \in \mathbb{R}^{d\times d}$, $\mathbf{W}^K \in \mathbb{R}^{d\times d}$, and $\mathbf{W}^V \in \mathbb{R}^{d\times d}$ are the linear projection matrices. Then, Q, K, and V are split into multiple parallel heads $Q^h \in \mathbb{R}^{(M+1)\times d_h}$, $K^h \in \mathbb{R}^{(M+1)\times d_h}$, and $V^h \in \mathbb{R}^{(M+1)\times d_h}$, where $d_h = \frac{d}{H}$ and H is the number of heads. For h-th head, the attentive feature is calculated via the scaled dot-product attention as

$$f_h = \text{Softmax}\left(\frac{Q^h\left(K^h\right)^{\mathrm{T}}}{\sqrt{d_h}}\right)V^h. \tag{4}$$

Through this equation and as mentioned earlier in this subsection, the connection weights (*i.e.*, the Softmax part in the above equation) are dynamically predicted for each image instance according to the queries and the keys with scaled dot product and softmax normalization. Output $T' \in \mathbb{R}^{(M+1)\times d}$ of MHA is obtained by concatenating these parallel attention heads and linear projection:

$$T' = \text{cat}(f_1, f_2, \ldots, f_H)\mathbf{W}', \tag{5}$$

where cat is the concatenation operator and $\mathbf{W}' \in \mathbb{R}^{d\times d}$ is the learnable projection matrix.

3) Token Prediction: To comprehensively exploit the supervision information from the global image and local patches, we propose a simple and effective token labeling method. It adds the constraints on the features of class token and patch tokens simultaneously. Let F_c denote the feature of class token, and F_p^m denote the feature of m-th patch token. A shared linear classifier FC with two output neurons transforms F_c and F_p^m to the logits G_c and G_p^m, respectively.

For the class token, we use the classical cross entropy loss, and it is formulated as:

$$\mathcal{L}_c = -\sum_{l=1}^{L}\log\frac{\exp(G_c^l)}{\sum_{l=1}^{L}\exp(G_c^l)}\cdot 1\{y=l\}, \tag{6}$$

where $1\{\cdot\}$ is the indicator function (*i.e.*, $1\{\text{False}\} = 0$ and $1\{\text{True}\} = 1$), and L is the number of classes ($L = 2$ in our work representing two classes of NIs and CG images).

Different from the hard constraint in Eqn. (6), where for an NI training sample the prediction score (Pr) must be close to 1 (or respectively close to 0 for a CG training sample), we add a soft constraint on patch token only requiring that the corresponding prediction score is bigger than (or smaller than) 0.5 for NI (or CG). In other words, we do not penalize training patches when an NI (or a CG) training patch is predicted as NI (or CG) with $\text{Pr} > 0.5$ (or $\text{Pr} < 0.5$). This design is reasonable because the forensic information hidden behind a small patch (*i.e.*, 16×16) sometimes may be not enough to produce the right prediction with very high confidence. The detailed formulation of loss function for the patch token is written as:

$$\mathcal{L}_p = -\frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{L}\log\phi\left(\frac{\exp(G_p^{ml})}{\sum_{l=1}^{L}\exp(G_p^{ml})}\right)\cdot 1\{y=l\}, \tag{7}$$

TABLE I

CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON LSCGB DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	*95.88*	66.87	73.75	71.12	66.12	66.48
ENet	*93.89*	73.69	77.23	71.81	78.62	69.17
TextureNet	*95.87*	89.41	85.42	84.48	83.75	78.89
CGNet	*94.89*	85.42	81.59	79.03	80.25	74.75
Ours	*98.00*	97.13	95.56	94.87	92.23	90.98

TABLE II

CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON SPL2018 DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	59.41	*97.50*	66.12	63.20	64.17	62.23
ENet	57.78	*97.25*	66.53	64.31	66.53	62.09
TextureNet	62.53	*98.22*	72.64	73.62	75.84	71.12
CGNet	66.08	*98.04*	78.89	77.23	79.45	75.84
Ours	79.53	*99.16*	90.70	88.62	88.48	86.67

TABLE III

CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON AUTODESK DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	55.85	48.13	*98.62*	91.98	91.81	86.95
ENet	52.39	52.79	*98.75*	89.04	92.92	88.23
TextureNet	62.73	76.47	*98.48*	88.75	83.75	84.17
CGNet	63.54	76.50	*98.62*	95.65	91.81	91.95
Ours	65.44	76.72	*99.03*	95.76	84.76	85.14

where $\phi(\cdot)$ is a thresholding function

$$\text{Pr} = \phi(\text{Pr}) = \begin{cases} 1-\epsilon, & \text{Pr} > 0.5; \\ \epsilon, & \text{Pr} < 0.5. \end{cases} \tag{8}$$

where ϵ is a small factor to prevent numerical instability.

The final training objective of CGFormer is the average of class token loss and patch token loss:

$$\mathcal{L} = \frac{\mathcal{L}_c + \mathcal{L}_p}{2}. \tag{9}$$

In [51], Jiang et al. used a token labeling technique to train better vision transformers for image recognition task. They applied a pre-trained image recognition model to generate a dense score map as the auxiliary objective on patch token. There exists two apparent differences between our token labeling and [51]: (1) For our method, the supervision information on patch tokens directly come from the training data, whereas [51] needs a so-called machine annotator introducing additional training cost and the performance of this annotator also affects the final results of ViTs; (2) Instead of directly using cross entropy loss as in [51], we additionally introduce a relaxation after softmax, *i.e.*, Eqn. (8).

IV. EXPERIMENTAL RESULTS

In this section, we first describe the experimental settings, including the datasets and implementation details. Then, we evaluate our method through comparison experiments and robustness analysis, and analyze our method via ablation studies. Finally, we make efforts to understand the working mechanism of our CGFormer with some visualizations.

TABLE IV
CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON ARTLANTIS DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	52.75	54.00	78.37	*98.89*	74.87	81.81
ENet	47.95	50.35	88.50	*98.89*	85.64	86.56
TextureNet	56.92	83.32	93.62	*98.75*	85.28	90.56
CGNet	54.15	77.72	95.25	*98.25*	87.37	85.98
Ours	67.20	87.29	97.78	*99.03*	87.95	91.81

TABLE V
CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON CORONA DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	53.45	68.87	92.21	85.14	*98.20*	91.53
ENet	55.44	52.04	92.37	83.06	*98.48*	91.95
TextureNet	54.44	70.75	89.03	82.64	*98.34*	92.50
CGNet	61.06	80.98	92.09	87.64	*98.12*	95.28
Ours	68.94	84.21	92.48	87.89	*99.16*	96.08

TABLE VI
CLASSIFICATION PERFORMANCE OF FIVE NETWORKS WHEN TRAINED ON VRAY DATASET

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
ScNet	52.25	71.29	89.87	90.16	94.73	*97.78*
ENet	56.78	64.96	90.16	87.72	95.79	*98.59*
TextureNet	53.14	78.60	83.75	85.14	94.87	*98.48*
CGNet	53.55	67.67	87.50	90.09	94.66	*98.62*
Ours	70.88	85.66	90.23	87.89	96.81	*99.03*

A. Experimental Setup

1) Datasets: In this work, we compare and evaluate our method on six publicly available datasets: LSCGB [2], SPL2018 [12], Artlantis [1], Autodesk [1], Corona [1], and VRay [1]. In the following, we briefly describe these datsets.

LSCGB is a large-scale CG image forensics dataset, which contains 71,168 NIs and 71,168 CG images. NIs are collected from existing CG forensics datasets (Columbia image dataset [52] and Tokuda dataset [53]), public NI datasets in computer vision (PASCAL VOC [54] and COCO [55]), some movies without special effect, and several photo websites. CG images mainly come from existing CG forensics datasets (Columbia image dataset [52] and Tokuda dataset [53]), 3D games and movies, cartoon, and generative adversarial networks (*e.g.*, StyleGAN [56], BigGAN [57], etc). For data split, we follow the original setting shared by [2]. In particular, the training set consists of 49,823 CG images and 49,813 NIs, and the test set contains 14,230 CG images and 14,233 NIs.

SPL2018 dataset comprises 6,800 CG images and 6,800 NIs, where CG images are created using over 50 graphics rendering software tools (*e.g.*, Maya, 3D Max, etc) and NIs are captured with several digital cameras. Following the original data split, 4,000 CG images and 4,000 NIs are used for training, and 1,600 CG images and 1,600 NIs for testing.

The remaining four datasets, *i.e.*, Autodesk, Artlantis, Corona, and VRay, are collected by Quan et al. [1]. NIs are from RAISE dataset [58] and VISION dataset [59], and CG images are produced by the above four rendering tools. For each dataset, the training set contains 5,040 NIs and 5,040 CG images, and the test set contains 360 NIs and 360 CG images.

2) Implementation Details: Our model is implemented with PyTorch 1.8.0. The GPU is an NVIDIA TITAN RTX. SGD optimizer with the momentum of 0.9 and the weight decay of 1e-4 is used to train our models. We set the batch size as 64. The learning rate is initialized to 0.003, and is adjusted using a cosine annealing schedule [60] with $T_{max} = 30$ and $\eta_{min} = 1e-7$. The training process stops after 150 epochs. The parameters of CGFormer is initialized using the weights of ViT pre-trained on ImageNet-21K [61] and fine-tuned on ImageNet-1K [62]. For the soft cross entropy loss on patch tokens (Eqn. (7)), we empirically set $\epsilon = 0.01$.

For all images, we resize the shorter edge as 512. In the training stage, we randomly crop a 224×224 subimage as the network input. In the testing stage, we obtain the prediction of a test image by averaging predictions of ten subimages (including subimages from the center and four corners, and their horizontally flipped version), for our method as well as for state-of-the-art methods to ensure fair comparisons. Following previous methods [2], [8], [30], we use the accuracy (in %) as the evaluation metric of forensic performance. In the following tables, the classification accuracy when training and testing data come from same dataset is in italic, while the performance when training and testing data are from different datesets (*i.e.*, generalization) is shown in normal font.

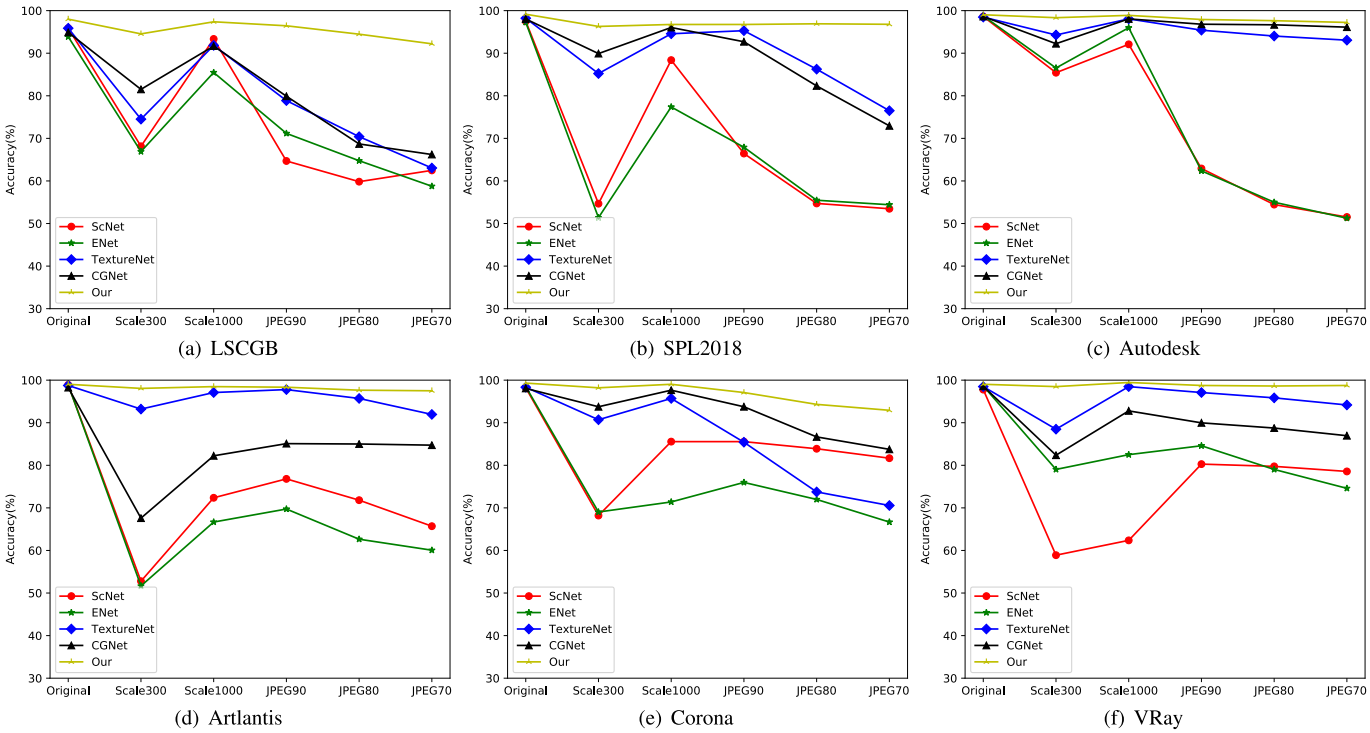
B. Comparisons With State-of-the-Art Methods

In this work, we compare our network with several advanced CG forensic methods, including ScNet [30], ENet [1], TextureNet [2], and CGNet [8].

Tables I-VI report the comparisons of five methods on six datasets in terms of conventional classification accuracy (*i.e.*, training and test data from same dataset) and generalization (*i.e.*, training and test data from different datasets). Specifically, in each table all models are trained on one dataset and then tested on all datasets. From these tables, we can see that our method achieves the superior overall performance. In terms of classification accuracy (in italics), our method surpasses the second best method by 2.12% (LSCGB), 0.94% (SPL2018), 0.28% (Autodesk), 0.14% (Artlantis), 0.68% (Corona), and 0.41% (VRay), respectively. Moreover, our method in many cases notably enhances the generalization capability of CG forensic detector. Taking the results in Table I as an example, we can see that our CGFormer outperforms the second best method TextureNet with significant gain of 7.72% (SPL2018), 10.14% (Autodesk), 10.39% (Artlantis), 8.48% (Corona), and 12.09% (VRay), respectively.

Next, we compare the robustness of all the above methods against several post-processing operations, including rescaling, JPEG compression, Gaussian noise, and salt and pepper noise.

Similar to [6], the rescaling operations contain down-scaling ("S300") and up-scaling ("S1000") with bilinear interpolation. The process is that we first resize the shorter edge of image as 300/1,000 pixels and then rescale it back to 512 pixels. For the JPEG compression, we consider three quality factors, including 90, 80, and 70. The corresponding results are shown in Fig. 6 and Table VII and VIII. The curves of our method are almost flat, which means that our CGFormer is very



(a) LSCGB (b) SPL2018 (c) Autodesk

(d) Artlantis (e) Corona (f) VRay

Fig. 6. Classification accuracies of the five methods on six datasets under rescaling and JPEG compression post-processing.

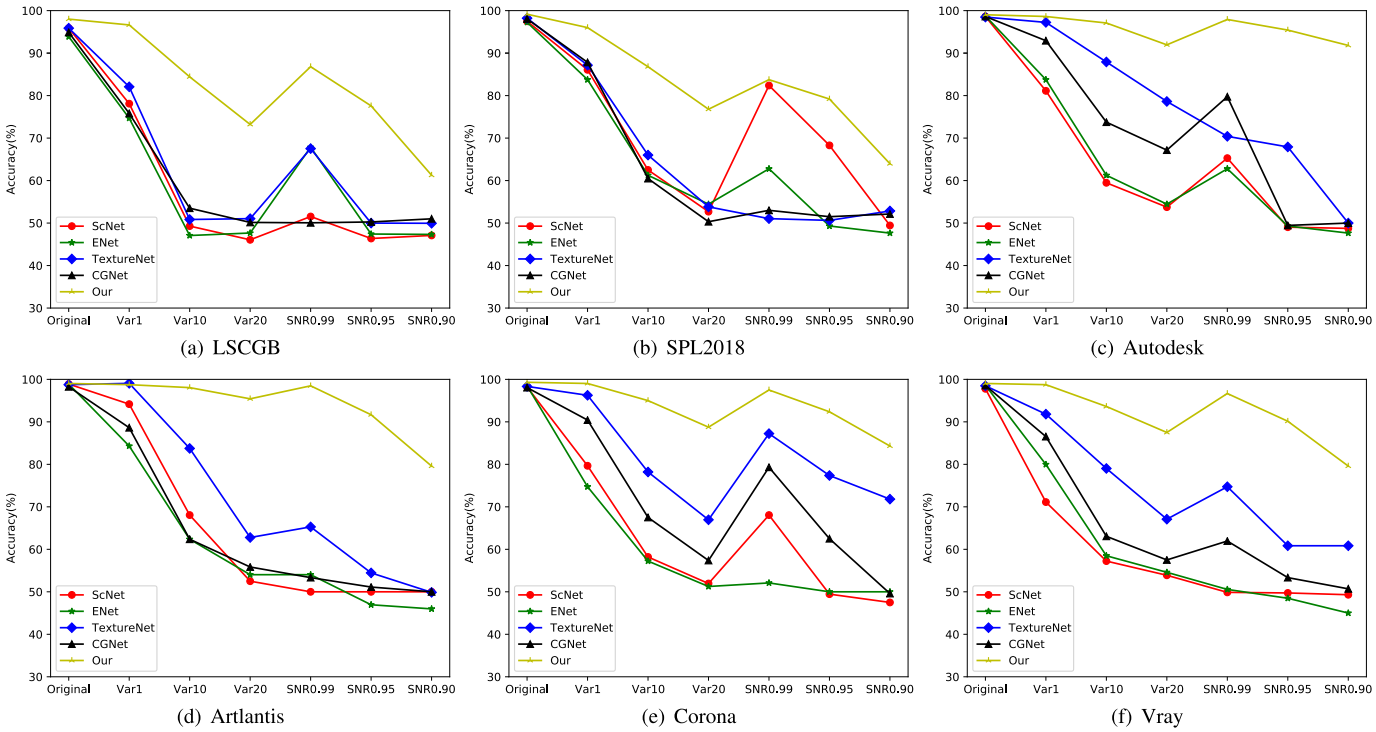
robust against rescaling and JPEG compression. For other CNN-based competitors, the performance drastically drops for down-scaling and decreases as the strength of JPEG compression increases. When the images are under rescaling and JPEG compression operation, the correlation of pixels with their local neighborhoods is potentially changed. CNN-based forensic detectors are mainly composed of multiple convolutional layers, where the convolutional operation is within a local window and might be sensitive to the change of local pixels correlation.

For the Gaussian noise, we add to the original pixel value a random noise with zero mean and variances of 1, 10, and 20, respectively. For the salt and pepper noise, the signal noise rate (SNR) is set as 0.99, 0.95, and 0.90, respectively. The corresponding results are shown in Fig. 7 and Table IX and X. We can see that the classification accuracies of CNN-based methods quickly decrease when the density of noise increases. In contrast, the performance of our method is relatively stable and apparently remains the best among all five methods. One possible reason is that our method mainly depends on the patch-wise relationship modeling, which is less vulnerable to the change of individual pixel values.

In addition, we also compare our method with several advanced methods designed for CNN-generated image detection, including SpecCNN [39], FSD [41], LFA [40], Xception2 [33], and CNNDet [31], where SpecCNN, FSD, and LFA are frequency domain-based methods. We train these models on LSCGB dataset and then test them on other datasets, including computer graphics-generated images (in Table XI) and AIGC (artificial intelligence generated content)-generated images (in Table XII). Following the setting in [31], all images are resized with a shorter edge of 256 pixels, and we apply center cropping in the testing stage. For AIGC-generated images, we evaluate 13 synthesis models of different categories: (1) unconditional GAN (ProGAN [63], StyleGAN [56], and BigGAN [57]); (2) conditional GAN (CycleGAN [64], StarGAN [65], and GauGAN [66]); (3) perceptual loss (CRN [67] and IMLE [68]); (4) low-level vision (SITD [69] and SAN [70]); (5) Deepfake (FaceForensics++ [71]); (6) diffusion model (stable diffusion [72] and midjourney[1]). The images generated by the first 11 CNN-based generation methods are shared by [31]. We collect two kinds of diffusion model-generated images, *i.e.*, SD (stable diffusion) and MJ (midjourney). For each diffusion model, we select 1,000 synthetic images with high realism and randomly select 1,000 real images from CNN_synth_testset [31].

From Table XI and XII, we find that our method in general achieves superior detection performance. Furthermore, we add five post-processing operations, *e.g.*, Scale160 (down-scaling), Scale500 (up-scaling), JPEG90, Var1, and SNR0.99, on 13 kinds of AIGC-generated images. For Scale160 and Scale500, we first rescale the shorter edge of the image as 160/500 pixels and then resize it back to 256 pixels. The corresponding detection results are reported in Table XIII-XVII. Among all competitive methods, our method demonstrates the best overall robustness. In addition, most detection models achieve rather limited performances when tested on AIGC-generated images. A possible reason is that there may exist differences between forensic traces on computer graphics-generated images (main focus of this paper and the dominant content of the LSCGB dataset representing about 90% of its computer-generated images) and traces on

[1]https://www.midjourney.org/



(a) LSCGB (b) SPL2018 (c) Autodesk

(d) Artlantis (e) Corona (f) Vray

Fig. 7. Classification accuracies of the five methods on six datasets under Gaussian noise and salt and pepper noise post-processing.

TABLE VII

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST RESCALING

Method	LSCGB		SPL2018		Autodesk		Artlantis		Corona		VRay	
	S300	S1000	S300	S1000	S300	S1000	S300	S1000	S300	S1000	S300	S1000
ScNet	68.13	93.55	54.57	88.25	85.10	91.81	52.71	72.37	68.14	85.04	59.23	62.34
ENet	68.04	83.89	50.84	77.63	84.89	90.45	52.45	66.87	68.23	70.31	78.45	81.81
TextureNet	75.44	91.83	85.92	93.69	94.50	98.77	93.50	98.49	90.31	95.45	88.23	99.03
CGNet	81.14	90.40	89.93	93.95	92.74	98.75	67.89	81.13	92.59	96.89	82.23	91.72
Ours	95.02	97.51	97.13	97.14	98.78	99.03	98.88	98.94	98.89	99.16	98.95	99.34

TABLE VIII

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST JPEG COMPRESSION

Method	LSCGB			SPL2018			Autodesk			Artlantis			Corona			VRay		
	J90	J80	J70	J90	J80	J70	J90	J80	J70	J90	J80	J70	J90	J80	J70	J90	J80	J70
ScNet	65.46	59.65	62.43	68.66	54.89	54.79	62.39	56.43	54.57	75.81	71.81	65.70	85.00	84.02	83.26	78.48	78.48	78.18
ENet	71.72	67.48	60.64	68.57	55.47	55.38	62.37	56.50	54.56	68.62	62.26	61.18	75.17	73.28	68.39	84.12	78.78	76.25
TextureNet	79.74	70.45	68.23	96.77	85.79	78.44	97.55	96.64	93.98	98.56	96.59	93.12	85.04	73.46	71.65	97.94	96.16	94.58
CGNet	79.88	69.86	66.61	91.94	82.13	73.35	98.42	98.34	98.12	84.95	84.96	84.95	94.51	88.13	85.14	89.95	88.98	87.97
Ours	96.48	94.65	92.28	97.14	97.14	97.14	98.62	98.58	98.34	98.62	98.42	98.34	97.78	95.64	93.76	98.48	98.62	98.34

AIGC-generated images. Future research efforts shall be devoted to investigating this interesting point.

C. Ablation Studies

1) Patch Size: Our CGFormer first conducts the patch embedding, which splits the original input image into many non-overlapped image patches. Here, we carry out ablation experiments to analyze the impact of different patch size on the forensic performance. Specifically, we evaluate the following three types of patch size: 8×8, 16×16, and 32×32. For each type, the model is trained and tested on all six datasets, and the numerical results are reported in Table XVIII. Among these three patch sizes, 16×16 achieves relatively superior performance. When patch size is 8×8, such a small patch contains less information for classification. For large patch size, *i.e.*, 32×32, the modeling of patch-wise relationship is somewhat coarse, which may then hamper the capability of the forensic detector. Therefore, we choose a patch size of 16×16 with the appropriate amount of information and relatively fine-grained patch relationship modeling.

2) Token Labeling: In this work, we introduce the token labeling to comprehensively exploit the supervision information of training data, *i.e.*, the ground-truth label on the entire image and local patches. As reported in Eqn. (7), we add a soft constraint on the prediction of each patch. Besides this

TABLE IX
THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST GAUSSIAN NOISE WITH THREE LEVELS OF VARIANCE

Method	LSCGB			SPL2018			Autodesk			Artlantis			Corona			VRay		
	V1	V10	V20	V1	V10	V20	V1	V10	V20	V1	V10	V20	V1	V10	V20	V1	V10	V20
ScNet	86.18	61.92	52.52	81.16	59.66	55.47	80.78	59.50	54.64	94.78	67.50	51.64	79.96	58.81	52.17	71.00	57.98	54.78
ENet	74.46	48.37	49.14	83.72	60.49	54.73	84.03	61.67	53.62	84.53	62.23	55.70	75.70	58.00	52.12	80.06	59.00	56.42
TextureNet	82.45	50.54	50.64	88.57	67.37	54.73	97.92	88.00	79.73	99.03	84.50	63.28	96.00	78.75	67.42	93.42	79.56	68.67
CGNet	75.17	52.99	50.12	88.16	60.22	51.22	93.65	74.23	68.54	88.95	61.39	56.84	90.21	67.23	58.31	87.64	63.09	59.06
Ours	96.41	84.56	72.39	96.10	87.29	77.79	98.75	96.84	92.44	99.03	98.09	94.88	98.98	95.03	89.51	98.84	93.56	89.21

TABLE X
THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST SALT AND PEPPER NOISE

Method	LSCGB			SPL2018			Autodesk			Artlantis			Corona			VRay		
	S0.99	S0.95	S0.90	S0.99	S0.95	S0.90	S0.99	S0.95	S0.90	S0.99	S0.95	S0.90	S0.99	S0.95	S0.90	S0.99	S0.95	S0.90
ScNet	51.69	47.90	48.78	83.70	70.06	52.28	66.59	50.23	50.00	51.17	51.12	51.48	68.49	50.39	49.14	50.42	51.00	50.96
ENet	66.04	49.32	49.32	63.29	49.94	48.48	63.63	50.00	48.98	52.64	48.00	47.98	53.14	50.56	50.56	51.81	49.42	47.96
TextureNet	66.04	50.00	50.00	50.79	50.72	52.29	71.39	69.39	50.96	65.06	54.89	50.24	86.95	78.83	73.12	76.20	63.17	63.03
CGNet	50.02	50.00	50.45	51.25	51.07	51.23	80.06	50.00	50.96	52.64	50.34	50.24	77.37	63.12	49.87	62.00	54.45	52.21
Ours	86.22	77.93	61.66	84.38	81.25	65.42	97.78	96.39	92.62	98.23	91.81	79.87	97.78	94.42	84.48	97.18	92.06	79.89

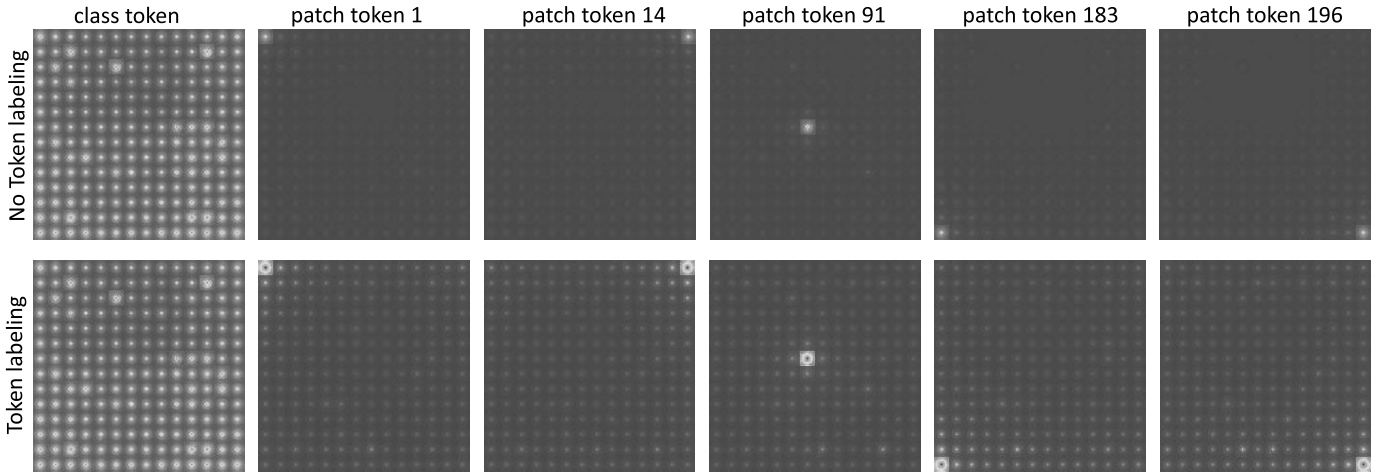


Fig. 8. Illustration of effective receptive field of our network without token labeling (the top row) and with token labeling (the bottom row).

TABLE XI
THE PERFORMANCE COMPARISON OF OUR METHOD WITH FIVE CNN-GENERATED IMAGE DETECTION METHODS

Network	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
SpecCNN	*80.16*	68.78	64.86	56.94	58.05	58.05
FSD	*66.58*	46.25	60.97	61.25	52.78	51.94
LFA	*71.18*	53.25	62.22	53.47	56.81	55.83
CNNDet	*92.53*	88.93	83.80	87.00	81.98	82.55
Xception2	*79.18*	83.84	61.39	58.19	59.17	58.61
Ours	***96.76***	**95.25**	**91.12**	**92.23**	**87.09**	**85.42**

soft cross entropy loss, we have also tried other strategies: hard labeling ("HL"), label smoothing ("LS"), two classifiers ("TC"), and mean pooling ("MP"). "HL" means that class token and patch token both use the classical cross entropy loss as Eqn. (6). "LS" adopts the label smoothing [73] for class token and patch token. Label smoothing was initially proposed in [73] for regularizing the training of CNN recognition models. Different from Eqn. (6), the label smoothing computes the cross entropy loss over a weighted mixture of original label of datasets with the uniform distribution.[2] "TC" refers to the variant where the class token and patch token are processed respectively with two different classification headers. "MP" conducts a global average on all patch tokens, and then sends it to the classification header. As shown in Fig. 4 (part of "Token Prediction"), in our experiments "Baseline" means that the model is trained without token labeling, *i.e.*, only using Eqn. (6). We conduct the ablation experiments on LSCGB and SPL2018 datasets, and the corresponding results are reported in Tables XIX and XX. When comparing the results of "Baseline" with those of other rows, we can find that the token labeling is indeed useful for the CG image forensic problem, in particular to improve generalization. Furthermore, our proposed soft cross entropy loss has superior overall performance, compared to the other four strategies.

[2]Label smoothing is implemented with the operation of timm.loss. LabelSmoothingCrossEntropy in PyTorch.

TABLE XII

THE PERFORMANCE COMPARISON OF OUR METHOD AND FIVE ADVANCED METHODS ON AIGC-GENERATED IMAGES

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	60.61	**71.45**	63.25	**80.58**	**94.20**	**75.07**	**59.59**	**58.77**	65.83	61.19	56.17	**80.95**	65.75	**68.72**
FSD	44.21	54.46	48.73	63.97	50.93	58.04	50.36	46.54	52.78	55.37	50.23	51.40	58.00	52.69
LFA	48.19	52.41	43.08	72.07	66.18	51.19	50.49	58.55	**67.22**	**63.70**	63.16	67.75	60.95	58.84
CNNDet	49.75	34.35	47.77	51.92	46.54	49.17	48.91	48.95	54.16	46.80	45.86	32.35	41.40	45.99
Xception2	67.40	64.35	64.85	66.99	52.00	53.96	50.00	50.00	62.78	63.48	51.53	69.00	60.00	59.71
Ours	**71.00**	62.72	**65.33**	65.66	87.52	66.39	50.04	50.03	53.06	61.42	**69.42**	80.65	**67.00**	65.40

TABLE XIII

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST SCALE160

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	54.32	**61.51**	55.52	52.88	66.03	49.44	48.07	51.80	50.00	**63.25**	49.62	77.40	**71.75**	57.82
FSD	50.08	54.94	44.60	**62.15**	51.70	56.49	48.52	41.42	50.28	47.73	50.43	52.95	64.90	53.06
LFA	52.62	51.61	51.08	48.60	51.25	40.55	50.78	49.09	55.00	40.81	42.66	46.50	51.55	48.62
CNNDet	51.30	49.42	51.17	47.54	46.52	46.69	50.00	50.00	48.61	49.31	50.36	56.15	48.30	49.64
Xception2	50.58	46.75	47.27	41.14	41.57	44.13	**74.27**	**73.79**	**65.56**	60.62	49.94	67.85	60.80	55.71
Ours	**70.75**	59.32	**58.33**	58.99	**79.44**	**60.49**	50.84	50.85	50.84	54.80	**61.11**	**80.50**	68.50	**61.90**

TABLE XIV

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST SCALE500

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	59.50	**71.36**	56.52	67.56	**85.32**	53.21	42.85	44.54	56.11	61.58	50.69	76.85	**71.15**	61.32
FSD	62.89	56.83	49.65	56.89	49.10	53.36	47.66	40.50	50.56	58.95	50.06	45.85	54.75	52.08
LFA	51.99	53.56	48.32	62.53	60.50	36.17	46.98	49.65	52.22	51.55	**53.12**	51.65	55.60	51.83
CNNDet	47.50	38.98	47.75	51.53	48.84	47.86	43.87	43.58	53.05	48.17	51.15	33.70	44.60	46.19
Xception2	52.44	51.58	56.43	**79.18**	50.48	47.55	50.00	50.00	50.00	53.94	49.97	56.70	55.50	54.13
Ours	**78.12**	63.98	**76.71**	71.72	76.49	**73.02**	**50.37**	**50.52**	**58.34**	**61.88**	52.20	**80.35**	65.30	**66.07**

TABLE XV

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST JPEG90

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	52.38	56.22	48.70	50.68	49.22	51.25	**54.69**	**53.05**	47.78	48.69	52.95	53.75	52.75	51.70
FSD	50.08	54.94	44.60	62.15	51.70	56.49	48.52	41.42	50.28	47.73	50.43	52.95	**64.90**	52.01
LFA	50.60	51.99	47.60	**64.00**	50.15	51.78	47.70	43.64	48.06	44.87	52.36	54.15	56.60	51.03
CNNDet	50.00	33.67	48.22	51.92	46.49	49.30	51.98	50.18	**55.00**	46.11	45.84	32.90	41.65	46.40
Xception2	51.01	50.13	50.32	50.57	50.03	51.14	50.00	50.00	50.00	51.07	50.05	50.60	49.45	50.33
Ours	**68.75**	**64.53**	**59.48**	57.09	**69.24**	**60.75**	49.76	50.15	44.73	**57.31**	**65.22**	**76.15**	61.00	**60.32**

TABLE XVI

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST VAR1

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	57.86	62.44	59.63	**71.08**	64.81	**65.70**	**62.16**	**62.59**	**61.67**	57.04	51.56	49.25	57.40	60.24
FSD	49.75	53.51	48.15	60.33	52.65	54.26	50.15	41.82	53.89	63.48	45.59	58.10	55.00	52.82
LFA	50.09	54.10	45.85	64.38	64.33	50.38	51.95	60.17	**61.67**	**64.68**	53.71	51.10	**61.85**	56.48
CNNDet	49.50	30.29	48.15	51.92	42.27	49.44	50.00	51.70	49.72	46.57	47.75	42.20	42.05	46.27
Xception2	**78.80**	**73.01**	59.42	59.84	81.09	51.98	50.00	50.00	60.28	62.05	52.38	**79.60**	61.75	**63.09**
Ours	68.75	63.43	**64.06**	62.92	**85.60**	63.46	50.04	50.05	49.73	58.91	**63.23**	74.90	59.15	62.63

TABLE XVII

THE PERFORMANCE COMPARISON OF ROBUSTNESS AGAINST SNR0.99

Network	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFake	SD	MJ	Avg
SpecCNN	51.01	50.91	51.02	51.74	52.73	48.62	50.85	49.00	51.67	44.15	49.66	58.00	49.80	50.70
FSD	51.38	49.30	49.85	56.67	53.05	51.97	51.85	48.13	49.72	49.88	49.92	**74.15**	**59.30**	53.47
LFA	49.49	50.10	50.03	49.92	49.90	49.94	51.79	52.00	53.89	48.45	**50.54**	48.85	49.45	50.33
CNNDet	49.50	51.17	51.97	54.23	49.67	49.88	**52.78**	**52.75**	48.61	47.94	46.17	49.05	47.30	50.07
Xception2	56.35	51.01	**53.05**	52.16	50.00	45.78	50.00	50.00	**61.94**	56.56	49.92	60.05	59.15	53.53
Ours	**63.50**	**64.80**	50.93	**59.23**	**58.58**	**62.50**	50.06	49.86	45.56	**60.74**	50.16	61.95	46.90	**55.75**

D. Visualization & Analysis

1) Effective Receptive Field: To further analyze the effect of token labeling (*i.e.*, Eqn. (7)), we visualize the effective receptive field of our network according to the method proposed in [46]. This is also used in a recent ViT-related research work [74]. Fig. 8 shows the effective receptive field of class

TABLE XVIII

CLASSIFICATION PERFORMANCE FOR DIFFERENT PATCH SIZES: 8 × 8 (P8), 16 × 16 (P16), AND 32 × 32 (P32)

Training \ Testing	LSCGB			SPL2018			Autodesk			Artlantis			Corona			VRay		
	P8	P16	P32	P8	P16	P32	P8	P16	P32	P8	P16	P32	P8	P16	P32	P8	P16	P32
LSCGB	***98.73***	*98.00*	*96.98*	96.75	**97.13**	96.85	87.37	**95.56**	95.55	87.50	**94.87**	94.73	89.73	**92.23**	90.70	85.00	**90.98**	90.84
SPL2018	**80.89**	79.53	77.56	***99.54***	*99.16*	*98.97*	86.04	**90.70**	89.87	85.84	88.62	**88.89**	87.02	**88.84**	88.20	85.01	86.67	**87.09**
Autodesk	**65.61**	65.44	65.40	**87.06**	76.72	85.91	*98.65*	***99.03***	*98.75*	95.42	95.76	**96.12**	86.89	84.76	**87.64**	86.55	85.14	**89.45**
Artlantis	63.99	67.20	**67.99**	90.26	87.29	**91.16**	**98.60**	97.78	98.20	*98.65*	***99.03***	*98.75*	87.33	87.95	**89.73**	**94.12**	91.81	92.92
Corona	66.05	**68.94**	66.80	86.77	84.21	**87.97**	91.24	**92.48**	91.12	87.04	**87.89**	87.45	*99.09*	***99.16***	*98.89*	95.66	96.08	**96.67**
VRay	66.98	**70.88**	69.88	82.86	**85.66**	**85.66**	**91.14**	90.23	90.14	**88.01**	87.89	86.81	**97.79**	96.81	97.50	***99.03***	***99.03***	***99.03***

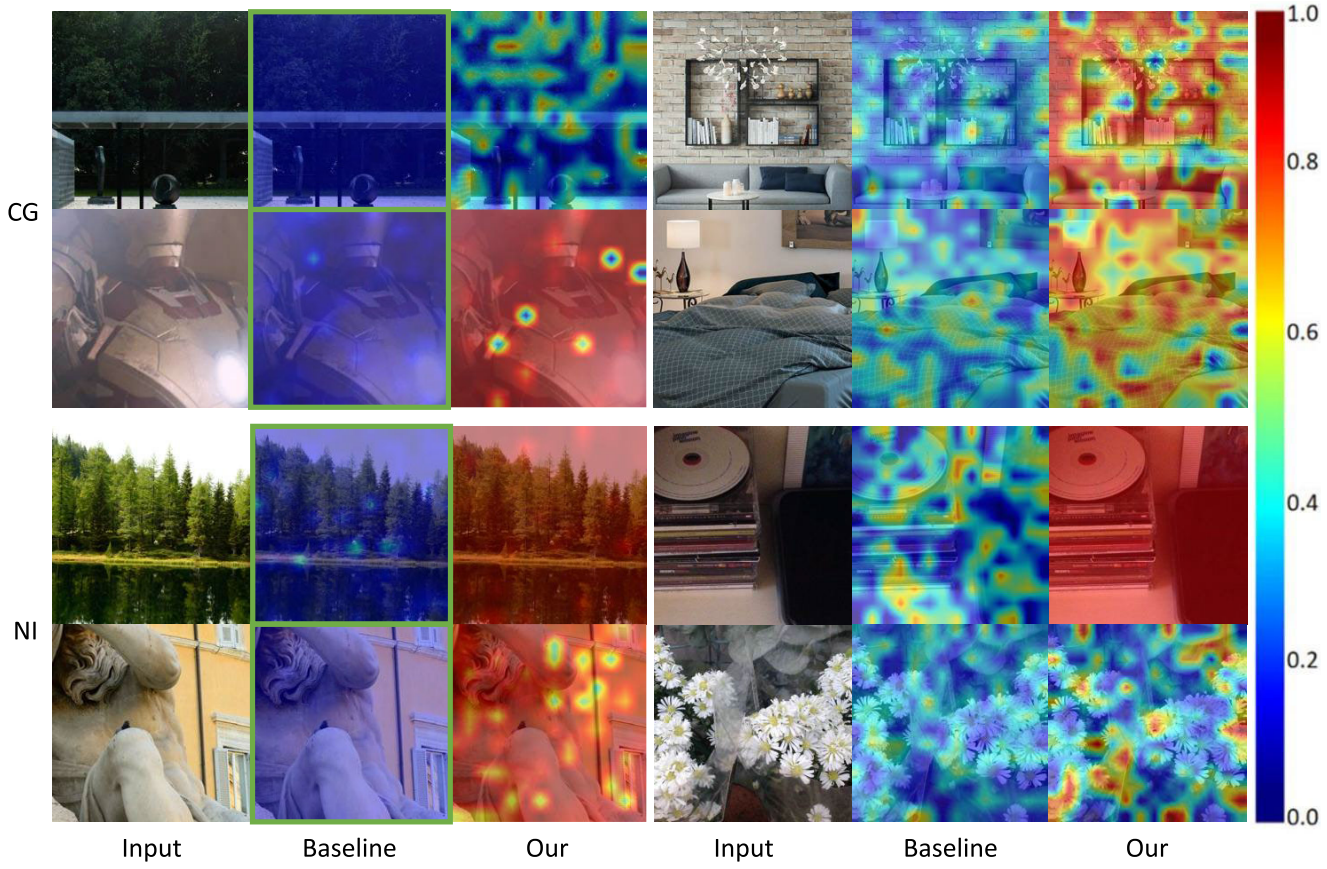


Fig. 9. Illustration of prediction confidence of patch tokens. Green box highlights the wrong prediction.

TABLE XIX

ABLATION STUDIES OF TOKEN LABELING WHEN TRAINED ON LSCGB

Method	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
Baseline	*96.13*	95.69	89.31	88.48	89.87	87.64
Hard label	*97.59*	96.13	94.87	93.34	91.53	**91.25**
Label smooth	*97.83*	96.41	93.48	94.17	91.12	90.28
Two classifiers	*97.73*	96.13	94.73	94.17	91.12	88.75
Mean pooling	*97.75*	96.35	94.31	93.48	91.81	90.14
Ours	***98.00***	**97.13**	**95.56**	**94.87**	**92.23**	90.98

TABLE XX

ABLATION STUDIES OF TOKEN LABELING WHEN TRAINED ON SPL2018

Method	LSCGB	SPL2018	Autodesk	Artlantis	Corona	VRay
Baseline	73.93	*98.47*	80.14	82.92	83.48	79.31
Hard label	75.39	*98.85*	83.48	84.87	86.09	80.70
Label smooth	76.01	***99.19***	86.12	86.95	88.20	83.20
Two classifiers	75.36	*98.88*	85.98	86.53	88.20	83.48
Mean pooling	76.15	*99.04*	85.14	86.95	87.78	83.48
Ours	**79.53**	*99.16*	**90.70**	**88.62**	**88.48**	**86.67**

token and several selected patch tokens, where our network is trained with/without token labeling. Comparing the two rows of Fig. 8, we can see that with token labeling (the bottom row), the effective receptive field of patch tokens are more uniform and can cover the full input image. This means that patch tokens capture more far-distance information and thus enhance the learned features.

In addition, we evaluate the forensic performance of the above patch tokens and the statistical results are shown in Table XXI. Comparing the numerical values of "w/o TL" and "w TL", we can find that the classification performance of patch tokens is in general considerably improved after adding our soft cross entropy loss. This performance improvement is also consistent with the visualization shown in Fig. 8.

TABLE XXI
THE CLASSIFICATION PERFORMANCE OF PATCH TOKENS. "TL" REFERS TO OUR PROPOSED TOKEN LABELING (EQN. (7))

		LSCGB		SPL2018		Autodesk		Artlantis		Corona		VRay	
	Patch Index	w/o TL	w TL	w/o TL	w TL	w/o TL	w TL	w/o TL	w TL	w/o TL	w TL	w/o TL	w TL
LSCGB	P1	*96.26*	*97.99*	95.81	96.75	89.45	95.48	89.31	94.73	90.14	92.89	88.20	91.48
	P14	*96.24*	*98.01*	95.66	96.85	89.45	95.56	89.03	94.59	90.00	92.23	87.64	91.45
	P91	*96.12*	*97.78*	95.75	96.88	89.31	95.32	88.48	94.73	89.59	92.25	87.92	90.88
	P183	*96.24*	*98.01*	95.66	96.91	89.45	95.62	88.89	94.45	89.73	93.04	87.92	90.79
	P196	*96.24*	*98.00*	95.72	96.85	89.53	95.82	89.03	94.31	89.87	92.64	87.92	90.64
SPL2018	P1	73.92	79.11	*98.44*	*99.07*	80.56	90.23	82.78	88.39	83.62	88.62	79.87	86.74
	P14	73.91	79.89	*98.47*	*99.03*	80.56	89.85	83.06	88.13	83.75	88.48	80.00	86.61
	P91	73.92	79.41	*98.35*	*99.13*	80.28	89.97	82.92	88.62	83.34	89.31	79.59	86.59
	P183	73.85	79.07	*98.47*	*99.13*	80.28	89.49	82.92	88.29	83.48	88.48	79.73	86.21
	P196	73.85	79.02	*98.50*	*99.07*	80.28	90.65	83.06	88.13	83.75	88.75	79.73	86.37

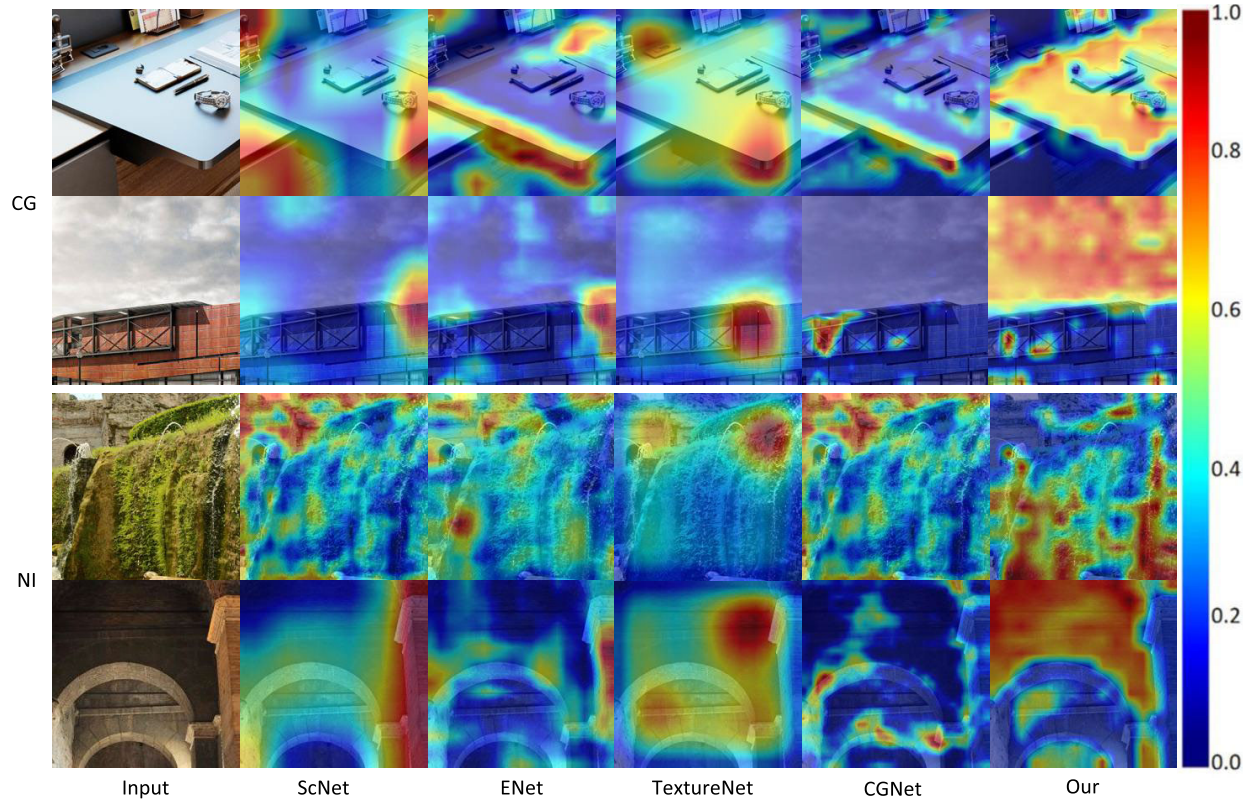


Fig. 10. Grad-CAM visualizations for the "CG" and "NI" category.

2) Prediction Confidence: Fig. 9 illustrates the prediction confidence for several NI and CG images using the predictions of patch tokens. "Baseline" means that the CGFormer is trained without token labeling, and the samples enclosed by the green boxes are wrongly classified by "Baseline". Compared to "Baseline", we can observe that more patches are correctly classified by introducing the token labeling. The reason is that token labeling adds the soft supervision on individual patches to take advantage of the useful information that has been overlooked in patches. Therefore, our CGFormer can obtain better forensic performance by comprehensively combining the contributions from class token and patch tokens.

3) Grad-CAM: Gradient-weighted class activation mapping (Grad-CAM) [75] is a prevalent tool to localize the important regions in the input image for the final prediction. This tool is conceptually simple and easily applicable to different network architectures like CNNs and Transformers and it is often utilized in the image forensics community for understanding the results of deep networks. As shown in Fig. 10, we visualize the Grad-CAM for NIs and CG images to compare our method with existing CNN-based methods. For the image in the first row of "CG" group in Fig. 10, our method pays more attention to the desk with unnatural light reflection and simple texture comparing to other competitors, and this contributes to the final prediction. By contrast, our CGFormer predicts the first sample in "NI" group as NI image mainly due to the natural texture in the vegetation. For the second row in "NI" group, most CNN-based methods focus on the semantic object (*e.g.*, the contour of castle), whereas our method depends on the natural shadows in the top part of castle. Similarly, our CGFormer predicts the second sample in "CG" group as CG image due to the rather unrealistic sky. To summarize, light, texture, shadow, and color are important cues for distinguishing CG images from natural images.

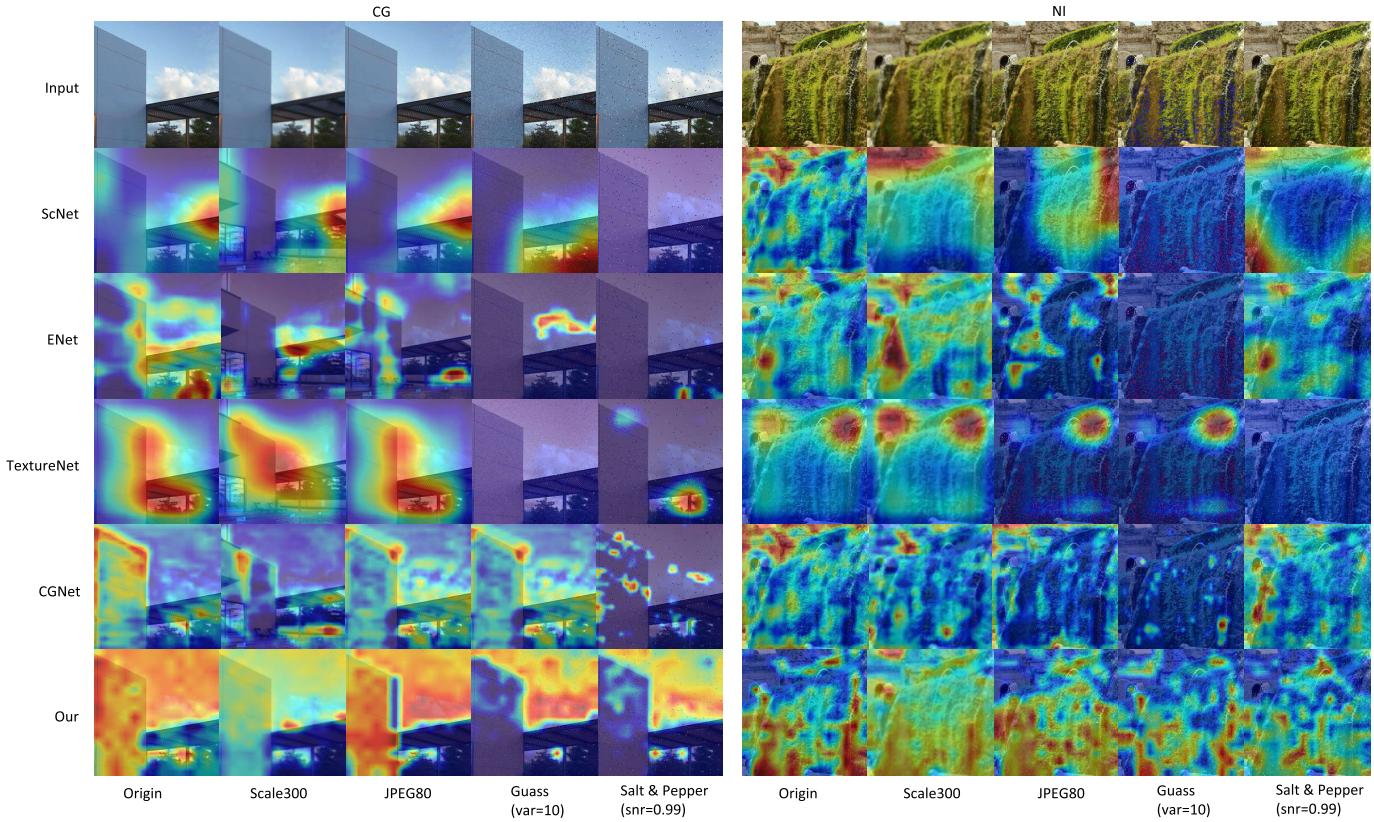


Fig. 11. Grad-CAM visualizations of five methods for CG images and NIs with post-processing of rescaling, JPEG compression, Gaussian noise, and salt and pepper noise.

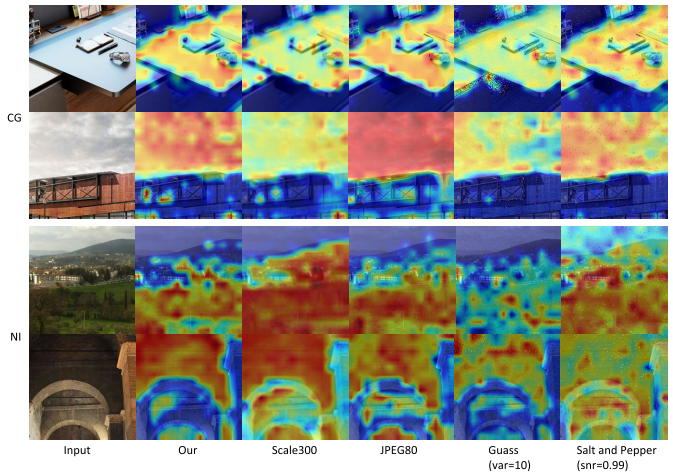


Fig. 12. Grad-CAM visualizations of our CGFormer for CG images and NIs with post-processing of rescaling, JPEG compression, Gaussian noise, and salt and pepper noise.

In addition, we visualize in Fig. 11 the Grad-CAM of NIs and CG images under different post-processing operations. Among all five methods, we can see that the Grad-CAM of our method has no apparent changes for the vast majority of samples. However, the post-processing operations clearly affect the capture of forensic traces for CNN-based competitors. This visualization again illustrates the robustness of our CGFormer against potential post-processing operations. Moreover, Fig. 12 visualizes the Grad-CAM of our CGFormer for more examples under different post-processing operations. The important regions related to the final prediction of our method are stable for these examples, which means the good robustness of our CGFormer.

V. Conclusion and Future Work

For the CG image forensic problem, we carefully analyzed the possible reasons of the limited generalization and robustness of existing CNN-based methods, including the locality and fixed kernels of convolutional layers, the sensitivity to the pixel values and their distributions in a local window, and the insufficient supervision information. Based on the observations and analyses on data and CNN, we proposed a ViT-based framework with token labeling. Patch embedding is applied to project the original input image into sequential tokens in a patch-wise manner, which serves for the subsequent patch relationship modeling and can also decrease the local sensitivity. To improve the model's adaptability and extract the discriminative information based on the relationship modeling of patch pairs, we adopt multiple transformer blocks with multi-head attention. A token labeling strategy is proposed to better utilize the supervision information and further guide the network to exploit the local-global relationship of a single image. Extensive results demonstrate the superior forensic performance of our proposed method.

In this work, we choose 16×16 as the patch size in the patch embedding because this setting achieves relatively superior results. In fact, CGFormer with 8×8 and 32×32 patches sometimes also achieves decent results. In the future, we would

like to design an appropriate architecture to combine these three patch sizes to further improve the forensic performance. We would also like to extend our framework to other image forensic problems, such as deepfakes detection.

REFERENCES

[1] W. Quan, K. Wang, D.-M. Yan, X. Zhang, and D. Pellerin, "Learn with diversity and from harder samples: Improving the generalization of CNN-based detection of computer-generated images," *Forensic Sci. Int., Digit. Invest.*, vol. 35, Dec. 2020, Art. no. 301023.

[2] W. Bai et al., "Robust texture-aware computer-generated image forensic: Benchmark and algorithm," *IEEE Trans. Image Process.*, vol. 30, pp. 8439–8453, 2021.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[4] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proc. ACM Int. Conf. Multimedia*, 2005, pp. 239–248.

[5] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 845–850, Feb. 2005.

[6] W. Quan, K. Wang, D.-M. Yan, and X. Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2772–2787, Nov. 2018.

[7] P. He, H. Li, H. Wang, and R. Zhang, "Detection of computer graphics using attention-based dual-branch convolutional neural network from fused color components," *Sensors*, vol. 20, no. 17, p. 4743, Aug. 2020.

[8] Y. Yao, Z. Zhang, X. Ni, Z. Shen, L. Chen, and D. Xu, "CGNet: Detecting computer-generated images based on transfer learning with attention module," *Signal Process., Image Commun.*, vol. 105, Jul. 2022, Art. no. 116692.

[9] J. Long, N. Zhang, and T. Darrell, "Do ConvNets learn correspondence?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] P. He, X. Jiang, T. Sun, and H. Li, "Computer graphics identification combining convolutional and recurrent neural networks," *IEEE Signal Process. Lett.*, vol. 25, no. 9, pp. 1369–1373, Sep. 2018.

[13] P. Yang, D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva, "A survey of deep learning-based source image forensics," *J. Imag.*, vol. 6, no. 3, p. 9, Mar. 2020.

[14] I. C. Camacho and K. Wang, "A comprehensive review of deep-learning-based methods for image forensics," *J. Imag.*, vol. 7, no. 4, p. 69, Apr. 2021.

[15] F. Pan, J. Chen, and J. Huang, "Discriminating between photorealistic computer graphics and natural images using fractal geometry," *Sci. China Ser. F, Inf. Sci.*, vol. 52, no. 2, pp. 329–337, Feb. 2009.

[16] G. Sankar, V. Zhao, and Y.-H. Yang, "Feature based classification of computer graphics and real images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1513–1516.

[17] R. Zhang, R.-D. Wang, and T.-T. Ng, "Distinguishing photographic images and photorealistic computer graphics using visual vocabulary on local image edges," in *Proc. Int. Workshop Digit. Watermarking*, 2012, pp. 292–305.

[18] Z. Li, J. Ye, and Y.-Q. Shi, "Distinguishing computer graphics from photographic images using local binary patterns," in *Proc. Int. Workshop Digit. Watermarking*, 2013, pp. 228–241.

[19] F. Peng and D.-L. Zhou, "Discriminating natural images and computer generated graphics based on the impact of CFA interpolation on the correlation of PRNU," *Digit. Invest.*, vol. 11, no. 2, pp. 111–119, Jun. 2014.

[20] F. Peng, D.-L. Zhou, M. Long, and X.-M. Sun, "Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis," *AEU-Int. J. Electron. Commun.*, vol. 71, pp. 72–81, Jan. 2017.

[21] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun. 2003, pp. 1–8.

[22] W. Chen, Y. Q. Shi, and G. Xuan, "Identifying computer graphics using HSV color model and statistical moments of characteristic functions," in *Proc. IEEE Multimedia Expo Int. Conf.*, Jul. 2007, pp. 1123–1126.

[23] L. Ozparlak and I. Avcibas, "Differentiating between images using wavelet-based transforms: A comparative study," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1418–1431, Dec. 2011.

[24] J. Wang, T. Li, Y.-Q. Shi, S. Lian, and J. Ye, "Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics," *Multimedia Tools Appl.*, vol. 76, no. 22, pp. 23721–23737, Nov. 2017.

[25] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proc. IEEE Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2017, pp. 1–6.

[26] Y. Yao, W. Hu, W. Zhang, T. Wu, and Y.-Q. Shi, "Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning," *Sensors*, vol. 18, no. 4, p. 1296, Apr. 2018.

[27] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2307–2311.

[28] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.

[29] D. B. Tarianga, P. Senguptab, A. Roy, R. S. Chakraborty, and R. Naskar, "Classification of computer generated and natural images based on efficient deep convolutional recurrent attention model," in *Proc. CVPR*, 2019, pp. 146–152.

[30] R.-S. Zhang, W.-Z. Quan, L.-B. Fan, L.-M. Hu, and D.-M. Yan, "Distinguishing computer-generated images from natural images using channel and pixel correlation," *J. Comput. Sci. Technol.*, vol. 35, no. 3, pp. 592–602, May 2020.

[31] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8692–8701.

[32] K. Chandrasegaran, N.-T. Tran, A. Binder, and N.-M. Cheung, "Discovering transferable forensic features for CNN-generated images detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 671–689.

[33] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? Understanding properties that generalize," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 103–120.

[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[35] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," 2018, *arXiv:1812.02510*.

[36] L. Li et al., "Face X-ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5000–5009.

[37] M. Kim, S. Tariq, and S. S. Woo, "FReTAL: Generalizing deepfake detection using knowledge distillation and representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1001–1012.

[38] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18699–18708.

[39] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2019, pp. 1–6.

[40] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3247–3258.

[41] T. Dzanic, K. Shah, and F. D. Witherden, "Fourier spectrum discrepancies in deep network generated images," in *Proc. Adv. Neural Inform. Process. Syst.*, 2020, pp. 3022–3032.

[42] R. Durall, M. Keuper, and J. Keuper, "Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7887–7896.

[43] K. Chandrasegaran, N.-T. Tran, and N.-M. Cheung, "A closer look at Fourier spectrum discrepancies for CNN-generated images detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7196–7205.

[44] C. Dong, A. Kumar, and E. Liu, "Think twice before detecting GAN-generated fake images from their spectral domain imprints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7855–7864.

[45] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[46] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4905–4913.

[47] Q. Han et al., "On the connection between local attention and dynamic depth-wise convolution," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–14.

[48] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–12.

[49] A. Vaswan et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[50] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–14.

[51] Z. Jiang et al., "All tokens matter: Token labeling for training better vision transformers," in *Proc. Adv. Neural Inform. Process. Syst.*, 2021, pp. 18590–18602.

[52] T.-T. Ng, S.-F. Chang, J. Hsu, and M. Pepeljugoski, "Columbia photographic images and photorealistic computer graphics dataset," ADVENT, Dept. Elect. Eng., Columbia Univ., Tech. Rep. 205-2004-5, 2004.

[53] E. Tokuda, H. Pedrini, and A. Rocha, "Computer generated images vs. digital photographs: A synergetic feature and classifier combination approach," *J. Vis. Commun. Image Represent.*, vol. 24, no. 8, pp. 1276–1292, Nov. 2013.

[54] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[55] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[56] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.

[57] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–11.

[58] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, Mar. 2015, pp. 219–224.

[59] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, p. 15, Dec. 2017.

[60] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.

[61] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[63] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[64] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[65] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[66] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2332–2341.

[67] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1520–1529.

[68] K. Li, T. Zhang, and J. Malik, "Diverse image synthesis from semantic layouts via conditional IMLE," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Mali, Oct. 2019, pp. 4219–4228.

[69] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3291–3300.

[70] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11057–11066.

[71] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[73] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[74] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12116–12128.

[75] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.