Detecting Computer-Generated Images by Using Only Real Images

Ji Li, Kai Wang

Detecting Computer-Generated Images by Using Only Real Images

Ji Li and Kai Wang

Abstract

This paper presents a simple yet effective method to detect fake synthetic images generated by recent deep generative models, the so-called deepfakes. Unlike existing methods that require a relatively large number of real and fake training images, our method follows a novel idea of using only real images during the training phase. Our proposal is to construct proxy negative training samples, representing fake images, by applying an appropriate transformation on the real images in the training set. The training of our detector leverages the popular CLIP model as well as a center loss to encourage clustering of real images, with the aim of obtaining discriminative features for the classification of real and fake images. The proposed forensic detector is conceptually simple and data-efficient, i.e., it can be trained by using a small amount of only 4K real images. Experimental results and comparisons show the effectiveness of our method in terms of generalization capability to detect fake images generated by various deep generative models.

Index Terms

Image forensics, fake image detection, deepfake, CLIP, color transfer, neural network.

I. Introduction

The proliferation of smartphones and social networking applications has significantly increased the ease of acquiring and sharing digital images in today's world. In the meantime, the availability of advanced image generation tools has made it straightforward to generate synthetic images with a very high level of visual realism. Such computer-generated fake images, which do not reflect reality of the physical world, can have serious negative impacts on society. They can be used to mislead public opinion, deceive consumers, and notably, impede law enforcement efforts if generated images are accepted as credible evidence in court. Indeed, the significant evolution of generative models, from Generative Adversarial Networks (GANs) [1] to the latest state-of-the-art diffusion models [2], has resulted in computer-generated images (see Fig. 1 for examples) of astonishing quality that are often indistinguishable from real images by humans. This advancement raises substantial concerns regarding their potential misuse for malicious purposes. Consequently, numerous image forensic techniques have been proposed over the last two decades [3–5] to detect various types of synthesized images.

Fig. 1. Computer-generated images by different generative models. Top row: GANs. Bottom row: Diffusion models.

J. Li and K. Wang are with Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France (e-mail: ji.li@gipsa-lab.grenoble-inp.fr, kai.wang@gipsa-lab.grenoble-inp.fr).

In image forensics, particularly in the detection of computer-generated images [6–8], deep learning-based methods, primarily using convolutional neural networks (CNNs), often excel in fully supervised scenarios with ample labeled training samples. Typically, deep learning-based classifiers trained on datasets of real images and fake ones from specific generative models (e.g., GANs, diffusion models) perform well within the same generative model family [9, 10]. For instance, a classifier trained on real and fake images from a specific GAN model can achieve high accuracy when detecting fake images from other GAN models. However, performance can significantly degrade when facing with images generated by diffusion models. Such classifiers often struggle to generalize to out-of-domain models, especially those not included in the training set. In addition, existing methods typically require a relatively large amount of training data. One of our goals is to design a *data-efficient* fake image detection method, capable of achieving high detection performance using a minimal amount of data.

In this paper, we first conduct experiments to investigate the transferability of different large pretrained image encoder models, varying in pretraining data and strategies, in the context of fake image detection. Based on the findings of this study, we propose a data-efficient method for detecting computer-generated images, capable of handling challenging situations (in particular related to the generalization capability on unseen image generative models), by training the detection model using *only* real images. Specifically, we build a model based on the very popular CLIP [11] image encoder, sufficiently exploiting the capabilities of this large pretrained model by fusing features from different intermediate blocks.

Importantly, we succeed in training the model using only a small number of real images and an equally small number of so-called *proxy negative samples* constructed from real images. We do not use any computer-generated image during the training phase. The proxy samples represent computer-generated images during training and in the meantime should be, to some extent, close to real images in order to make the classification challenging. Our main motivation is as follows: different generative models generate fake images with their corresponding intrinsic distribution and trace, and it is impractical to include all generative models in the training set; by contrary, our proposed detector, trained to classify between real images and a set of challenging proxy samples without any specific trace of generative model, could be more favorable to achieve better generalization capability. Additionally, we implement center loss on the embeddings of real images, aiming to cluster their feature representations, which further enhances the network's discrimination ability.

We tested our method on datasets that contain GANs, diffusion models, and others. By leveraging only real images and proxy samples constructed from them, without introducing additional information about fake images, our method, trained by using only 4K samples of real images, achieves superior performance compared to the baselines, demonstrating strong generalization capability across different domains with high accuracy. Our main contributions can be summarized as: (1) we explore the transferable capabilities of different pretrained large models for computer-generated image detection; (2) we propose the use of real images as well as the proxy negative samples for training, enhancing the classifier's generalization ability; and (3) we conduct experiments to show the effectiveness of our method in terms of detection accuracy and generalization across unseen domains.

The paper is structured as follows. Section II reviews related work. Section III details our approach. Section IV presents experimental results. Finally, Section V concludes the paper and gives directions for future research.

II. Related Work

Computer-generated images often contain operation traces left during the generation process, which are termed artificial fingerprints. These fingerprints can be utilized to determine whether images are real or fake. In general, image detection methods can be roughly divided into frequency-based methods and spatial-based ones.

A. Frequency-Based Detection

In some cases, fingerprints in computer-generated images can be more easily detected in the frequency domain after applying the Fourier transform to the synthetic images [12, 13]. Zhang et al. [14] proposed a GAN simulator to replicate artifacts generated by the common pipeline shared among different GANs. Frank et al. [15] investigated artifacts across various GAN architectures in the frequency domain and illustrated how artifacts induced by different upsampling strategies in networks can be leveraged for identifying deepfake images.

While similar artifacts can also manifest in diffusion models, they are notably more pronounced in GAN models [16] due to the inherent use of upsampling operations in the generator, leading to discernible patterns in the Fourier domain. Chandrasegaran et al. [17] argued that discrepancies in high-frequency spectral decay are not inherent characteristics of CNN-generated images, and such features lack robustness for synthetic image detection. They demonstrated that a slight modification to the final upsampling layer of the architecture enabled computer-generated images to evade recently proposed forensic detectors relying on high-frequency Fourier spectrum decay attributes for CNN-generated image detection.

B. Spatial-Based Detection

In the spatial domain, image forensic techniques analyze pixel values directly. Methods in this domain focus on detecting inconsistencies or artifacts introduced during fake image generation [16, 18, 19]. Wang et al. [9] suggested that CNN-generated images exhibit systematic subtle flaws when compared to authentic real images. Through careful pretraining and

data augmentation, a simple image classifier (ResNet-50) trained on a specific CNN generator (ProGAN) can generalize well to unseen GAN architectures.

Although the classifier in [9] achieves impressive performance on images generated within the same domain (e.g., GANs), it exhibits a significant drop in performance when confronted with diffusion-generated images, with accuracy results falling below 60%. To address this issue, the authors of [10] proposed a novel image representation called Diffusion Reconstruction Error (DIRE), which measures the error between an input image and its reconstruction counterpart by a trained diffusion model. By leveraging these reconstruction errors as fingerprints, a binary classifier can effectively distinguish between real and fake images, even for unseen diffusion models, demonstrating robustness against various perturbations. However, this generalization capability did not consistently perform well on all models, particularly when applied to images generated from unseen data sources.

Ojha et al. [20] discovered that the decision boundary for the classifier of [9] is closely bound to the particular fake domain. Whenever an image contains the (low-level) fingerprints particular to the generative model used for training (e.g., ProGAN), the image gets classified as fake; otherwise, it is classified as real. This tendency arises because the classifier easily latches onto the low-level image artifacts that differentiate fake images from real images, leading to overfitting to subtle artifacts specific to the training set and resulting in suboptimal performance on unseen generators. Ojha et al. [20] utilized the fixed pretrained image encoder of CLIP (ViT-L/14) [11] as a feature extractor and explored nearest neighbor and linear probing classification based on these feature maps. With training on the same dataset as [9] including 720K real/ProGAN images, their promising results across different domains, which mainly owe to the quality of the image embeddings of CLIP, inspire us to explore the capabilities of this very popular pretrained image encoder model.

III. Approach

In this paper, we attempt to perform data-efficient fake image detection with high accuracy. Naturally, transfer learning [21] comes to mind as it leverages knowledge acquired from a large dataset to improve the performance of a model on a different, often smaller dataset.

A. Analysis of Large Pretrained Models

Considering the excellent performance of the pretrained CLIP model [11] in various downstream tasks [22, 23], its encoded representations likely contain sufficient and important information about images. In this subsection, we investigate the sources of the modeling ability of the CLIP encoder, questioning whether different modeling strategies during pretraining or the scale of the dataset make a major contribution. Our goal is to explore the potential of large models when adapting to the fake image detection task. The image encoder used in CLIP is the Vision Transformer (ViT) [24], specifically the ViT/L-14 variant. Therefore, we considered several different open-source pretrained models based on vision transformers as follows:

- **ViT/L-16** [24], supervised classification on ImageNet-22K (with over 14 million images of about 22K classes)
- **Swin Transformer-L** [25], supervised classification on ImageNet-22K
- **BEiT-L** [26], masked autoencoder on ImageNet-22K
- **Swag** [27], supervised classification on Instagram-3.6B (about 3.6 billion images from Instagram)
- **MAE** [28, 29], masked autoencoder on Instagram-3B (about 3 billion images from Instagram)
- **MAWS** [29], masked autoencoder + supervised classification on Instagram-3B

It is important to note that all models use ViT-L as the feature extractor, except for the Swin Transformer. These models employ different modeling methods and were pretrained on various datasets (more details can be found in the original papers). We followed the same training methods as in [20], freezing the pretrained models as feature extractors and adding a linear layer as the classifier head. To assess the potential of large models for data-efficient fake image detection, we used only 8K real/ProGAN images for training and then tested the trained detectors on 19 other datasets.

TABLE I
Average detection accuracy (last column, in %) of different large pretrained models.

Model	Pretraining dataset	Modeling strategy	Author	Accuracy
ViT/L-16	ImageNet-22k	Supervised	Google	58.22
Swin Transformer-L		Supervised	Microsoft	79.34
BEiT-L		Masked autoencoder	Microsoft	62.56
Swag	Instagram-3.6B	Supervised	Meta	78.63
MAE	Instagram-3B	Masked autoencoder	Meta	82.08
MAWS		Masked autoencoder + supervised	Meta	82.57
CLIP	WIT-400M	Contrastive learning	OpenAI	**85.85**

The average classification accuracy results are shown in Table I. By comparing ViT/L-16, Swin Transformer-L, and Swag models, we found that the Swin Transformer exhibits superior feature extraction capabilities compared to ViT/L-16. Furthermore, scaling up the pretraining data from ImageNet-22K to Instagram-3B also enhances model performance, as demonstrated by comparing BEiT-L, MAE, and MAWS, which all conducted pretraining with masked autoencoder on different datasets. Among all the models, the CLIP encoder demonstrated the best performance. We attribute this to its strong feature extraction capabilities, which benefit from the extensive 400M curated pretraining data, as well as the image-text contrastive learning approach. Our observation is in line with other studies which show that in general CLIP has the best overall performance on a good range of downstream tasks compared to other large pretrained models [30–32]. To the best of our knowledge, this is the first time that different large models pretrained on different data sources and with different approaches have been compared in the context of a computer-generate image detection task.

B. Detection Model

Based on our analysis, we adopt CLIP:ViT-L/14 as the feature extractor. To fully exploit the capabilities of the CLIP image encoder for transferable detection, we utilize intermediate representations from the CLIP encoder, rather than solely relying on the last-layer output features [20]. While the features from the last layer of CLIP contain more high-level information about images, intermediate representations capture more fine-grained details, which are crucial for detecting fake images. Previous method [33] leverages intermediate representations of CLIP via a learnable block, here we propose a more concise way. Figure 2 illustrates the architecture of our model.

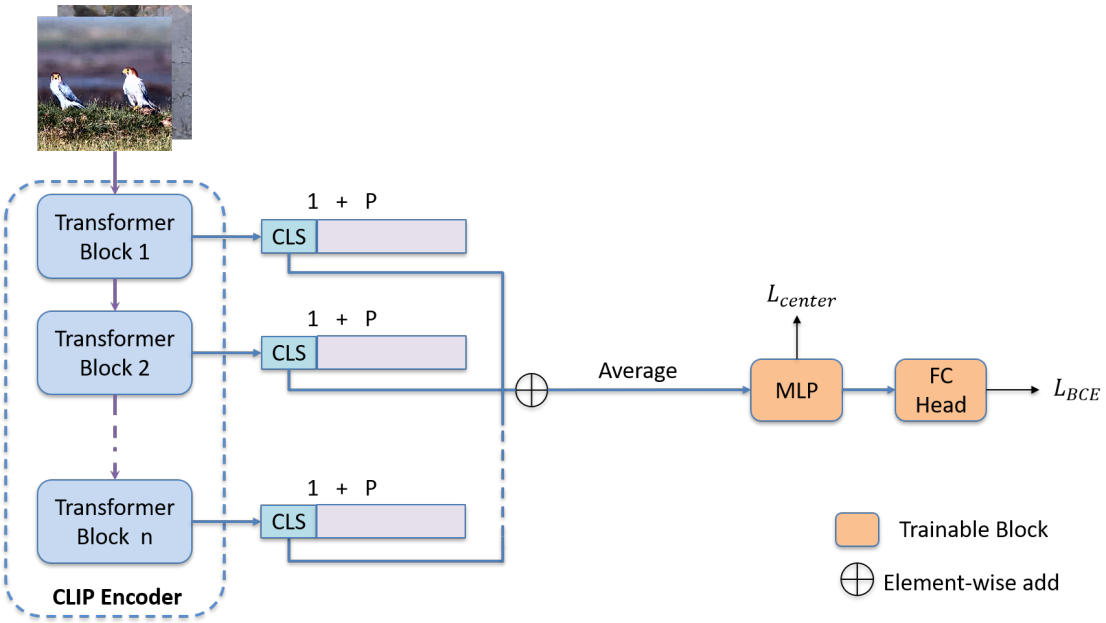


Fig. 2. The architecture of our model.

The feature extractor consists of n successive Transformer encoder blocks [34], where n is 23. For a batch of input $X \in \mathbb{R}^{b\times 3\times w\times h}$, with w and h the width and height of image and b the batch size, each Transformer block outputs the feature representations $Z_i \in \mathbb{R}^{b\times(1+p)\times d}$ by calculating multi-head self-attention, where p denotes the number of patches, and d is the embedding dimension. To fuse the features from different levels, we propose computing the average value of the n class tokens stemming from each of the n Transformer blocks as the final representation F, which takes into consideration all intermediate layers:

$$F = \text{Avg}\left(\oplus\left\{Z_i^{[0]}\right\}_{i=1}^{n}\right) \in \mathbb{R}^{b\times d}, \tag{1}$$

where $\oplus$ denotes element-wise addition, and $Z_i^{[0]} \in \mathbb{R}^{b\times 1\times d}$ denotes the class token from the output of Transformer block n. The Avg is the above equation is calculated along the second dimension. During training, all the parameters of the feature extractor are frozen.

Following this, an MLP layer is used for further purification of features and adapting the CLIP features to the image detection task by projecting the features to a dimension of 256. Finally, the classification head, consisting of a linear layer with sigmoid activation, predicts the output (real/fake).

C. Loss Function

To better learn the distinctive information of real images, we adopt a combination of BCE (Binary Cross Entropy) loss and Center Loss when training the model. Center loss is commonly used in one-class classification tasks [35, 36], and it enhances the discriminative power of deep features by minimizing the intra-class variance:

$$L_{\text{center}} = \sum_{i=1}^{N} \|f_i^R - c\|^2, \tag{2}$$

where:

- N is the number of samples,
- f_i^R denotes the feature vector of real images output by MLP layer,
- c is the center of the real class, here we set it as origin.

By minimizing the volume of a data-enclosing hypersphere in feature embeddings of real images, the classifier can separate the real from the others more easily. The total loss function is defined as:

$$L = L_{\text{BCE}} + \lambda L_{\text{center}}, \tag{3}$$

where λ is a hyperparameter that balances the two loss terms which are in practice of different orders of magnitude. In all our experiments, we set λ as 0.01.

D. Proxy Negative Samples

Given the rapid development of new generative models, it is challenging to generalize classifiers to unseen models that may generate fake images with unknown distributions and using unknown techniques. In the meantime, natural images exhibit rather stable, common characteristics within their domain. A model trained to recognize these intrinsic characteristics (with pretraining on a very large number of natural images) has the potential to effectively distinguish real images from synthetic ones. To investigate the feasibility of training a detection model using only real images, we conducted preliminary experiments. However, training a model exclusively with samples from a single class is very difficult and frequently leads to "feature collapse" [36], wherein the model fails to learn discriminative features. Thus, incorporating negative samples (with same label as computer-generated images) is necessary to provide essential discriminatory information for the successful training of the detector.

We propose to use proxy data, constructed from real images, to simulate computer-generated images for training the model. These proxy images are expected to be close to the distribution of real images while introducing distinct discriminative features. Additionally, they do not have specific traces left by a particular generative model, thus this approach is expected to enhance generalization. During training, we use these proxy images as negative samples instead of computer-generated images, to calculate the BCE loss in Eq. (3).

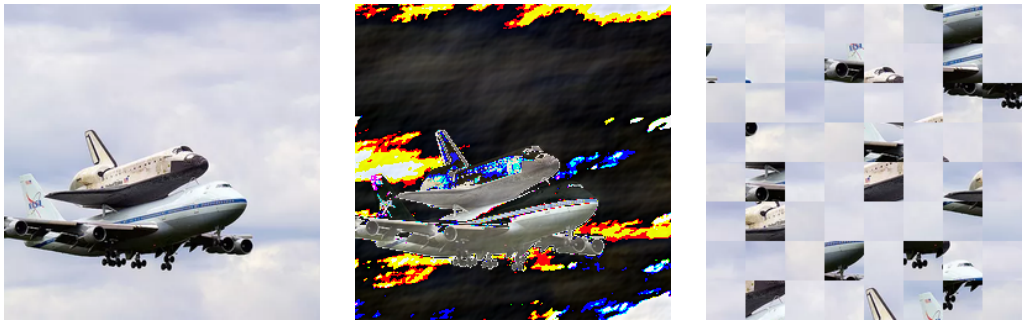
We experimented with several methods of constructing proxy negative samples, visualized in Figures 3, 4, and 5. The details of the proxy sample construction processes are as follows:

- **Frequency-domain Masking** (FM). Inspired by [37], We utilize the Fast Fourier Transform (FFT) to compute the frequency representation of an image, then divide the frequency bands into low, mid, and high regions to isolate the contributions of specific frequency components to the overall image features. By setting the frequencies to zero in specific ranges, we obtain masked representations. Masked images are then reconstructed by applying the inverse FFT. In our experiment, we randomly mask 30% of the low-band and high-band frequency range of the image, denoted as FM (low) and FM (high), respectively.
- **Mix-Up** (MU). New samples are created through linear interpolation of two images [38]. We randomly select two semantic categories of real images in the training set, then randomly choose one image from each category to perform the mix-up. Here, the mixing weight w is set to 0.5:

$$\tilde{x} = wx_i + (1-w)x_j. \tag{4}$$

- **Patch Shuffling** (PS). An image is divided into several smaller non-overlapping patches, which are then shuffled randomly to create a new, altered version of the original image. This method disrupts the spatial relationships between the patches while retaining the low-level features present within each patch. In our experiments, we set the patch size to 32.
- **Color Transfer** (CT). This method [39] changes the appearance of a source image to match the color pattern of a target image. The process involves the following steps:
  1) Convert the source and target images from RGB to a suitable color space (e.g., LAB).
  2) Compute the mean and standard deviation of each color channel for both images:

$$\mu_S = \frac{1}{N}\sum_{i=1}^{N} S_i, \quad \sigma_S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(S_i-\mu_S)^2}; \quad \mu_T = \frac{1}{N}\sum_{i=1}^{N} T_i, \quad \sigma_T = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i-\mu_T)^2}, \tag{5}$$



(a) Original image (b) Frequency Masking (c) Patch Shuffled

Fig. 3. Visualization of Frequency Masking and Patch Shuffling.

(a) Image A (b) Image B (c) Mix-Up image

Fig. 4. Visualization of Mix-Up.

(a) Source (b) Target (c) Color Transfer

Fig. 5. Visualization of Color Transfer.

where S_i and T_i are the pixel color channel values of the source and target images, respectively, and N is the number of pixels.

3) Adjust the source image's color channels to match the target image's statistics:

$$S_i' = \frac{\sigma_T}{\sigma_S}(S_i - \mu_S) + \mu_T, \tag{6}$$

where S_i' is the adjusted pixel color channel value.

4) Convert the adjusted image back to the RGB color space.

Similar to the mix-up method, we first randomly select semantic categories of source and target images, then randomly select source and target images within these categories.

IV. Experiments

A. Dataset and Evaluation Metrics

In previous works [9, 20], the training set comprised 720K images, evenly split between 360K real images from LSUN [40] and 360K fake images generated by ProGAN. To demonstrate the effectiveness and data-efficiency of our method, we randomly selected 4K LSUN real images from this dataset. Based on these real images, we generated 4K proxy images using

the transformation methods described in Section III-D. Thus, our training set consists of only 8K images, equally divided between real and proxy images.

For testing, we selected synthetic datasets from different domains as outlined in [20]. These domains include:

- **GANs**: ProGAN, CycleGAN, BigGAN, StyleGAN, GauGAN, StarGAN.
- **Diffusion models**: Latent Diffusion Model (LDM, 3 variants, see [20] for details), Guided Diffusion Model, GLIDE (3 variants, see [20] for details).
- **Autoregressive model**: DALL-E.
- **Facial manipulation with deep learning**: FaceForensics++.

We intentionally excluded the "Low-Level Vision" and "Perceptual Loss" categories from [20], which contain SITD, SAN, CRN, and IMLE. These images are either sourced from video games or pertain to image post-processing, and are not directly relevant to our task of detecting computer-generated images.

For comparability purposes, we assess the performance of detectors using average precision and classification accuracy as our evaluation metrics, consistent with previous approaches for AI-generated image detection [9, 10, 20]. When calculating classification accuracy, we use a fixed threshold of 0.5, which aligns with realistic scenarios where there is no prior information on the test data. Additionally, we report the average metric values across the test datasets to provide a comprehensive summary of the model's performance.

B. Different Methods for Constructing Proxy Negative Samples

We first evaluate the performance of our approach with different transformation methods for proxy sample construction. Tables II and III report the mean classification accuracy (ACC) and mean average precision (mAP) results on fifteen datasets, which are divided into different domains.

TABLE II
AVERAGE ACCURACY (IN %) OF DIFFERENT PROXY SAMPLE CONSTRUCTION METHODS. THE BEST RESULT IS HIGHLIGHTED IN BOLD.

Proxy sample construction method	GANs	Diffusion models	FaceForensics	DALL-E
PS	57.56	51.73	50.10	54.40
MU	84.46	62.99	58.50	67.40
FM (low)	**96.77**	80.23	**84.05**	98.50
FM (high)	95.51	83.75	81.30	**98.95**
CT	93.73	**89.17**	82.80	98.30

TABLE III
MEAN AVERAGE PRECISION (MAP) RESULTS (IN %) OF DIFFERENT PROXY SAMPLE CONSTRUCTION METHODS.

Proxy sample construction method	GANs	Diffusion models	FaceForensics	DALL-E
PS	96.77	76.16	73.84	91.07
MU	98.13	92.72	**94.07**	96.56
FM (low)	**99.76**	95.75	93.84	99.88
FM (high)	99.71	95.37	91.84	99.90
CT	99.01	**96.14**	91.25	**99.99**

Our findings indicate that, even without using computer-generated images for training, our approach achieves comparable detection accuracy across various domains. This is particularly evident when applying Frequency Masking and Color Transfer for proxy sample construction. Conversely, the performance of Patch Shuffling and Mix-Up methods is unsatisfactory, yielding rather acceptable mAP but low accuracy. We suspect that the distribution of proxy images generated using these methods is too divergent from that of real images. Consequently, the classifier tends to overfit to artifacts specific to these proxy images and often predicts most samples as "real", leading to suboptimal performance.

C. Comparison with State of the Art

We compared our approach with several state-of-the-art methods, which include: (i) CNNSpot [9], which fine-tunes a ResNet-50 on 720K real/ProGAN images; (ii) PatchForensics [14], which detects fingerprints in the frequency domain of images; (iii) CoOccurrence [8], which uses a combination of co-occurrence matrices and deep learning for classification; (iv) DIRE [10], which uses reconstruction errors of images for detection; (v) UniFD [20], which employs CLIP features followed by linear probing to separate real and fake images. All baselines are trained on the same dataset in a supervised manner, i.e., 720K real/ProGAN images from [9].

Tables IV and V present the mean classification accuracy (ACC) and mean average precision (mAP) results of the different methods, respectively. Compared to other baselines, our method, trained on only 4K real images and 4K proxy color transferred images, demonstrates superior performance, achieving an average ACC of 91.18% and an mAP of 97.22%.

TABLE IV
GENERALIZATION RESULTS (ACC, IN %) WITH COMPARISONS TO OTHER METHODS. THE BEST RESULT IS HIGHLIGHTED IN BOLD.

Method	Data source	Data scale	GANs						Face Forensics	Diffusion models							DALL-E	Total avg ACC
			Pro GAN	Cycle GAN	Big GAN	Style GAN	Gau GAN	Star GAN		LDM			Guided	GLIDE				
										200s	200s w/CFG	100s		100-27	50-27	100-10		
CNNSpot	LSUN / ProGAN	360K / 360K	99.99	85.20	70.20	85.70	78.95	91.70	53.47	54.03	54.96	54.14	60.07	60.78	63.80	65.66	55.58	68.95
Patch Forensics	LSUN / ProGAN	360K / 360K	75.03	68.97	68.47	79.16	64.23	63.94	75.54	76.50	76.10	75.77	67.41	74.81	73.28	68.52	67.91	71.71
Co-Occurrence	LSUN / ProGAN	360K / 360K	97.70	63.15	53.75	92.50	51.10	54.70	57.10	70.70	70.55	71.00	60.50	70.25	69.60	69.90	67.55	68.00
DIRE	LSUN / ProGAN	360K / 360K	**100.00**	67.73	64.78	83.08	65.30	**100.00**	**94.75**	82.70	84.05	84.25	**83.20**	87.10	90.80	**90.25**	58.75	82.45
UnivFD	LSUN / ProGAN	360K / 360K	**100.00**	**98.50**	**94.50**	82.00	**99.50**	97.00	66.60	94.19	73.76	94.36	70.03	79.07	79.85	78.14	86.78	86.28
Ours	LSUN / CT	4K / 4K	96.70	94.50	91.85	96.15	83.50	99.65	82.80	97.85	**97.40**	97.80	76.60	83.10	87.65	83.80	**98.30**	91.18
	LSUN / CT + FM	4K / 4K	99.15	93.50	89.90	**98.05**	80.50	99.60	85.00	**97.95**	**97.40**	**98.10**	81.10	**88.40**	**91.90**	87.20	98.10	**92.39**

TABLE V
GENERALIZATION RESULTS (AP, IN %) WITH COMPARISONS TO OTHER METHODS. THE BEST RESULT IS HIGHLIGHTED IN BOLD.

Method	Data source	Data scale	GANs						Face Forensics	Diffusion models							DALL-E	Total mAP
			Pro GAN	Cycle GAN	Big GAN	Style GAN	Gau GAN	Star GAN		LDM			Guided	GLIDE				
										200s	200s w/CFG	100s		100-27	50-27	100-10		
CNNSpot	LSUN / ProGAN	360K / 360K	**100.00**	93.47	84.50	99.54	89.49	98.15	89.02	70.62	71.00	70.54	73.72	80.65	84.91	82.07	70.59	89.02
Patch Forensics	LSUN / ProGAN	360K / 360K	80.88	72.84	71.66	85.75	65.99	69.25	76.55	87.10	86.72	86.40	75.03	85.37	83.73	78.38	75.67	78.75
Co-Occurrence	LSUN / ProGAN	360K / 360K	99.74	80.95	50.61	98.63	53.11	67.99	59.14	91.21	89.02	92.39	70.20	89.32	88.35	82.79	80.96	79.63
DIRE	LSUN / ProGAN	360K / 360K	**100.00**	76.73	72.80	97.06	68.44	**100.00**	**98.55**	95.17	95.43	95.77	**94.29**	96.18	97.30	**97.53**	68.73	90.26
UnivFD	LSUN / ProGAN	360K / 360K	**100.00**	99.46	**99.59**	97.24	**99.98**	99.60	82.45	99.14	92.15	99.17	87.77	94.74	95.34	94.57	97.15	95.89
Ours	LSUN / CT	4K / 4K	99.86	**100.00**	97.42	99.29	97.48	**100.00**	91.25	99.76	99.57	99.77	86.07	95.28	96.80	95.75	**99.99**	97.22
	LSUN / CT + FM	4K / 4K	99.99	**100.00**	99.48	**99.77**	99.96	**100.00**	94.75	**99.90**	**99.64**	**99.91**	89.93	**96.55**	**97.60**	96.36	**99.99**	**98.25**

We observe that while methods trained on real/ProGAN images can perform well within the GAN domain, they may fail to generalize to other domains. In contrast, our method exhibits better generalization across different domains. This improved generalization is due to our use of proxy images as negative samples, which allows our model to learn the distinguishing features of real images more effectively, making it less sensitive to the specific domain of the fake images.

To further enhance our method, we combined different types of proxy images to augment the distribution of negative samples. Based on the earlier experiment, we selected the two best-performing types of proxy images: Color Transfer (CT) and Frequency Masking (FM). Specifically, we randomly selected 2K CT images and 2K FM images (low) as negative samples to train the model. Compared to using only CT images as negative samples, this combination further improved performance, with the average ACC rising from 91.18% to 92.39% and the mAP increasing from 97.22% to 98.25%. These results verify the effectiveness of our approach and inspire us to further explore the potential of other methods for generating proxy samples.

D. Effect of Center Loss

We conducted ablation studies to evaluate the effectiveness of applying center loss on the embeddings of real images. Table VI shows the average detection accuracy on the test set, where "w/o Center loss" refers to training the model without center loss. The results demonstrate the positive impact of this additional training constraint on real image features. When training with CT, the accuracy improved by 2.94% after applying center loss; when training with CT + FM, center loss enhanced the accuracy by 0.64%.

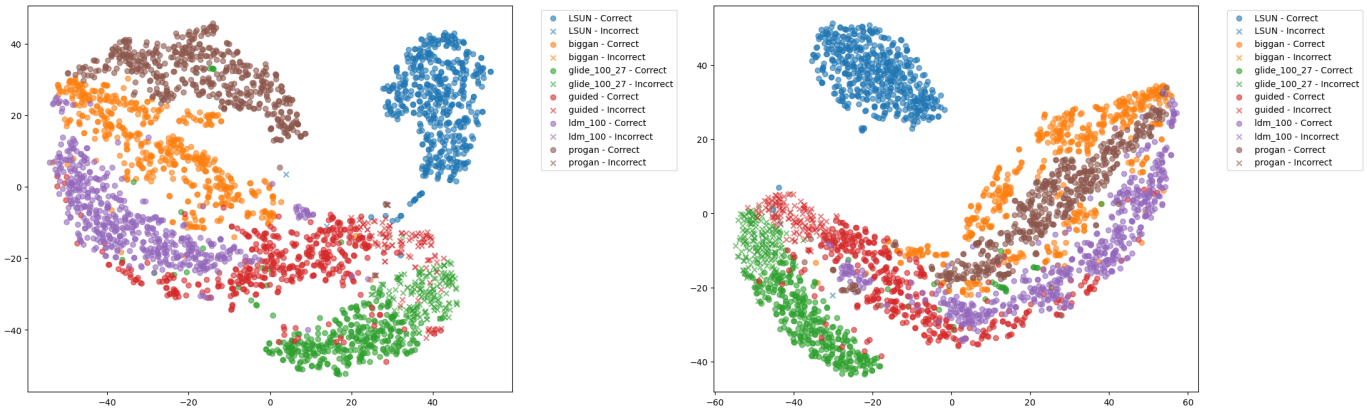
Additionally, Fig. 6 presents the t-SNE visualization of features from different test sets, output from the MLP layer of the model. The visualization shows that, after applying center loss, the features of real images (shown in blue) are more closely clustered, thus enhancing their discriminative power.

V. CONCLUSION AND OUTLOOK

In this paper, we proposed a data-efficient method for detecting computer-generated images using large pretrained models, specifically leveraging the CLIP image encoder. To the best of our knowledge, our approach demonstrates for the first time

TABLE VI
AVERAGE DETECTION ACCURACY (IN %) ON TEST SET.

Proxy samples	w/o center loss	with center loss
CT	88.24	91.18
CT + FM	91.75	92.39



(a) w/o center loss (b) with center loss

Fig. 6. t-SNE visualization of different test sets using the feature space output by the MLP layer of our model. Circle means that the sample is predicted correctly, while cross stands for the wrong prediction.

in the literature that it is feasible to achieve high detection accuracy by using during training only a small set of real images and proxy counterparts constructed from real images, thus avoiding the need for a large dataset of computer-generated images. This is interesting and significant from both an academic research and practical utility perspective. Technically, we explored various construction methods to simulate negative samples and found that frequency masking and color transfer methods were particularly effective. Combining these methods further improved our model's performance, underscoring the potential of diverse proxy samples in enhancing generalization capability of fake image detection.

We conducted experiments to compare our approach with state-of-the-art methods and demonstrated superior performance in terms of both mean average precision and average classification accuracy. Our method's robustness across different domains of synthetic image generation, including GANs, diffusion models, and autoregressive models, highlights its generalization capability. Additionally, the application of center loss on the embeddings of real images further improved feature discrimination, as evidenced by the t-SNE visualizations and enhanced accuracy metrics.

The success of our model in learning the intrinsic properties of real images, rather than overfitting to specific generative model artifacts, points to a promising direction for future research. In particular, our findings suggest that further exploration of proxy sample construction techniques and their combinations could yield even more robust detection frameworks. Understanding the varying performances of different proxy sample construction methods will be essential in refining these techniques. Additionally, a deeper comprehension of the features learned by the CLIP model and how they contribute to distinguishing real images from synthetic ones will provide valuable insights for improving model performance.

In conclusion, our work presents a significant step towards more efficient and generalizable computer-generated image detection, offering a viable solution in scenarios with limited access to extensive labeled datasets. Future research shall focus on refining proxy image generation methods, exploring additional loss functions, and extending the approach to handling more diverse forms of digital content, such as the detection of fake videos [41] and other multimedia forgeries.

ACKNOWLEDGMENTS

This work is partially funded by the French National Research Agency (Grants numbers ANR-23-IAS4-0004-02 and ANR-15-IDEX-02). The authors would like to thank Dr. Patrick Bas and Dr. Georges Quénot for helpful discussions.

REFERENCES

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in Neural Information Processing Systems* **27**, 2672–2680 (2014).

[2] Ho, J., Jain, A., and Abbeel, P., "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems* **33**, 6840–6851 (2020).

[3] Piva, A., "An overview on image forensics," *International Scholarly Research Notices* **2013**(1), 496701 (2013).

[4] Verdoliva, L., "Media forensics and deepfakes: an overview," *IEEE Journal of Selected Topics in Signal Processing* **14**(5), 910–932 (2020).

[5] Castillo Camacho, I. and Wang, K., "A comprehensive review of deep-learning-based methods for image forensics," *Journal of Imaging* **7**(4), 69 (2021).

[6] Quan, W., Wang, K., Yan, D.-M., and Zhang, X., "Distinguishing between natural and computer-generated images using convolutional neural networks," *IEEE Transactions on Information Forensics and Security* **13**(11), 2772–2787 (2018).

[7] Bai, W., Zhang, Z., Li, B., Wang, P., Li, Y., Zhang, C., and Hu, W., "Robust texture-aware computer-generated image forensic: Benchmark and algorithm," *IEEE Transactions on Image Processing* **30**, 8439–8453 (2021).

[8] Nataraj, L., Mohammed, T. M., Chandrasekaran, S., Flenner, A., Bappy, J. H., Roy-Chowdhury, A. K., and Manjunath, B., "Detecting GAN generated fake images using co-occurrence matrices," *arXiv preprint arXiv:1903.06836* (2019).

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

[10] Wang, Z., Bao, J., Zhou, W., Wang, W., Hu, H., Chen, H., and Li, H., "DIRE for diffusion-generated image detection," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 22445–22455 (2023).

[11] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., "Learning transferable visual models from natural language supervision," in [*Proceedings of the International Conference on Machine Learning*], 8748–8763, PMLR (2021).

[12] Dzanic, T., Shah, K., and Witherden, F., "Fourier spectrum discrepancies in deep network generated images," *Advances in Neural Information Processing Systems* **33**, 3022–3032 (2020).

[13] Jeong, Y., Kim, D., Ro, Y., Kim, P., and Choi, J., "Fingerprintnet: Synthesized fingerprints for generated image detection," in [*Proceedings of the European Conference on Computer Vision*], 76–94, Springer (2022).

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

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

[16] Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., and Verdoliva, L., "On the detection of synthetic images generated by diffusion models," in [*Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*], 1–5, IEEE (2023).

[17] Chandrasegaran, K., Tran, N.-T., and Cheung, N.-M., "A closer look at Fourier spectrum discrepancies for CNN-generated images detection," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 7200–7209 (2021).

[18] Liu, B., Yang, F., Bi, X., Xiao, B., Li, W., and Gao, X., "Detecting generated images by real images," in [*Proceedings of the European Conference on Computer Vision*], 95–110, Springer (2022).

[19] Chai, L., Bau, D., Lim, S.-N., and Isola, P., "What makes fake images detectable? understanding properties that generalize," in [*Proceedings of the European Conference on Computer Vision*], 103–120, Springer (2020).

[20] Ojha, U., Li, Y., and Lee, Y. J., "Towards universal fake image detectors that generalize across generative models," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 24480–24489 (2023).

[21] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q., "A comprehensive survey on transfer learning," *Proceedings of the IEEE* **109**(1), 43–76 (2020).

[22] Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K., "How much can CLIP benefit vision-and-language tasks?," *arXiv preprint arXiv:2107.06383* (2021).

[23] Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., and Li, H., "Tip-adapter: Training-free adaption of CLIP for few-shot classification," in [*Proceedings of the European Conference on Computer Vision*], 493–510, Springer (2022).

[24] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929* (2020).

[25] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B., "Swin transformer: Hierarchical vision transformer using shifted windows," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 10012–10022 (2021).

[26] Bao, H., Dong, L., Piao, S., and Wei, F., "BEiT: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254* (2021).

[27] Singh, M., Gustafson, L., Adcock, A., de Freitas Reis, V., Gedik, B., Kosaraju, R. P., Mahajan, D., Girshick, R., Dollár, P., and Van Der Maaten, L., "Revisiting weakly supervised pre-training of visual perception models," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 804–814 (2022).

[28] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R., "Masked autoencoders are scalable vision learners," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 16000–16009 (2022).

[29] Singh, M., Duval, Q., Alwala, K. V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer, C., Girshick, R., et al., "The effectiveness of MAE pre-pretraining for billion-scale pretraining," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 5484–5494 (2023).

[30] Lin, Z., Geng, S., Zhang, R., Gao, P., De Melo, G., Wang, X., Dai, J., Qiao, Y., and Li, H., "Frozen CLIP models are efficient video learners," in [*European Conference on Computer Vision*], 388–404, Springer (2022).

[31] Barraco, M., Cornia, M., Cascianelli, S., Baraldi, L., and Cucchiara, R., "The unreasonable effectiveness of CLIP features for image captioning: an experimental analysis," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 4662–4670 (2022).

[32] Huang, T., Dong, B., Yang, Y., Huang, X., Lau, R. W., Ouyang, W., and Zuo, W., "CLIP2Point: Transfer CLIP to point cloud classification with image-depth pre-training," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 22157–22167 (2023).

[33] Koutlis, C. and Papadopoulos, S., "Leveraging representations from intermediate encoder-blocks for synthetic image detection," *arXiv preprint arXiv:2402.19091* (2024).

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., "Attention is all you need," *Advances in Neural Information Processing Systems* **30** (2017).

[35] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M., "Deep one-class classification," in [*Proceedings of the International Conference on Machine Learning*], 4393–4402, PMLR (2018).

[36] Perera, P. and Patel, V. M., "Learning deep features for one-class classification," *IEEE Transactions on Image Processing* **28**(11), 5450–5463 (2019).

[37] Doloriel, C. T. and Cheung, N.-M., "Frequency masking for universal deepfake detection," in [*Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*], 13466–13470, IEEE (2024).

[38] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412* (2017).

[39] Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P., "Color transfer between images," *IEEE Computer Graphics and Applications* **21**(5), 34–41 (2001).

[40] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J., "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365* (2015).

[41] Saealal, M. S., Ibrahim, M. Z., Yakno, M., and Arshad, N. W., "Three-dimensional convolutional approaches for the verification of deepfake videos: The effect of image depth size on authentication performance," *Journal of Advances in Information Technology* **14**(3), 488–494 (2023).