Team NYCU at Defactify4: Robust Detection and Source Identification of AI-Generated Images Using CNN and CLIP-Based Models

Tsan-Tsung, Yang[1,*,†], I-Wei, Chen[2,†], Kuan-Ting, Chen[1,†], Shang-Hsuan, Chiang[1,‡] and Wen-Chih, Peng[1,‡]

[1]Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Department of Electronics and Electrical Engineering , National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Abstract

With the rapid advancement of generative AI, AI-generated images have become increasingly realistic, raising concerns about creativity, misinformation, and content authenticity. Detecting such images and identifying their source models has become a critical challenge in ensuring the integrity of digital media. This paper tackles the detection of AI-generated images and identifying their source models using CNN and CLIP-ViT classifiers. For the CNN-based classifier, we leverage EfficientNet-B0 as the backbone and feed with RGB channels, frequency features, and reconstruction errors, while for CLIP-ViT, we adopt a pretrained CLIP image encoder to extract image features and SVM to perform classification. Evaluated on the Defactify 4 dataset, our methods demonstrate strong performance in both tasks, with CLIP-ViT showing superior robustness to image perturbations. Compared to baselines like AEROBLADE and OCC-CLIP, our approach achieves competitive results. Notably, our method ranked Top-3 overall in the Defactify 4 competition, highlighting its effectiveness and generalizability. All of our implementations can be found in https://github.com/uuugaga/Defactify_4

Keywords

AI-Generated Images, Source Model Identification, CNN and CLIP Models, Robust Detection

1. Introduction

Recent advancements in text-to-image generation models have made it possible to produce high-quality images from simple prompts. This development poses challenges to content creators and raises concerns about the authenticity of online content. Over the years, various generative models, including Generative Adversarial Networks (GANs) [1], Variational Autoencoders (VAEs) [2], Stable Diffusion [3], DALL·E [4], and Midjourney [5], have emerged and continuously improved.

To effectively understand and regulate AI-generated images, it is crucial not only to detect whether the content is real or fake but also to identify the specific model used to generate it. However, research efforts on the identification of the source model remain limited. The Defactify 4 workshop dataset [6], which includes state-of-the-art text-to-image models such as Stable Diffusion, DALL·E, and Midjourney, addresses this gap by offering a benchmark for two tasks: (A) classifying AI-generated content and (B) identifying the source models. The dataset also accounts for real-world scenarios where images are often modified differently. To emphasize the generalizability of the detection methods, this dataset also put some perturbations on the generated images.

In this paper, we adopt two primary approaches—CNN-based and CLIP-based classifiers—to evaluate their performance on both tasks. Additionally, we conduct comprehensive experiments, including baseline comparisons, robustness evaluations under perturbations, and ablation studies on data augmentation. Our findings can be summarized as follows:

- Both CLIP-ViT and CNN-based methods effectively detect AI-generated content and identify the source model. However, CLIP-ViT demonstrates superior robustness when images are subjected to perturbations in real-world scenarios.
- Our methods achieve competitive performance or even better compared to strong baselines, including AEROBLADE [1] and OCC-CLIP [2].
- Ablation studies reveal that applying perturbations—such as Gaussian noise, JPEG compression, brightness reduction, and Gaussian blurring—during training significantly improves the models' generalization ability.

These results underscore the importance of robust model design and data augmentation for detecting and identifying AI-generated images in practical applications.

2. Related Works

2.1. Revolution on Image Generation

The landscape of image generation has undergone a dramatic transformation with the rise of deep learning techniques. Initially, Generative Adversarial Networks (GANs) [1] revolutionized the field by introducing a two-network architecture—a generator and a discriminator—that enables the production of realistic images through adversarial training. Another milestone in this evolution was the development of Variational Autoencoders (VAEs) [2], which employed a probabilistic approach to generative modeling. VAEs utilized a likelihood-based objective to learn a lower-dimensional latent space representation of input data.

More recently, the focus has shifted towards diffusion models, which have demonstrated significant improvements in image quality and authenticity. Unlike GANs and VAEs, which rely on adversarial learning and latent space encoding, respectively, Denoising Diffusion Probabilistic Models

AAAI-25 Defactify 4 Workshop, Feb 2025, Philadelphia, Pennsylvania, USA
*Corresponding author.
†These authors contributed equally.
alexyang0826@hotmail.com (T. Yang); ken12300326@gmail.com (I. Chen); larrybrown901120@gmail.com (K. Chen); andy10801@gmail.com (S. Chiang); wcpeng@cs.nycu.edu.tw (W. Peng)

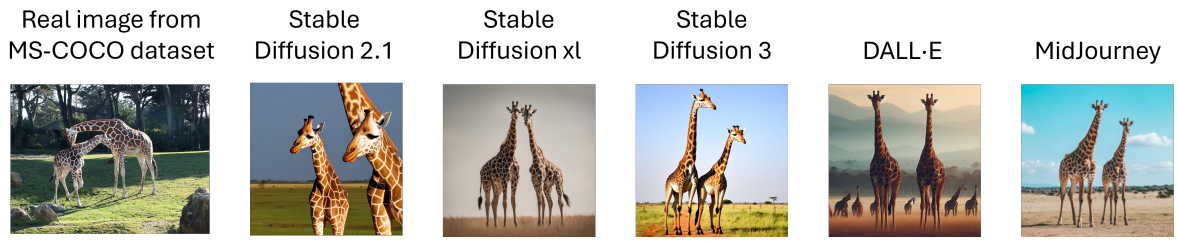

Figure 1: Given the same prompt, "Two tall giraffes standing next to each other on a field", the different models can generate different image content and still keep the text consistency. The left one is the real image, while the others are all generated by text-to-image models. It is clear that each model has its own "assumption" and "style" on the given prompt, which can be captured as the model's features.

(DDPMs) [3] aim to learn a denoising process. Starting with pure noise, DDPMs iteratively refine it through a series of denoising steps, transforming random noise into coherent, meaningful images. Remarkably, DDPMs can generate images with greater diversity and fewer artifacts than GANs and VAEs, often producing more realistic and visually appealing results.

Building on these advancements, the field has also seen a rise in text-to-image generation, with models such as DALL·E [4], Imagen [5], Stable Diffusion [6], Midjourney [7] enabling users to generate detailed images from simple textual prompts. These models leverage pre-trained text encoders and VAEs to map both text and images into a shared latent space, where they then perform a diffusion process on the latent representations rather than on the original images. These models can now generate images that are visually compelling and contextually relevant to the prompts, often producing results that are remarkably close to the user's original vision.

2.2. Classifying AI-Generated Images

Image generation is a double-edged sword: it offers excellent potential across various fields, from creative industries to healthcare research, yet it also facilitates the propagation of misinformation on social media. This underscores the importance of detecting AI-generated content to preserve the integrity of information.

One intuitive approach is to learn a binary classifier, which directly detects whether the image is real or fake. Marra et al. [8] investigate various convolutional neural network (CNN)-based models, for identifying GAN-generated content. Their findings suggest that CNN-based models effectively detect images generated by GANs. Cozzolino et al. [9] applied CLIP [10] to encode image captions as the fake content and train the SVM classifier to detect the AI-generated image. They demonstrated that CLIP features provide excellent generalization, achieving strong performance even with a limited number of examples. Instead of directly training, Alam et al. [11] aimed to add more extra feature to the classifier. They converted RGB channels into YCbCr channels and applied Spatial Fourier Transformation to capture spatial shifts. By combining deep neural networks with feature fusion, their method outperforms existing state-of-the-art techniques.

Other researchers assume that AI-generated content can be easily reconstructed by AI-modules. Wang et al. [12] found that DM images can be approximately reconstructed by the diffusion model, but real images cannot. The difference between the reconstructed image and the original image is recorded as Diffusion Reconstruction Error (DIRE). Then DIRE can be used as a feature for training to determine whether it is real or fake, and the generalization will be much higher. Ricker et al. [13] propose the AEROBLADE framework, performing reconstruction on auto-encoders. They use VGG network's hidden layer to measure the reconstruction distance, which is called LPIPS_2 distance. Their experiment result is awesome without any training process.

2.3. Source Model Identification

To effectively understand, regulate, and categorize AI-generated content, it is essential to identify the model used to generate the image accurately. However, current research primarily focuses on distinguishing between real and AI-generated images. For source model identification, some approaches require direct access to or modifications of the source model. This is not practical in real-world scenarios since some generative models are not publicly accessible, like MidJourney and DALL·E. The most recent work on source model identification, OCC-CLIP [14], modifies the problem into a few-shot learning setting, achieving good performance and scalability to larger datasets. By combining prompt learning with adversarial augmentation, OCC-CLIP keeps the original CLIP [10] parameters fixed and tunes only a learnable context. Additionally, it employs multiple one-class classifiers to predict multi-class targets, yielding excellent results in both binary and multi-class classification tasks.

3. Method

3.1. Problem Formulation

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ denote the image-label pairs in the dataset, where the i-th sample consists of an image x_i and a corresponding label y_i. For task A, the goal is to detect whether the image is AI-generated or not. In this case, y_i is a binary label, where $y_i \in \{\text{real}, \text{fake}\}$. In contrast, task B is more challenging, as it requires identifying the source model of the image. Therefore, y_i now represents the source model label; that is, $y_i \in \{\text{real}, \text{SD}_{2.1}, \text{SD}_{xl}, \text{SD}_3, \text{DALL·E}, \text{MidJourney}\}$, where SD is an abbreviation of Stable Diffusion and the subscript is the model's version.

Formally, the goal is to learn a decision function $f_\theta(x)$ that minimizes the classification error. It can be formulated as:

$$\theta^* = \arg\min_{\theta} \mathbb{E}\left[\mathcal{L}(f_\theta(x), y)\right] \tag{1}$$

where $\mathcal{L}$ is a error measurement of the prediction and ground truth and θ is the model parameters.

3.2. CNN or CLIP-ViT

To explore the effectiveness of CNN-based and CLIP-based methods on AI-generated image detection, we select EfficientNet-B0 [15] and CLIP-ViT [10] to compare their performance.

3.2.1. EfficientNet-B0

EfficientNet-B0 [15] is a convolutional neural network (CNN) that balances accuracy and efficiency by using a compound scaling method. This approach uniformly scales the network's depth, width, and resolution, enabling high performance with relatively fewer parameters compared to traditional CNNs.

To enrich the model with more detailed information, we adopt feature augmentation strategies inspired by Alam et al. [11] and Ricker et al. [13]. Specifically, we incorporate additional input features—frequency information and reconstruction error—alongside the standard RGB image. Formally, we can denote the original RGB image as I, the reconstruction error as E and the frequency domain feature as F, where $I \in \mathbb{R}^{h\times w\times 3}$, $E, F \in \mathbb{R}^{h\times w\times 1}$ and h, w represent the image's height and width. By concatenating these features along the channel dimension, the final input to the model becomes:

$$X = \text{concat}(I, E, F) \in \mathbb{R}^{h\times w\times 5} \tag{2}$$

where concat$(\cdot)$ denotes channel-wise concatenation.

For EfficientNet-B0 method, we adopt the cross-entropy loss to find the approximately solution of θ^*, the original classification error $\mathcal{L}$ from equation 1 can be rewritten as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_i = c)\log p(f_\theta(X_i) = c|X_i, \theta) \tag{3}$$

where X_i is the augmented input for the i-th sample, $\mathbb{I}(\cdot)$ is the indicator function and $f_\theta(X_i)$ is the model output given the input X_i.

3.2.2. CLIP-ViT

CLIP-ViT [10] combines the power of Vision Transformers (ViT) with a contrastive pretraining framework. Designed for multi-modal learning, CLIP aligns images and text in a shared latent space, enabling robust zero-shot and transfer learning capabilities.

In our approach, we utilize the pretrained CLIP-ViT model without prompt for image feature extraction. Given an input image x_i, the CLIP encoder f_{CLIP} maps RGB images into a high-dimensional feature space with feature size d:

$$z_i = f_{\text{CLIP}}(X_i) \in \mathbb{R}^d \tag{4}$$

For classification, we apply a Support Vector Machine (SVM) [16] with a Radial Basis Function (RBF) kernel to separate different classes. The decision function of the SVM is defined as:

$$f_\theta(z) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K(z, z_i) + b\right) \tag{5}$$

where α_i are the Lagrange multipliers, $K(z, z_i) = \exp\left(-\gamma\cdot\|z - z_i\|^2\right)$ is the RBF kernel, γ controls the kernel's spread, b is the bias term.

The SVM optimization problem aims to maximize the margin while allowing soft-margin classification, the equation 1 can be written as:

$$\mathcal{L}(\theta) = \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{N}\xi_i \tag{6}$$

subject to:

$$y_i\cdot f_\theta(z_i) \geq 1 - \xi_i,\ \ \xi_i \geq 0 \tag{7}$$

where C balances margin maximization and classification error and ξ_i are slack variables for misclassification tolerance. For multi-class classification, we employ the One-vs-Rest (OvR) strategy, where a separate binary SVM classifier is trained for each class c against all other classes

4. Experiments

4.1. Setup

4.1.1. Data

We utilize the dataset provided in this shared task [17] [18] to evaluate the performance of our method. The dataset consists of four parts: training, validation, testing, and final testing. The first three parts are standard splits of the dataset, while the final testing part introduces perturbations to the original images. Detailed statistics for all four parts can be found in Table 1, and the training set is a balanced dataset, with each label containing exactly 7000 samples. Fig 1 gives an example of this dataset.

To improve our model's robustness, we apply the following data augmentation techniques to the original training dataset:

- **Compression**: Considering real-world scenarios where images are often compressed in JPEG format, we apply JPEG compression with a quality parameter of 50.
- **Blurring**: We apply Gaussian blurring with a sigma value of 5 and a kernel size of (5, 5).
- **Noise Perturbation**: We add small Gaussian noise to the original image with a standard deviation of 0.3.
- **Brightness Transformation**: We adjust the image brightness using a brightness factor of 0.5.

Table 1
Statistics of the shared task dataset.

Split Name	Number of Samples
Training	42000
Validation	9000
Testing	9000
Final Testing	45000

4.1.2. Implementation Details

For the EfficientNet method, all RGB images are resized to $(512, 512, 3)$ and further processed to extract additional

features: frequency and reconstruction error. The frequency feature is obtained by converting the image to grayscale, applying a 2D Fast Fourier Transform (FFT), shifting the zero-frequency component to the center, and computing the logarithmic magnitude spectrum. To calculate the reconstruction error, we use the VAE from the `runwayml/stable-diffusion-v1-5` model to reconstruct each image and compute the pixel-wise absolute difference between the reconstructed and original images. These two features are concatenated with the original RGB channels, resulting in a five-channel input of size $(512, 512, 5)$. The classifier is trained from scratch using the EfficientNet-B0 backbone with the Adam optimizer, a learning rate of 1×10^{-4}, and a total of 30 epochs.

For CLIP-ViT, we select the pre-trained model `openai/clip-vit-base-patch16` to extract the image features. After that, we trained a SVM classifier based on the image features. We performed a grid search over the following hyperparameters: $c \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. The final predictions are based on the best-performing hyperparameter set identified through grid search.

Our experiments were conducted on a machine equipped with 48 AMD Ryzen Threadripper 3960X 24-Core Processors, 237GB of RAM and 2 NVIDIA GeForce RTX 3090 GPUs. The source code is publicly available at: https://github.com/uuugaga/Defactify_4

4.2. Evaluation

In these two tasks, we select both accuracy and macro-f1 score as our metrics in both tasks. Since task A is a little bit imbalanced on real images, macro-f1 score is much more reliable on these two tasks.

4.2.1. Comparison to Baselines

We compare our method with some state-of-the-art baselines by modifying the source code released by the original authors. For both tasks, we select AEROBLADE [13] and OCC-CLIP [14] as our baseline methods due to their simplicity and effectiveness. Despite their relatively straightforward approaches, these methods have shown strong performance in real-world scenarios.

AEROBLADE [13] applies three types of pre-trained autoencoders for reconstruction. It assumes that the reconstruction distance for AI-generated content differs from that of real images. For our implementation of AEROBLADE, we follow the parameter settings from the original authors, with modifications to the batch size (set to 16) and the image resolution (set to 512 × 512). Additionally, we adopt the original experimental results and choose the LPIPS_2 distance as the metric for reconstruction distance. Using the training set, we plot a kernel density estimation of the LPIPS_2 distance, as shown in Fig. 2. We set the distance threshold to -0.035 to classify the images as either real or fake. Since AEROBLADE in the original paper is only suitable for binary classification, we only compare with it in task A.

OCC-CLIP [14] is another baseline. It freezes the CLIP model's weight by only tuning the learnable context. It combines both prompt learning and adversarial augmentation in a few-shot source model identification. We alter the context length to 32 and follow the original setting as a multiple two-class classifiers. In this case, We train 5 classifiers to detect whether it is real or one of the target models, and each of them is trained on 7000 real images and 7000 target fake images. The final prediction is followed by the authors of OCC-CLIP:

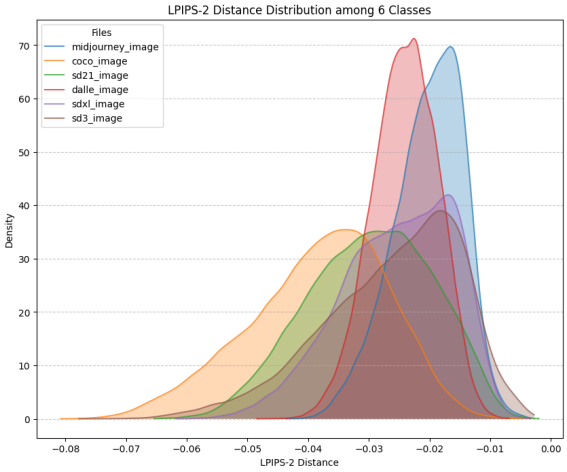

Figure 2: The LPIPS_2 distance distribution of generated content from different models. It's obvious that the distribution has a lot of overlapped areas, which, in turn, indicates that AEROBLADE's performance would not be great.

$$\hat{y}_i = \begin{cases} j, & \max\limits_{j \in 1,\ldots,5} p(\hat{y}_i = 1; x_i, \theta_j) > 0.5 \\ 0, & otherwise. \end{cases} \tag{8}$$

where θ_j is the j-th OCC-CLIP for classifying the content is real or generated from model j. It means that the final prediction is made by selecting the target model with the highest probability, or classifying it as real if none of the classifiers exceed the threshold 0.5.

Table 2 shows that our method outperforms AEROBLADE and keeps almost the same performance as OCC-CLIP in task A. However, in task B, our method surpasses OCC-CLIP by approximately 0.12 in accuracy and macro-f1, which corresponds to a **14%** improvement.

Table 2
Compare our methods with AEROBLADE and OCC-CLIP in tasks A and B on the validation set, with accuracy and macro-f1 score. The bold font highlights the best performance, while the underlined font indicates the second-best result. Both of our methods achieve competitive results on Task A and outperform all other baseline methods on Task B.

Method	Task A		Task B	
	Acc.	F1	Acc.	F1
AEROBLADE	0.8149	0.6986	-	-
OCC-CLIP	**0.9934**	**0.9881**	0.8693	0.8721
Ours: EfficientNet-B0	<u>0.9849</u>	<u>0.9833</u>	**0.9951**	**0.9951**
Ours: CLIP-ViT	0.9421	0.9421	<u>0.9377</u>	<u>0.9317</u>

4.2.2. Robustness to Perturbations

In real-world scenarios, images are often subjected to various processing steps, such as compression, noise addition, brightness adjustments, or blurring, especially during online sharing or editing. The results presented in

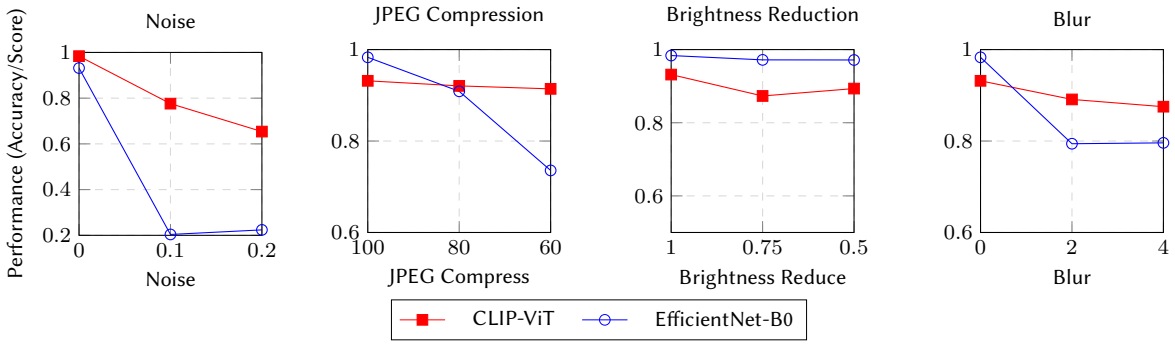

Figure 3: The generalization of CLIP-ViT and EfficientNet on different perturbations. The red square line indicates the CLIP-ViT and the blue circle one indicates EfficientNet. The result shows that CLIP-ViT's performance is better while EfficientNet's performance drops dramatically.

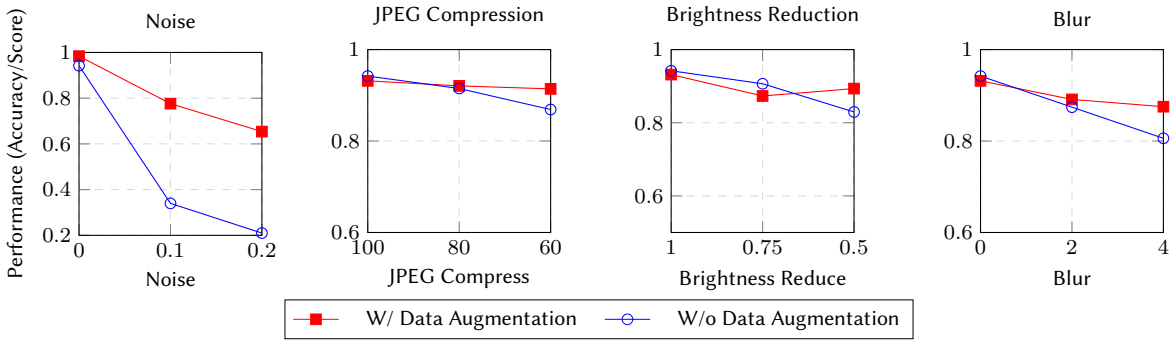

Figure 4: The importance of data augmentation. The red square line indicates training with data augmentation and the blue circle one indicates training without data augmentation.

Figure 3 highlight the robustness of models when faced with such perturbations. Across all experiments, the CLIP-ViT backbone model consistently demonstrated superior performance compared to the EfficientNet backbone. For instance, in the noise test (the first sub-figure), CLIP-ViT maintained high accuracy even with increasing noise levels, while EfficientNet suffered a significant performance degradation. Similarly, in JPEG compression (the second sub-figure) and brightness reduction (the third sub-figure), CLIP-ViT achieved more stable accuracy across different perturbation intensities. The blur test (the last sub-figure) further underscores CLIP-ViT's robustness, maintaining competitive performance even under severe blurring conditions. These results emphasize the importance of designing robust models capable of handling real-world image variations effectively.

4.2.3. Ablation Study: Importance of Data Augmentation

The results of the ablation study, as illustrated in Figure 4, demonstrate the significant impact of data enhancement on model performance under various perturbations. For example, in the noise experiment (the first sub-figure), models with data augmentation consistently outperformed those without, achieving much higher accuracy across all noise levels. Similarly, in the JPEG compression test (the second sub-figure), data-augmented models exhibited superior performance, particularly at higher compression levels. Brightness reduction (the third sub-figure) and blur experiments (Figure the last sub-figure) also highlighted the robustness introduced by data augmentation, with models showing improved accuracy and stability when augmentation techniques were employed. These findings underline the critical role of data augmentation in mitigating performance degradation caused by common perturbations, thereby enhancing the model's generalizability and robustness in real-world scenarios.

5. Conclusion

This paper evaluated CNN-based and CLIP-based methods for detecting AI-generated images and identifying their source models using the Defactify 4 workshop dataset. Our key findings include:

- Both EfficientNet-B0 and CLIP-ViT models perform well in detection and source identification, with CLIP-ViT showing greater robustness against real-world image degradations.
- Our methods achieve competitive or superior results compared to baselines like AEROBLADE and OCC-CLIP, especially in source model identification.
- Data augmentation with perturbations (e.g., Gaussian noise, JPEG compression) significantly improves model generalization and robustness.

These results highlight the effectiveness of CNN and CLIP encoders in AI-generated content detection. Future work will focus on enhancing feature interpretability for source model attribution.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[2] D. P. Kingma, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[3] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[4] OpenAI, Dalle-3, 2024. Https://openai.com/index/dall-e-3/.

[5] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al., Photorealistic text-to-image diffusion models with deep language understanding, Advances in neural information processing systems 35 (2022) 36479–36494.

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684–10695.

[7] Midjourney, 2024. Https://www.midjourney.com/.

[8] F. Marra, D. Gragnaniello, D. Cozzolino, L. Verdoliva, Detection of gan-generated fake images over social networks, in: 2018 IEEE conference on multimedia information processing and retrieval (MIPR), IEEE, 2018, pp. 384–389.

[9] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, L. Verdoliva, Raising the bar of ai-generated image detection with clip, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4356–4366.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.

[11] I. Alam, M. S. Muneer, S. S. Woo, Ugad: Universal generative ai detector utilizing frequency fingerprints, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 4332–4340.

[12] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, H. Li, Dire for diffusion-generated image detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 22445–22455.

[13] J. Ricker, D. Lukovnikov, A. Fischer, Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 9130–9140.

[14] F. Liu, H. Luo, Y. Li, P. Torr, J. Gu, Which model generated this image? a model-agnostic approach for origin attribution, 2024. arXiv:2404.02697v2.

[15] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[16] C. Cortes, Support-vector networks, Machine Learning (1995).

[17] R. Roy, A. Aziz, S. Bajpai, N. Imanpour, G. Singh, S. Biswas, K. Wanaskar, P. Patwa, S. Ghosh, S. Dixit, N. R. Pal, V. Rawte, R. Garimella, A. Das, A. Sheth, V. Sharma, A. N. Reganti, V. Jain, A. Chadha, Defactify-image: A comprehensive dataset for human vs. ai generated image detection, in: proceedings of DeFactify 4: Fourth workshop on Multimodal Fact-Checking and Hate Speech Detection, CEUR, 2025.

[18] R. Roy, N. Imanpour, A. Aziz, S. Bajpai, G. Singh, S. Biswas, K. Wanaskar, P. Patwa, S. Ghosh, S. Dixit, N. R. Pal, V. Rawte, R. Garimella, A. Das, A. Sheth, V. Sharma, A. N. Reganti, V. Jain, A. Chadha, Overview of image counter turing test: Ai generated image detection, in: proceedings of DeFactify 4: Fourth workshop on Multimodal Fact-Checking and Hate Speech Detection, CEUR, 2025.