Training-free Source Attribution of AI-generated Images via Resynthesis

Pietro Bongini[⋆c], Valentina Molinari[⋆†], Andrea Costanzo[⋆]
Benedetta Tondi[⋆], Mauro Barni[⋆]
[⋆] *University of Siena, Department of Information Engineering and Mathematics Siena, Italy*
[†] *IMT School, Lucca, Italy*
[c] *Corresponding author: pietro.bongini@unisi.it*

Abstract

Synthetic image source attribution is a challenging task, especially in data scarcity conditions requiring few-shot or zero-shot classification capabilities. We present a new training-free one-shot attribution method based on image resynthesis. A prompt describing the image under analysis is generated, then it is used to resynthesize the image with all the candidate sources. The image is attributed to the model which produced the resynthesis closest to the original image in a proper feature space. We also introduce a new dataset for synthetic image attribution consisting of face images from commercial and open-source text-to-image generators. The dataset provides a challenging attribution framework, useful for developing new attribution models and testing their capabilities on different generative architectures. The dataset structure allows to test approaches based on resynthesis and to compare them to few-shot methods. Results from state-of-the-art few-shot approaches and other baselines show that the proposed resynthesis method outperforms existing techniques when only a few samples are available for training or fine-tuning. The experiments also demonstrate that the new dataset is a challenging one and represents a valuable benchmark for developing and evaluating future few-shot and zero-shot methods.

1 Introduction

Generative AI has recently known a very fast development. In particular, a rapid evolution of image generative models was driven by several major elements of novelty, including the introduction of diffusion models and the consequent proliferation of text-to-image generators [1]. On the one hand, this process has brought enormous benefits to quality, availability, realism, and personalization. On the other hand, it has raised issues at multiple levels [2]. In particular, concerns about the misuse of generative AI have increased at the same fast pace

[3]. As a consequence, efforts for the detection of AI-generated images (often referred to as fake images) and the identification of their source have increased accordingly [4].

Among these efforts, the goal of source attribution consists in identifying which generator produced a synthetic image [5, 6]. This problem can be tackled together with detection or as a standalone task. Our work focuses on the latter setting: we assume to already know that a given image is synthetic and wish to determine the model that was used to generate it.

Synthetic Image Attribution (SIA) is made even more challenging by the sheer number of sources available online and by the large number of new ones published every month [7]. It is, then, unfeasible to train a new SIA model every time a new generator appears. In addition, acquiring large amounts of data from commercial generators to fully train a SIA model on them is very expensive. For these reasons, few-shot attribution, where only a small number of examples per source is used for training, is attracting increasing interest [8]. This task can be addressed using a classifier trained on a handful of examples per class [9], or with a modular approach that attributes images by clustering, distance, or similarity [10].

In this paper, we introduce a training-free method for SIA based on resynthesis, which works in a one-shot setting. The synthetic image is assumed to be from a generator within a set of known sources. A textual description of the image is obtained with a captioning model and used to produce a *resynthesis* of the to-be-attributed image with each of the candidate sources. The *resynthesis* which is most similar to the *original* is assumed to be the one generated by the same source. The input image is then attributed accordingly [11]. To judge image similarity, given the impossibility of using pixel-wise distances [12], we resort to a combination of high-level semantic features and low-level signatures [12], extracted with a pretrained CLIP (Contrastive Language Image Pretraining) model [13, 14]. The distance is measured in the feature space using a simple distance function.

Prior work has introduced various benchmarks for SIA [15, 16, 17]. However, most datasets are limited to open-source generators or to a small number of sources, limiting their potential to represent real-world settings where many generative methods, including commercial ones, are available to the average user. Moreover, existing datasets often do not include *resyntheses*, making it difficult to train and test distance-based or similarity-based methods [18]. Hence, as a second contribution, we propose a new dataset incorporating the resynthesis mechanism. The dataset focuses on head-and-shoulder photo-portrait style images produced by text-to-image generators. It incorporates two key features: i) images generated by a set of 14 sources, including 7 commercial generators; ii) secondary descriptions used to generate *resyntheses* of each image, allowing the use of resynthesis-based methods. We used the new dataset to test the performance of our source attribution method and to compare it with several baselines, establishing benchmarks for future research. In summary, the main contributions of our work are:

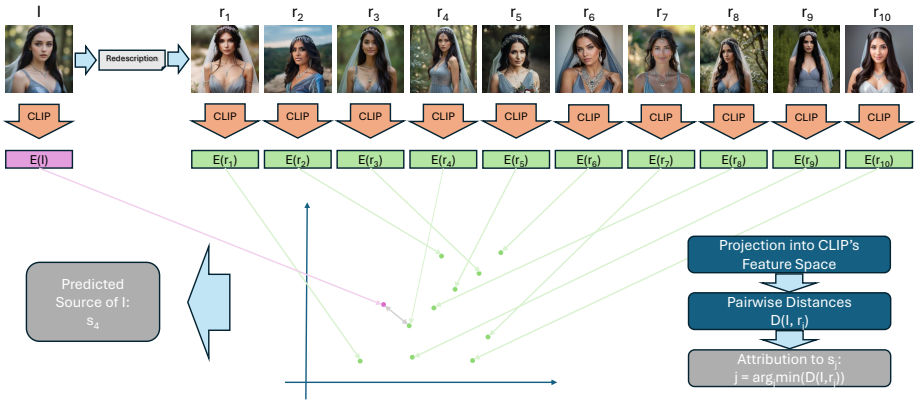


Figure 1: Workflow of the proposed source attribution method based on resynthesis. The *original* image I in this example was generated by Freepik. From left to right, the *resyntheses* r_j were produced by: Bing, Firefly, Flux-dev, Freepik, Imagen3, Leonardo AI, Midjourney, Nightcafe, Stable Diffusion 3, Starry AI. The features of I and all the r_j are extracted with CLIP. The distances between the projections $E(I)$ and each $E(r_j)$ in CLIP's feature space are measured, and I is attributed to the source s_{j*} with the minimum distance $d(E(I), E(r_{j*}))$. In this example, the image is correctly attributed to s_4, which corresponds to Freepik.

- a training-free, one-shot source image attribution method based on image resynthesis;
- a dataset for few-shot SIA, incorporating a resynthesis mechanism based on text image description and generation of *resyntheses*;
- a set of experiments including comparisons with state-of-the-art few-shot source attribution methods, to assess the performance of our method and validate its superiority in a data scarcity regime, also setting baselines for future research on the new dataset.

2 Method

Resynthesis is used for image classification in various settings [19]. In a real-world case, the prompt used to generate a synthetic image is usually unavailable. Given an image to attribute, our approach relies on a secondary description of the *original* image to build *resyntheses* (resynthesized versions of the image) with all the candidate sources. The description is obtained with Chat-GPT [20] and has a maximum length of 200 characters. In such panel of *resyntheses*, the *resynthesis* which mostly resembles the *original* image should be the one

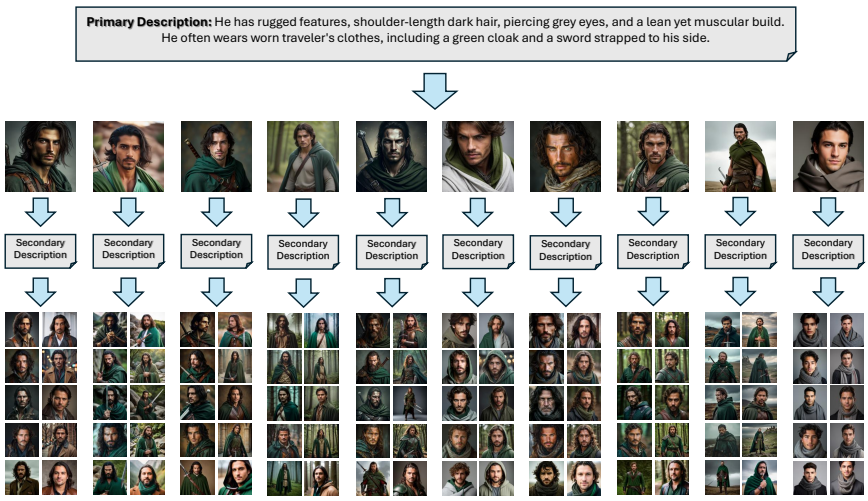


Figure 2: Dataset building procedure. This example corresponds to *character* #42: *Aragorn (The Lord of The Rings)*. Starting from the top and going down: The primary description is used as a prompt for the 10 core generative models introduced in Section 3.1, which produce the images in the first row. The models are in alphabetical order, from left to right. Each image is then passed to Chat-GPT to obtain a secondary description. The latter is used to generate the *resyntheses* in the bottom double-column groups, using all the 10 core generators again. In each group of *resyntheses*, from left to right and from top to bottom, the generators are again in alphabetical order.

generated by the same model. As a consequence, attribution can be obtained by distance calculation: the *original* image is attributed to the generator that produced the closest *resynthesis*. Formally, given a set of n candidate text-to-image sources $S = \{s_1, \dots s_n\}$, and a distance function d, an image I is attributed to the source s_j by extracting a prompt $p(I)$ describing the content of I, then obtaining a set of *resyntheses* $\{r_j = s_j(p(I)) \ \forall s_j \in S\}$, and finally calculating the distances $\{d_j = d(r_j, I) \ \forall s_j \in S\}$. The image I is attributed to the source s_{j^*} achieving the minimum distance. A pre-trained CLIP Large model (input size 336×336) [21] is used as a feature extractor and the distance is measured in the corresponding feature space. Defining the feature extraction as an operator $E()$ and indicating by $d()$ the distance function, we have:

$$s_{j^*} : j^* = \arg\min_j (d(E(s_j(p(I))), E(I))). \tag{1}$$

We tested several distances d, including Euclidean, Manhattan, Mahalanobis, and Cosine distances. The CLIP model is frozen, we do not fine-tune it. Hence, our model is training-free, and can be considered a one-shot method. The only operations required to attribute an image, in fact, are the extraction of the sec-

ondary description with a generic image-to-text language model (chat-GPT in our case [20]), and the generation of one *resynthesis* for each candidate source. A summary of our methodology is shown in Figure 1.

3 Dataset

To assess the performance of the proposed SIA method and compare it with few-shot baselines, we created a new dataset[1]. The dataset has innovative features including: i) images generated by 14 generators (10 core sources, and 4 for the dataset extension), ii) 7 commercial generators, iii) integration of the resynthesis mechanism with the inclusion of *original* images and their corresponding *resyntheses*. The dataset contains 12,000 head-and-shoulder images of novel characters, produced with 14 text-to-image generators described in Section 3.1.

3.1 Generators

The sources used to generate the dataset comprise 7 open-source models (4 of which were used for the dataset extension) and 7 commercial generators. They are listed in the following, with the * symbol indicating commercial models, and the ^{e} symbol the models used in the dataset extension:

- Bing (Dall-E 3) * [22]
- Firefly * [23]
- Flux.1.dev [24]
- Freepik [25]
- Imagen3 * [26]
- Leonardo AI * [27]
- Midjourney * [28]
- Nightcafe * [29]
- Stable Diffusion 3 [30]
- Starry AI * [31]
- AuraFlow ^{e} [32]
- Pixart ^{e} [33]
- Playground v2.5 ^{e} [34]
- Tencent Hunyuan ^{e} [35]

[1]The dataset will be made publicly available upon publication of the paper.

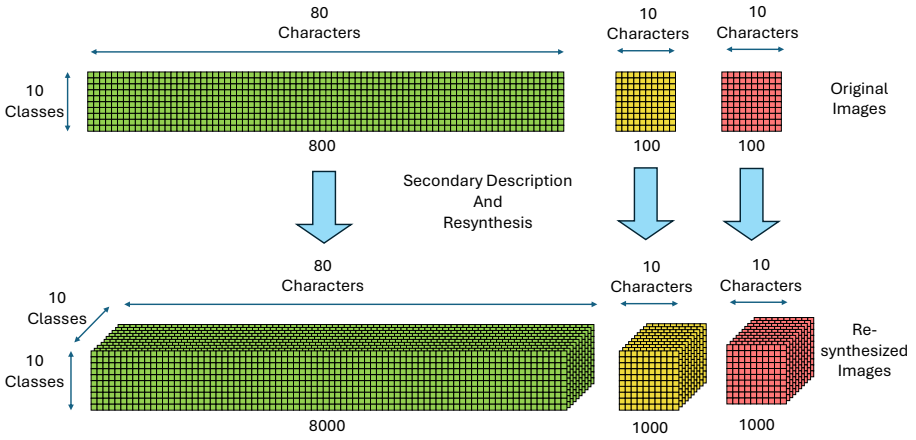


Figure 3: Dataset composition and split. Each of the 100 *characters* is generated with all the sources, obtaining $1,000$ *original* images. Their secondary descriptions are extracted with ChatGPT and used to generate 10 *resyntheses* each, one with each source. These are the components of the main dataset. The dataset is split along the *character* index, meaning that all the 10 *original* images of each character and their 100 *resyntheses* belong to the same set. The dataset extension (Ex) is then obtained by taking the test set and adding the other 4 sources. 40 new *original* images are produced in this way, along with their $40 \times 14 = 560$ *resyntheses* and 400 *resyntheses* of the existing *original* images with the 4 new sources.

3.2 Dataset Construction

We used 100 descriptions of novel characters summarized with ChatGPT [20] to get prompts of uniform length (max 200 character). We built an index on the *character* axis and used it to split the dataset. This prevents images originating from the same prompt ending up in different sets and guarantees that the classes are always balanced. For each prompt, we generated 10 *original* images, one with each of the core sources described in Section 3.1, obtaining $1,000$ *original* images. These are linked into groups of 10 images produced from the same description by the *character* index. To generate the *resyntheses*, a *secondary* description for each *original* image was produced by Chat-GPT [20] using the command: "Provide a detailed description of the character represented in this image in less than 200 characters". *Original* images of the same character from different sources have different *secondary* descriptions, based on the semantical elements introduced by the models and on low-level signatures that can influence language model style. For each secondary description, we generated 10 *resyntheses* (one with each core model). The 10 *resyntheses* of each *original*

image come from the same secondary description and are linked. They are also linked to the other 9 *original* images from the same *primary* description and to all their *resyntheses*, constituting groups of 100 *resyntheses* originating from the same prompt. We split the dataset in training, validation and test sets of 80, 10, and 10 characters. Consequently, the training set is composed of 800 *original* images (OrTr) and their $8,000$ *resyntheses* (ResTr), while the validation and test sets are composed each of 100 *original* images (OrVa, OrTe) and their $1,000$ *resyntheses* (ResVa, ResTe). The procedure is shown in Figure 2. For few-shot experiments on more than 10 classes, we built an extension with 4 additional sources, using only the 10 test set characters. We have 40 additional *original* images, their 560 *resyntheses* generated with all the 14 models, and 400 *resyntheses* of the 100 core *original* images of the test set obtained with the new sources. These are added to the test set, obtaining the extended dataset, which totals 140 *originals* (OrEx) and their 1960 *resyntheses* (ResEx). A scheme is shown in Figure 3.

4 Experiments

4.1 Baselines

4.1.1 CLIP Feature Classifiers

In addition to resynthesis, we use CLIP-Large (input size 336×336) to get image features which are fed into a small classifier: either a Multi-Layer Perceptron (CLIP+MLP) or a Support Vector Machine (CLIP+SVM). The CLIP parameters are frozen and not fine-tuned. The hyperparameters of both heads were optimized on the validation set. For CLIP+MLP, the architecture has two hidden layers with 512 and 32 units, ReLU activation and a softmax output layer. In few-shot scenarios, the architecture is reduced to a single hidden layer of 512 units. The MLP was trained using Adam [36], with initial learning rate 10^{-3}, and l_{2_α} set to 10^{-4}, for a maximum of $1,000$ epochs. Early stopping was applied on the validation loss with a patience of 10 epochs. CLIP+SVM employed a Radial Basis Function (RBF) kernel with regularization $C = 1.0$, tolerance 10^{-3}, kernel degree 3, and no iteration limit.

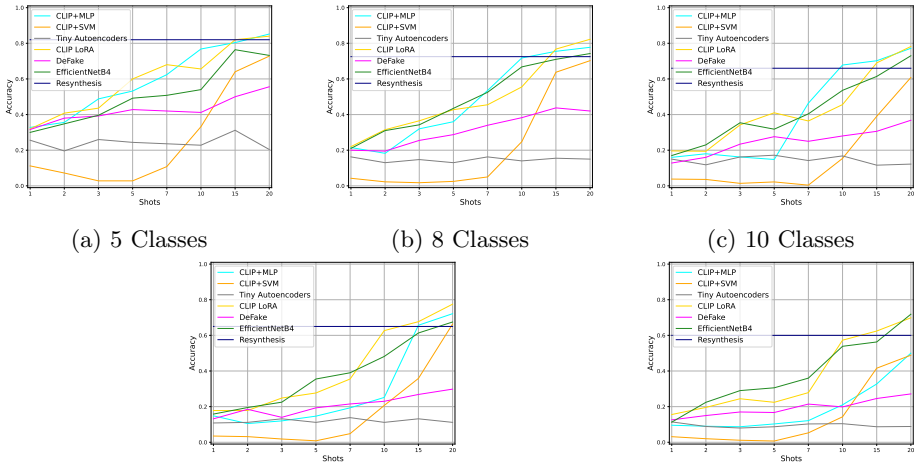
4.1.2 De-Fake

De-Fake [9] leverages subtle artifacts left by generators for SIA. The architecture comprises a detector and an attributor. We use only the attributor module, as all images in our dataset are synthetic. All experiments were conducted using the default parameters reported in [9].

4.1.3 CLIP-LoRA

CLIP-LoRA [37] applies a low-rank adaptation to CLIP to fine-tune it on few-shot tasks. We apply the LoRA adapter to CLIP Large. The classification

head is a MLP with two hidden layers of 768 and 1024 units respectively, both with ReLU activation. The model is fine-tuned for 12 epochs (10 in few-shot experiments) using Adam [36], initial learning rate 10^{-3}, and batch size 16.



(a) 5 Classes (b) 8 Classes (c) 10 Classes

(d) 12 Classes (e) 14 Classes

Figure 4: Accuracy in a few-shot scenario (T1) with different numbers of classes. For the experiments with 12 and 14 classes, the test-only sources were used. The Accuracy is plotted with respect to the number of shots available for training. Our Resynthesis method is the navy-blue line. The performance of our training-free method does not depend on the number of shots, and is therefore constant. The intersection between the line representing our method's accuracy and a baseline's accuracy line corresponds to the number of shot needed by that baseline to outperform our training-free (one-shot equivalent) method.

4.1.4 EfficientNetB4

Convolutional Neural Networks (CNNs) are usually the best option for image classification. The CNN which usually shows the best performance in SIA [39] is EfficientNetB4 [38]. This model is not easy to fine-tune without a large number of samples per class. Hence, our dataset is challenging for it. We fine-tuned EfficientNetB4 for 12 epochs (10 in few-shot experiments), using Adam [36], initial learning rate 10^{-4}, and batch size 16.

4.1.5 Tiny Autoencoders

Tiny Autoencoders [10] is a modular SIA method. For each suspect source, a small CNN autoencoder is trained on a handful of images generated by that source. We used the standard parameters introduced in [10].

Table 1: Accuracy on task T1 - plain source attribution, and T2 - robust source attribution. Each column corresponds to a different post-processing operator applied to test set images. The best result of each column is highlighted in bold.

Task	T1	T2									
Method	Plain	Blur	Brightness	Contrast	Crop	Greyscale	JPEG	Resize	Rotation	Social	WEBP
CLIP+MLP	0.95	0.78	0.88	0.65	0.81	0.72	0.68	0.95	0.82	**0.81**	0.68
CLIP+SVM	**0.97**	**0.86**	**0.94**	**0.74**	**0.82**	**0.74**	0.70	**0.97**	**0.83**	0.80	0.73
Tiny Autoencoders [10]	0.16	0.12	0.15	0.12	0.03	0.14	0.16	0.16	0.15	0.16	0.16
CLIP-LoRA [37]	0.78	0.77	0.62	0.63	0.50	0.62	0.70	0.79	0.77	0.75	**0.74**
De-Fake [9]	0.72	0.65	0.67	0.68	0.72	0.64	**0.72**	0.72	0.70	0.71	0.72
EfficientNetB4 [38]	0.61	0.63	0.57	0.60	0.37	0.64	0.59	0.62	0.61	0.60	0.60
Resynthesis (ours)	0.66	0.62	0.62	0.52	0.50	0.54	0.53	0.67	0.62	0.58	0.47

4.2 Tasks and settings

The *resyntheses* are used for attributing the *originals* based on resynthesis. The methods are evaluated using the Accuracy metric. Tests were carried out on the following tasks.

T1 - Few-Shot SIA only a handful of examples are available to train the methods. The experiments are carried out with different numbers of shots (1,2,3,5,7,10,15,20) and different numbers of classes (5,8,10,12,14). All the tests were carried out on the dataset extension. We averaged the results on 5 repetitions of the experiment, each with a different seed for weight initialization and with different training examples (the same for all the models).

T2 - Plain SIA consists in evaluating SIA performance on OrTe dataset, exploiting OrTr, ResTr, OrVa, and ResVa to train the trainable models.

T3 - Robust SIA similar to T2, the only difference being that the test images are post-processed with a single operator at a time. None of the methods use data augmentation with post-processing operations during fine-tuning. Ten different tests are carried out, using a different post-processing operator each time: Blur; Brightness (1.2 to 2.4 increase), Contrast (1.2 to 2.4 increase); Crop (central, 0.5 to 0.9 the original area); Greyscale; JPEG ($QF \in [50, 99]$); Resize (0.4 to 2.0 the original size); Rotation (-5° to +5°, aspect preserving); Social (simulating an upload on Instagram); WEBP ($QF \in [50, 99]$). Every operator except Blur and Greyscale has a parameter adjusting the transformation strength: its value is randomized for every image, but the same value is used for all the methods.

For task T1 (few-shot), the training samples are limited to the number of shots. Let k be the number of shots, the training ssamples are randomly drawn among the resyntheses of k characters (the same number for each source), while the other 100-k characters are used for testing. Our resynthesis approach does not use the training samples, attributing the *original* images by measuring the distance from their *resyntheses*, in a one-shot fashion. In the following, we report only the results obtained with the Euclidean distance, as other distances yielded

worse performance. While task T1 is the main focus of our experimentation, T2 and T3 are presented for a reference performance in a regular training scenario, where few-shot is not needed. In the latter, the baselines can be trained on OrTr, ResTr, or both. The methods requiring training or fine-tuning were trained once and then tested on both T2 and T3. For each method, we trained three different versions: one using OrTr, one using ResTr, and one using both. The best version of each model was selected based on validation accuracy. For De-Fake, the best version was the one trained on OrTr, while for all the other methods the best version was the one trained on ResTr. The results refer to these versions.

4.3 Results and Discussion

The results of task T1 are reported in Figure 4. Our method outperforms the baselines when few examples are available, on all the numbers of classes. Resynthesis is the best method when 10 or less shots per class are available. Since our method is computationally equivalent to a one-shot method, it is always the best choice for 10 or less shots. This is shown by the navy-blue line representing the performance of resynthesis: line crossing always occurs at 10 or 15 shots. When more shots are available, the best methodologies are CLIP LoRA and CLIP+MLP (the latter shows bad performance on 14 classes). Table 1 displays Accuracy on tasks T2 and T3. The best method is CLIP+SVM, with CLIP+MLP following close. Their advantage comes from the fact that CLIP weights are not fine-tuned, with all the training being dedicated to the SVM/MLP. This, combined with the high informative potential of CLIP features on both low-level and high-level features, allows these two simple models to obtain the best results. The SVM has a well-known advantage in terms of robustness, which makes it outperform the other methods also on the various operators of task T3. The only exceptions are JPEG (De-Fake), WEBP (CLIP-LoRA), and Social (CLIP+MLP). CLIP-LoRA and De-Fake, in particular, show high-robustness. The fact that also these two methods are based on CLIP reinforces the idea that CLIP is capable of embedding high-quality information at all levels. Unsurprisingly, in these tasks, resynthesis is not capable of competing with models which are trained or fine-tuned, even if its performance is not bad.

5 Conclusions

We presented a new training-free method for SIA based on resynthesis, and a dataset for few-shot SIA that contains secondary descriptions and image *resyntheses*. Results show that our method has very good performance with respect to state-of-the-art baselines, outperforming them consistently when 10 or less shots are available for training. Since our algorithm needs to generate just one *resynthesis* per class, it is equivalent to a one-shot method, and hence constitutes the best attribution method when the number of shots is between 1 and 10. This was demonstrated with different numbers of sources, ranging between 5 and 14. We also introduced a new dataset which proved to be a challenging benchmark for

few-shot SIA. Future work will investigate more sophisticated distance functions for the resynthesis method and analyze the impact of the secondary description on overall performance. We will also assess whether ChatGPT is the most suitable choice for generating secondary descriptions when the original prompt is produced by a different method, and explore alternative models for this task. In addition, new image categories will be incorporated into the dataset, and resynthesis performance will be evaluated on these expanded categories.

Acknowledgment

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion models in generative ai: A survey," *arXiv preprint arXiv:2303.07909*, 2023.

[2] H. H. Jiang, L. Brown, J. Cheng, M. Khan, A. Gupta, D. Workman, A. Hanna, J. Flowers, and T. Gebru, "Ai art and its impact on artists," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 363–374, 2023.

[3] N. Marchal, R. Xu, R. Elasmar, I. Gabriel, B. Goldberg, and W. Isaac, "Generative ai misuse: A taxonomy of tactics and insights from real-world data," *arXiv preprint arXiv:2406.13843*, 2024.

[4] D. Tariang, R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Synthetic image verification in the era of generative artificial intelligence: What works and what isn't there yet," *IEEE Security & Privacy*, 2024.

[5] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to gans: Learning and analyzing gan fingerprints," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7556–7566, 2019.

[6] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Do gans leave artificial fingerprints?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 506–515, 2019.

[7] N. Zhang and H. Tang, "Text-to-image synthesis: A decade survey," 2024.

[8] Y. Sheng, W. Si, and J.-Y. Zhu, "Few-shot image attribution for ai-generated art," in *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

[9] Z. Sha, Z. Li, N. Yu, and Y. Zhang, "De-fake: Detection and attribution of fake images generated by text-to-image generation models," in *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pp. 3418–3432, 2023.

[10] L. Bindini, G. Bertazzini, D. Baracchi, D. Shullani, P. Frasconi, and A. Piva, "Tiny autoencoders are effective few-shot generative model detectors," in *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, 2024.

[11] M. Albright, S. McCloskey, and A. Honeywell, "Source generator attribution via inversion.," in *CVPR workshops*, vol. 8, p. 3, 2019.

[12] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do neural networks dream of deepfakes? analyzing synthetic image detectors," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2019.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[14] D. Cioni, C. Tzelepis, L. Seidenari, and I. Patras, "Are clip features all you need for universal synthetic image origin attribution?," 2024.

[15] R. Goebel, N. ElSayed, B. Fröhlich, and A. Kappeler, "Synthetic image attribution: A benchmark," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[16] J. Frank, T. Eisenhofer, A. Rahmati, A. Fischer, D. Kolossa, and T. Holz, "Leveraging the attribution signal for training robust gan detectors," in *International Conference on Learning Representations (ICLR)*, 2020.

[17] D. Gragnaniello, F. Marra, and L. Verdoliva, "Detection of gan-generated fake images over social networks," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1602–1607, 2020.

[18] N. Carlini, H. Farid, I. Goodfellow, N. Papernot, C. Raffel, and R. Shokri, "On the difficulty of detecting gan-generated images," in *IEEE Symposium on Security and Privacy (S&P)*, pp. 2100–2114, 2021.

[19] M. Dong, F. Li, Z. Li, and X. Liu, "Prsn: Prototype resynthesis network with cross-image semantic alignment for few-shot image classification," *Pattern Recognition*, vol. 159, p. 111122, 2025.

[20] OpenAI, "Chatgpt: Language model," 2025.

[21] OpenAI, "Vit-l/14 clip model with 336px input resolution." https://huggingface.co/openai/clip-vit-large-patch14-336, 2021.

[22] OpenAI, "Dall-e 3." https://cdn.openai.com/papers/dall-e-3.pdf, 2024.

[23] Adobe, *Adobe Firefly*, 2023. {https://firefly.adobe.com/}.

[24] B. F. Labs, "Flux." https://github.com/black-forest-labs/flux, 2024.

[25] Freepik, "Freepik AI Image Generator," 2024.

[26] Imagen-Team-Google, "Imagen 3," 2024.

[27] Leonardo AI, "Leonardo AI: AI-Powered Creative Image Generation Platform," 2024.

[28] MidJourney, "MidJourney: An AI-powered image generation tool," 2024.

[29] NightCafe, "Nightcafe studio: Ai art generator," 2025.

[30] S. AI, "Stable diffusion 3 medium." https://huggingface.co/stabilityai/stable-diffusion-3-medium, 2024.

[31] S. AI, "Starry ai," 2023.

[32] fal AI, "Auraflow." https://huggingface.co/fal/AuraFlow, 2024.

[33] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, and Z. Li, "Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis," 2023.

[34] D. Li, A. Kamko, E. Akhgari, A. Sabet, L. Xu, and S. Doshi, "Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation," 2024.

[35] Z. Li, J. Zhang, Q. Lin, and et al., "Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding," 2024.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] M. Zanella and I. Ben Ayed, "Low-rank few-shot adaptation of vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1593–1603, 2024.

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

[39] S. Mandelli, P. Bestagini, and S. Tubaro, "When synthetic traces hide real content: Analysis of stable diffusion image laundering," in *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, 2024.