ImagiNet: A Multi-Content Benchmark for Synthetic Image Detection

Anonymous submission

Abstract

Recent generative models produce images with a level of authenticity that makes them nearly indistinguishable from real photos and artwork. Potential harmful use cases of these models, necessitate the creation of robust synthetic image detectors. However, current datasets in the field contain generated images with questionable quality or have examples from one predominant content type which leads to poor generalizability of the underlying detectors. We find that the curation of a balanced amount of high-resolution generated images across various content types is crucial for the generalizability of detectors, and introduce *ImagiNet*, a dataset of 200K examples, spanning four categories: photos, paintings, faces, and miscellaneous. Synthetic images in *ImagiNet* are produced with both open-source and proprietary generators, whereas real counterparts for each content type are collected from public datasets. The structure of ImagiNet allows for a two-track evaluation system: i) classification as real or synthetic and ii) identification of the generative model. To establish a strong baseline, we train a ResNet-50 model using a self-supervised contrastive objective (SelfCon) for each track which achieves evaluation AUC of up to 0.99 and balanced accuracy ranging from 86% to 95%, even under conditions that involve compression and resizing. The provided model is generalizable enough to achieve zero-shot state-of-the-art performance on previous synthetic detection benchmarks. We provide ablations to demonstrate the importance of content types and publish code and data.

Introduction

State-of-the-art generative models are rapidly improving their ability to produce nearly identical images to authentic photos and artwork. Diffusion models (DMs) (Ho, Jain, and Abbeel 2020; Rombach et al. 2022a), variational autoencoders (VAEs) (Harvey, Naderiparizi, and Wood 2022), and generative adversarial networks (GANs) (Goodfellow et al. 2014) are being utilized in various ways to achieve data augmentation, text-to-image and image-to-image generation, inpainting and outpainting. They facilitate the production of visuals and spatial effects for downstream use in the entertainment, gaming, and marketing industries. On the other hand, these models can be misused by malicious actors (Masood et al. 2021). Thus, there is an increasing demand for improved synthetic image recognition models. Prior work (Wu, Zhou, and Zhang 2023; Gragnaniello et al. 2021; Corvi et al. 2022) employs standard classifiers but struggles with overfitting, bias, and poor generalization to novel generators, limiting effectiveness in synthetic content detection. One key area that has yet to be fully explored in synthetic detection is the creation of training datasets with a broader range of content types and generator sources.

Train/Eval	Corvi2022	Wu2023	ArtiFact	Ours
Balanced	✓/✗	✓/✓	-/✗	✓/✓
Multiple generators	✗/✓	✓/✓	-/✓	✓/✓
Proprietary generators	✗/✓	✗/✓	-/✗	✓/✓
Multiple content types	✗/✗	✓/✓	-/✓	✓/✓
Synthetic resolution	256×256 / 1024×1024	1024×1024 / 8000×8000	- / 200×200	1792×1024 / 1792×1024

Table 1: Feature comparison of previous synthetic datasets. '-' signifies that data is not available.

Previous datasets (Table 1) primarily feature GAN-generated images and lack diversity in resolution, generator types, and content, leading to biases and overfitting issues (Corvi et al. 2022; Gragnaniello et al. 2021; Wu, Zhou, and Zhang 2023; Torralba and Efros 2011). Rahman et al. (2023) provide a diverse benchmark with multiple generators and content types, but the resized low-resolution images make it more suitable for benchmarking rather than training.

We propose a new benchmark and balanced training set for synthetic image detection called *ImagiNet*[1]. It includes images from novel open-source and proprietary generators. Our main goal is to study ways to address the challenge of generalizability by training on diverse data. The images are created by either GAN (Goodfellow et al. 2014), DM (Rombach et al. 2022b), or a proprietary generator – Midjourney (Holz 2023) or DALL·E (Betker et al. 2023). Our benchmark includes two main testing tracks: synthetic image detection and model identification. Testing is performed under perturbations like JPEG compression and resizing, simulat-

[1]https://anonymous.4open.science/r/imaginet-E3DC

Real		Synthetic	
Source	**Number**	**Source**	**Number**
Photos (30%)			
ImageNet	7.5K	StyleGAN-XL	7.5K
LSUN	7.5K	ProGAN*	7.5K
COCO	15K	SD v2.1/SDXL v1.0	15K
Paintings (22.5%)			
WikiArt	11.25K	StyleGAN3	11.25K
		SD v2.1/SDXL v1.0	5.625K
Danbooru	11.25K	Animagine XL	5.625K
Faces (22.5%)			
FFHQ	22.5K	StyleGAN-XL	11.25K
		SD v2.1/SDXL v1.0	11.25K
Uncategorized (25%)			
Photozilla	25K	Midjourney*	12.5K
		DALL·E 3*	12.5K
Total	**100K**	**Total**	**100K**

Table 2: *ImagiNet* dataset structure with two main categories and four subcategories. * signifies images sourced from public datasets.

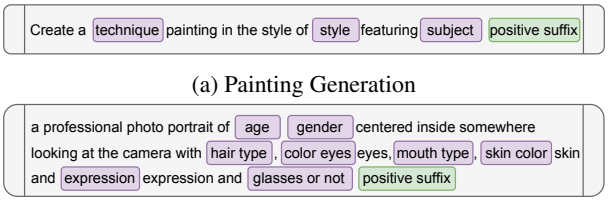
ing social network conditions as in previous works (Corvi et al. 2022). All images are high-resolution, similar to those on social networks, for more consistent results.

Dataset Construction

The *ImagiNet* dataset consists of images from various open-source and proprietary image generators to encompass the distinct "fingerprints" they impart.

Dataset Structure (Table 2) – The dataset structure is designed to represent real-world scenarios where images of different content types might be used. *ImagiNet* examples are split into two main categories – real and synthetic images. To mitigate content-related biases, the dataset is divided into four subcategories – photos, paintings, faces, and miscellaneous. Such images are commonly found on the World Wide Web and are the main subject of generative applications. We provide a balanced amount of synthetically generated images and real counterparts in each subcategory. The source datasets and generator models are given in Table 2. Images from models marked with * are sourced as follows: ProGAN from Wang et al. (2020), Midjourney from Pan et al. (2023), DALL·E 3 from LAION (LAION 2023); in addition we generated 800 DALL·E 3 images to reach our desired dataset size. Synthetic groups are generated with pre-trained models: GAN images are labeled as GAN, Stable Diffusion as SD, and proprietary models as standalone.

Real Images Sampling – The real images are randomly sampled from each real counterpart dataset described in Table 2 (Russakovsky et al. 2015; Yu et al. 2016; Lin et al. 2015; Tan et al. 2019; Anonymous, community, and Branwen 2022; Karras, Laine, and Aila 2019; Singhal et al. 2021). The images in our test set are sampled from the validation and testing splits of these sets.



(a) Painting Generation

(b) Face Generation

Figure 1: Prompt structures for image generation.

Image Generation Procedure – To generate images with GANs (StyleGAN-XL (Sauer, Schwarz, and Geiger 2022), StyleGAN3 (Karras et al. 2021)), we sample random latent code (it is selected according to model requirements) for a given seed and feed the generator with it unconditionally. For DMs and private generators (SD v2.1 (Rombach et al. 2022b), SDXL v1.0 (Podell et al. 2023), Animagine XL (Taqwa 2024), DALL·E 3 (Betker et al. 2023)), however, textual guidance is needed, thus we first search manually for appropriate negative prompts and positive suffixes to increase the quality of the produced images. The construction of each prompt is in descriptive form. For photos, we utilize the captions from COCO (Chen et al. 2015) to prompt the generators and achieve images with sufficient quality. For paintings, instead of using a pre-generated set of captions for prompting, we create lists of styles, techniques, and subjects with GPT-3.5 Turbo (Brown et al. 2020). After that, we fit these characteristics of the paintings in a descriptive sentence shown in Figure 1a, which guides the model to generate varied images. The gaps are filled respectively with an item from the given list, and in the end, a positive suffix is added. The procedure for face generation is similar – Figure 1b presents the structure of the prompt. All the lists for filling in the guiding instructions, as well as the positive suffixes and negative prompts. The last model AnimagineXL, a fine-tuned SDXL (Podell et al. 2023) variant for art generation, uses only tags from the Danbooru dataset (Anonymous, community, and Branwen 2022) for prompting.

Dataset Splits – From the whole set, we sample 80% of the images from each category and subcategory with an equal number of images from the different generators. The number of images in the training set is 160K, respectively 40K are left for testing. We aim to provide a balanced (an equal number of images for each model) calibration set sampled from the training set. It consists of 80K examples in total.

Labelling and Evaluation Tracks – All the images of the dataset are labelled. They have four labels – source (real or synthetic), content type, generator group (e.g., GAN), and specific generator (e.g., ProGAN). In our benchmark, we have two tracks – synthetic image detection and model identification. Perturbations are applied on the test set to simulate social network conditions (Corvi et al. 2022). First, we do a large square crop (ranging from 256 to the smaller dimension of the image) of the image and, after that, resize it to

256×256. After that, we compress 75% of the images with JPEG or WebP compression.

Dataset Access – We provide the synthetic images we generated for this work, along with those from DALL·E 3, which are collected under a Creative Commons Zero license. Both the real counterparts and the additional part of synthetic content (Midjourney and ProGAN examples) can be downloaded from their sources. The whole dataset can be reconstructed with the scripts in our repository, which also includes the list of sources and our synthetic data.

Baseline Training

To train our baseline, we initialize a ResNet-50 model with pre-trained ImageNet weights and modify its early layers to avoid downsampling, following Gragnaniello et al. (2021).

In the first stage of training, we train a backbone with a contrastive objective $\mathcal{L}_{SC}$, as proposed by Bae et al. (2022):

$$\mathcal{L}_{SC} = \sum_{\substack{i \in A \\ \omega \in \Omega}} \frac{-1}{|P(i)||\Omega|} \sum_{\substack{p \in P(i) \\ \omega' \in \Omega}} \log \frac{\exp(\omega(x_i) \cdot \omega'(x_p)/\tau)}{\sum_{l \in Q(i)} \exp(\omega(x_i) \cdot \omega'(x_l)/\tau)} \quad (1)$$

where $A \equiv \{1, ..., N\}$ is a set of indices for all batch examples, $Q(i) \equiv A \backslash \{i\}$ (similarity between z_i and z_i is redundant), and $P(i) \equiv \{p \in Q(i) : \hat{y}_p = \hat{y}_i\}$ is the set of positive examples for a given example i.

A sub-network is attached to the backbone. Its main responsibility is to produce an alternative view of the images in the latent space instead of additional augmented samples to design the SelfCon loss with a single-viewed (augmented once) batch. The sub-network could be a fully connected layer or another architecture with the same function as the backbone. The sub-net $H_{sub}(.)$ is attached to the backbone and projects the latent representations $F_m(.)$ obtained after the m-th ResNet block. The network has two output mapping functions $\Omega \equiv \{H_{sub}(F_m(.)), H(F(.))\}$ for a given input x_i. In our case, the mapping functions $H(.)$ and $H_{sub}(.)$ output representations in $\mathbb{R}^{128}$. This involves accumulating $\mathcal{L}_{SC}$ applied on two labellings - synthetic detection and model identification labels, with each loss assigned equal weight. When optimizing the model detection objective, real images in the batch are ignored. To address the increased memory demands of removing downsampling in early ResNet-50 layers and the large batch size requirements of SelfCon, we adopt gradient caching (Gao et al. 2021), a technique initially designed for language model contrastive losses. We modify it for use with SelfCon (Bae et al. 2022), SupCon (Khosla et al. 2021), and SimCLR (Chen et al. 2020). This approach calculates the loss on the entire batch but accumulates gradients in smaller chunks, allowing for large batch sizes and efficient training on memory-constrained GPUs.

The second stage involves calibrating the model. We detach the sub-network and projection heads, replacing the output projection head with a multilayer perceptron classifier. This classifier is then trained using cross-entropy loss on a balanced dataset to perform both origin and model detection. We update the batch normalization statistics within the backbone's residual blocks, following Schneider et al. (2020), to enhance robustness against perturbations not encountered during pre-training.

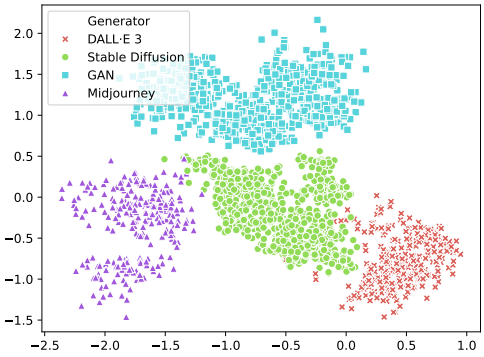


Figure 2: Dimensionality reduction vizualization of the backbone representations for a subset of *ImagiNet*.

Experiments and Results

First, we evaluate the described baseline against existing synthetic datasets. Then, we examine the importance of balancing content types in *ImagiNet* for the performance of detectors.

Baseline – During the first stage, the backbone is optimized with SGD (Ruder 2017) for 400 epochs with batch size $N = 200$ on the *ImagiNet* training set. The initial learning rate of 0.005 is warmed up linearly (Ma and Yarats 2021) for 10 epochs and is cosine annealed (Loshchilov and Hutter 2017) afterwards. The second stage continues for 5 epochs with AdamW optimizer (Loshchilov and Hutter 2017) and constant learning rate 0.0001, weight decay 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. After the pre-training procedure, we visualize the model representations of the images in the test set by applying Autoencoder dimensionality reduction (Meng et al. 2017). The plots in Figure 2 show the ability of our model to cluster each generator's images.

As shown in Table 3, the baseline achieves AUC of up to 0.99 and balanced accuracy over 95% on *ImagiNet*. To demonstrate its generalization abilities we evaluate zero-shot performance on the datasets from (Wu, Zhou, and Zhang 2023) in Table 4, and (Corvi et al. 2022) in Table 5. Our baseline is able to outperform the original method of Wu2023 and remains comparable on Corvi2022's benchmark. The baseline shows a substantial improvement of 12% in ACC on DALL·E 2 examples since it is trained on DALL·E 3 images. The results on StyleGAN3 and StyleGAN2 are increased by 1-2%. Table 6 presents a comparison of the inference time of our detector with previous models. We also train the model proposed in Corvi2022 on *ImagiNet* to demonstrate that the balanced dataset elicits generalizable performance regardless of the architecture and training procedure.

ACC / AUC	Grag2021	Corvi2022	Wu2023	Corvi2022*	Ours*
GAN	0.6889 / 0.8403	0.6822 / 0.8033	0.6508 / 0.6971	0.8534 / 0.9416	**0.9372** / 0.9886
SD	0.5140 / 0.5217	0.6112 / 0.6851	0.6367 / 0.6718	0.8693 / 0.9582	**0.9608** / 0.9922
Midjourney	0.4958 / 0.5022	0.5826 / 0.6092	0.5326 / 0.5289	0.8880 / 0.9658	**0.9652** / 0.9949
DALL·E 3	0.4128 / 0.3905	0.5180 / 0.5270	0.5368 / 0.5482	0.8906 / 0.9759	**0.9724** / 0.9963
Mean	0.5279 / 0.5637	0.5985 / 0.6562	0.5892 / 0.6115	0.8753 / 0.9604	**0.9589** / 0.9930

Table 3: *ImagiNet* test set evaluation – best **ACC**/AUC. * means trained on *ImagiNet*.

ACC / AUC	Wu2023	Ours*
DreamBooth	0.9049 / 0.9733	**0.9601** / 0.9950
MidjoruneyV4	0.8907 / 0.9495	**0.9675** / 0.9959
MidjourneyV5	0.8540 / 0.9224	**0.9745** / 0.9991
NightCafe	**0.8962** / 0.9652	0.8931 / 0.9644
StableAI	0.8806 / 0.9534	**0.9574** / 0.9947
YiJian	0.8392 / 0.9233	**0.9045** / 0.9726
Mean	0.8776 / 0.9479	**0.9428** / 0.9870

Table 4: Practical test set (Wu, Zhou, and Zhang 2023) evaluation – best **ACC**/AUC. * means trained on *ImagiNet*.

ACC / AUC	Corvi2022	Corvi2022*	Ours*
ProGAN	0.9117 / 0.9994	0.9030 / 0.9995	0.8974 / 0.9991
StyleGAN2	0.8662 / 0.9455	0.8675 / 0.9479	**0.8884** / 0.9759
StyleGAN3	0.8557 / 0.9416	0.8705 / 0.9440	**0.8824** / 0.9707
BigGAN	**0.8952** / 0.9699	0.8980 / 0.9882	0.8934 / 0.9864
EG3D	**0.9062** / 0.9756	0.8450 / 0.9160	0.8964 / 0.9913
Taming Tran	**0.9112** / 0.9902	0.8538 / 0.9278	0.8829 / 0.9651
DALL·E Mini	**0.9117** / 0.9914	0.9015 / 0.9792	0.8924 / 0.9786
DALL·E 2	0.6507 / 0.7590	0.7370 / 0.8302	**0.7729** / 0.8590
GLIDE	**0.9062** / 0.9780	0.8730 / 0.9429	0.8539 / 0.9347
Latent Diff	**0.9117** / 0.9998	0.9017 / 0.9989	0.8959 / 0.9902
Stable Diff	**0.9117** / 0.9999	0.9030 / 0.9998	0.8969 / 0.9956
ADM	**0.7927** / 0.8772	0.7875 / 0.8710	0.7704 / 0.8550
Mean	**0.8692** / 0.9523	0.8618 / 0.9446	0.8686 / 0.9585

Table 5: Corvi test set (Corvi et al. 2022) evaluation – best **ACC**/AUC. * means trained on *ImagiNet*.

Content Type Balancing – To investigate the influence of specific content types and identify potential biases, we conducted an ablation study inspired by Leave-One-Out Cross-Validation (LOOCV). Separate models were trained, each with one content type excluded from its training data, while maintaining equal training data overall. The isolation of the specific category influence allows us to identify potential biases through drastic changes in performance when tested on the unseen group of examples.

From the synthetic images in our *ImagiNet* dataset, we focused on those generated by Stable Diffusion due to its presence in all image subcategories, thus eliminating potential generator-specific biases. We sampled a balanced subset containing 4500 real and 4500 synthetic (Stable Diffusion only) images per subcategory (photos, paintings, faces). For each model, we used a ResNet-18 architecture, training it from scratch for 200 epochs to avoid any biases from pre-trained models. Each model was trained on 18000 images

Grag2021	Corvi2022	Wu2023	Ours
24.30	49.53	**16.01**	25.10

Table 6: Inference time in milliseconds for 448×448 image on RTX 4090 GPU.

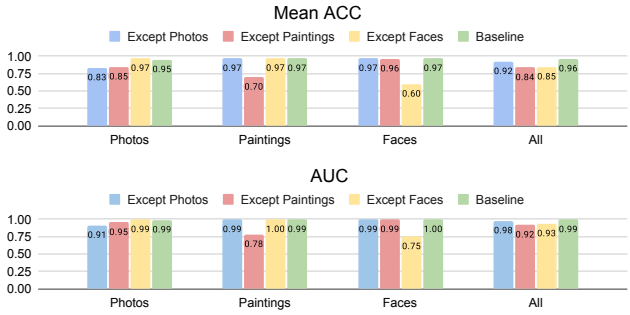


Figure 3: Mean accuracy and AUC on the different models trained by leaving one content type out.

with one category left out. For evaluation, we sample 1000 real and 1000 synthetic images for each category.[2]

Figure 3 demonstrate that models trained by excluding a specific content type exhibit overfitting and generally lower synthetic accuracy when tested on that content type. Notably, the "Except Faces" model overfits the real image distribution, suggesting that bias is introduced not only by synthetic images but also by real images. The AUC plot in Figure 3 reveals high variance from expected values for the "Except Painting" and "Except Faces" models on their respective content types, highlighting the inability to distinguish between the real and synthetic classes at all possible thresholds. This suggests that training on diverse content types is essential for mitigating bias. The baseline model, trained on all types, does not overfit on the test set.

Conclusion

In this work: (1) we demonstrate the importance of balancing content types in synthetic image datasets; (2) we provide a modest-in-size but high quality benchmark for training and evaluating synthetic detectors; (3) we provide a strong baseline which generalizes on third-party datasets.

[2]Our analysis revealed no significant bias toward the resolution of real images across different content type groups.

References

Anonymous; community, D.; and Branwen, G. 2022. Danbooru2021: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset. https://gwern.net/danbooru2021.

Bae, S.; Kim, S.; Ko, J.; Lee, G.; Noh, S.; and Yun, S.-Y. 2022. Self-Contrastive Learning: Single-viewed Supervised Contrastive Framework using Sub-network. arXiv:2106.15499.

Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Jianfeng Wang, L. L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; Manassra, W.; Dhariwal, P.; Chu, C.; Jiao, Y.; and Ramesh, A. 2023. Improving Image Generation with Better Captions.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollar, P.; and Zitnick, C. L. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. arXiv:1504.00325.

Corvi, R.; Cozzolino, D.; Zingarini, G.; Poggi, G.; Nagano, K.; and Verdoliva, L. 2022. On the detection of synthetic images generated by diffusion models. arXiv:2211.00680.

Gao, L.; Zhang, Y.; Han, J.; and Callan, J. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. arXiv:2101.06983.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. arXiv:1406.2661.

Gragnaniello, D.; Cozzolino, D.; Marra, F.; Poggi, G.; and Verdoliva, L. 2021. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. arXiv:2104.02617.

Harvey, W.; Naderiparizi, S.; and Wood, F. 2022. Conditional Image Generation by Conditioning Variational Auto-Encoders. arXiv:2102.12037.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239.

Holz, D. 2023. Midjourney.

Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2021. Alias-Free Generative Adversarial Networks. In *Proc. NeurIPS*.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2021. Supervised Contrastive Learning. arXiv:2004.11362.

LAION. 2023. DALLE-3 images by LAION, Accessed on 05/11/2023.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312.

Loshchilov, I.; and Hutter, F. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983.

Ma, J.; and Yarats, D. 2021. On the adequacy of untuned warmup for adaptive optimization. arXiv:1910.04209.

Masood, M.; Nawaz, M.; Malik, K. M.; Javed, A.; and Irtaza, A. 2021. Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward. arXiv:2103.00484.

Meng, Q.; Catchpoole, D.; Skillicom, D.; and Kennedy, P. J. 2017. Relational autoencoder for feature extraction. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Pan, J.; Sun, K.; Ge, Y.; Li, H.; Duan, H.; Wu, X.; Zhang, R.; Zhou, A.; Qin, Z.; Wang, Y.; Dai, J.; Qiao, Y.; and Li, H. 2023. JourneyDB: A Benchmark for Generative Image Understanding. arXiv:2307.00716.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. arXiv:2307.01952.

Rahman, M. A.; Paul, B.; Sarker, N. H.; Hakim, Z. I. A.; and Fattah, S. A. 2023. ArtiFact: A Large-Scale Dataset with Artificial and Factual Images for Generalizable and Robust Synthetic Image Detection. arXiv:2302.11970.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022a. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022b. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Ruder, S. 2017. An overview of gradient descent optimization algorithms. arXiv:1609.04747.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575.

Sauer, A.; Schwarz, K.; and Geiger, A. 2022. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. arXiv:2202.00273.

Schneider, S.; Rusak, E.; Eck, L.; Bringmann, O.; Brendel, W.; and Bethge, M. 2020. Improving robustness against common corruptions by covariate shift adaptation. arXiv:2006.16971.

Singhal, T.; Liu, J.; Blessing, L. T. M.; and Lim, K. H. 2021. Photozilla: A Large-Scale Photography Dataset and Visual Embedding for 20 Photography Styles. arXiv:2106.11359.

Tan, W. R.; Chan, C. S.; Aguirre, H.; and Tanaka, K. 2019. Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork. *IEEE Transactions on Image Processing*, 28(1): 394–409.

Taqwa, F. 2024. Animagine XL based on SDXL.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR 2011*, 1521–1528.

Wang, S.-Y.; Wang, O.; Zhang, R.; Owens, A.; and Efros, A. A. 2020. CNN-generated images are surprisingly easy to spot...for now. In *CVPR*.

Wu, H.; Zhou, J.; and Zhang, S. 2023. Generalizable Synthetic Image Detection via Language-guided Contrastive Learning. arXiv:2305.13800.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2016. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. arXiv:1506.03365.