VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
SOFTWARE ENGINEERING PROGRAMME

Distinguishing Real from Synthetic: Machine Learning Approaches for Distinguishing AI-Created Images from Photographs

Atskyrimas tarp tikro ir dirbtinio: mašininio mokymo metodai dirbtinio intelekto sukurtų vaizdų atskyrimui nuo nuotraukų

Master's Thesis

Author: Mindaugas Zabotka (signature)

Supervisor: Assist. Prof. Dr. Vytautas Valaitis (signature)

Reviewer: P'Ship Assoc. Prof. Jonas Matuzas (signature)

Vilnius – 2025

Summary

This work addresses the growing challenge of distinguishing between AI-generated images and real photographs, a critical issue in maintaining authenticity and trust in digital environments. With advancements in generative models such as GANs, VAEs, and diffusion models, AI-generated images have become increasingly realistic, raising ethical, social, and security concerns.

The research investigates the feasibility of creating a universal neural network model to classify AI-generated and real images effectively. A diverse dataset of 2,060 images was created. Several neural networks, using the Vision Transformer (ViT) architecture, were trained on this dataset and evaluated for their ability to generalize to images from previously unseen generative models.

Key findings reveal that while the models achieved higher accuracy on known and diffusion-based unseen generators, they struggled significantly with data from generators such as Recraft. This emphasises the inherent difficulty of developing a universal model due to the absence of consistent features across generative models and the rapid evolution of these technologies. The study contributes valuable insights into the limitations of current detection methodologies.

Keywords: neural networks, GAN, VAE, diffusion model, generated images

Santrauka

Šiame darbe sprendžiamas vis didėjantis iššūkis atskirti dirbtinio intelekto sukurtus vaizdus nuo tikrų nuotraukų, ši tema yra labai svarbi siekiant išlaikyti autentiškumą ir pasitikėjimą skaitmeninėje aplinkoje. Tobulėjant generatyviniams modeliams, tokiems kaip GAN, VAE ir difuzijos modeliams, dirbtinio intelekto generuojami vaizdai tampa vis tikroviškesni, o tai kelia etinių, socialinių ir saugumo problemų.

Tyrime nagrinėjama galimybė sukurti universalų neuroninio tinklo modelį, kuris leistų efektyviai klasifikuoti tarp dirbtinio intelekto generuojamų ir realių vaizdų. Buvo sukurtas įvairus 2 060 vaizdų duomenų rinkinys. Šiame duomenų rinkinyje buvo apmokyti keli neuroniniai tinklai, kurie naudoja „Vision Transformer“ (ViT) architektūrą, ir įvertinti jų gebėjimai tinkamai klasifikuoti vaizdus sukurtus anksčiau nematytais generatyviniais modeliais.

Pagrindinės išvados rodo, kad neuroninio tinklo modeliai pasiekė didelį tikslumą klasifikuodamas vaizdus iš žinomų ir difuzijos pagrindu sukurtų nematytų generatorių. Tačiau šiem modeliam nepavyko tiksliai klasifikuoti duomenų sukurtų su „Recraft“ generatoriumi. Tai parodo sunkumą susijusių su universalaus modelio sukūrimu, nes nėra pastovių požymių tarp skirtingų generatyvinių modelių ir šios technologijos sparčiai tobulėja. Šis tyrimas suteikia vertingų įžvalgų apie dabartinių aptikimo metodų trūkumus.

Raktiniai žodžiai: neuroniniai tinklai, GAN, VAE, difuzijos modeliai, sugeneruoti vaizdai

TABLE OF CONTENTS

Introduction

The rapid advancements in generative models and the ease of access to them have significantly increased the proliferation of AI-generated images across social media platforms and other online spaces. From creative arts to scientific applications, generative models such as GANs (Generative Adversarial Networks), VAEs (Variational Autoencoders), and diffusion models have revolutionized content creation by producing high-resolution and realistic images. While these developments have enabled creativity and innovation, they have also introduced numerous significant ethical, social, and security concerns about the misuse of this technology. AI-generated images, for instance, have been weaponized to create deceptive content such as fake identities, manipulated news, and deepfakes, which erode public trust in the authenticity of online content. Addressing the challenge of reliably distinguishing between AI-generated and real images has thus become a critical area of research.

Distinguishing between AI-generated and real photographs has become a critical task for maintaining authenticity and trust in digital environments. Existing detection methods have primarily focused on convolutional neural networks (CNNs) trained to detect artifacts specific to certain generative models. While these approaches have demonstrated reasonable success, they often lack generalizability across diverse generative models. As new and more sophisticated generative techniques emerge, the gap between the capabilities of image generators and detectors widens. This highlights the need for universal detection models that can classify images from previously unseen generators with high accuracy.

The goal of this work is to explore the feasibility of developing a universal neural network for classifying AI-generated images and photographs. The specific objectives of this research are:

1. To identify universal image features that differentiate AI-generated images from real photographs, independent of the specific generative model used.

2. To create a diverse dataset comprising of high resolution generated images, which can be used for robust model training and evaluation.

3. To train a neural network model to achieve robust classification and evaluate the generalization capabilities of the model on images from unseen generators.

4. To analyse the limitations of universal detection models and make propositions for improving performance in future research.

The foundation of this research lies in the hypothesis that AI-generated images possess inherent artifacts or patterns that differ from real images. And that these artifacts can be exploited by advanced neural networks to distinguish between real and generated images. The methodology employed in this work includes:

1. A dataset comprising real photographs and AI-generated images from multiple state-of-the-art generative models will be created.
2. The applicability of techniques suggested in recent studies for universal model training will be analysed.
3. A neural network model utilizing state-of-the-art techniques will be trained.
4. The model's performance will be evaluated using images from unseen generative models to assess its generalization ability.

Research in the domain of AI-generated image detection has primarily focused on using CNN-based architectures. However, recent studies into Vision Transformers (ViTs) and hybrid architectures have shown to be superior for this task, thanks to their ability to capture global relationships within images. Despite these advances, there is a lack of consensus on whether a universal detection model is feasible, particularly given the diversity and rapid evolution of generative techniques. This work builds on this body of knowledge by addressing the feasibility and challenges of universal detection.

1. Literature Review

1.1. Image Generation

The field of image generation has seen significant advancements, leading to the development of highly sophisticated generative models. These models, such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models, have dramatically improved the ability to create realistic digital images, such that it has become hard for people to distinguish them from real images [LHB+23]. This section explores the various image generation models, delving into their unique methodologies, strengths, and applications. Understanding these diverse approaches is crucial not only for appreciating the technological progress in image generation but also for addressing the challenges associated with detecting these generated images. As generative models continue to produce increasingly realistic outputs, distinguishing between these synthetic images and genuine photographs becomes a critical task, especially in the context of combating disinformation, identity theft, and other malicious activities.

As mentioned in [OLL23], there are different categories of synthetic images. One category includes images that have a portion of the original image altered with tools like Adobe photoshop[1] or methods for creating deepfakes, which involve changing the face of a person in an image or video. Another recent feature of Adobe's photoshop and DALL-E[2] allows users to alter a portion of a real image by inserting objects or modifying parts of the scene based on the user's prompt. The other category contains images which have been fully generated by a generative model, and this is the category of images that this literature review will focus on.

Generative Adversarial Networks

GANs consist of two neural networks that are trained simultaneously and compete to outperform one another. One of these networks, the generator, tries to create images that can fool the other network, the discriminator, into classifying them as real, while the discriminator aims to accurately classify between real images and those generated by the generator. Through this adversarial process, both networks improve at their respective tasks—the generator produces more realistic images and the discriminator becomes better at distinguishing between real and generated images. Although GANs have been used for a while and have managed to achieve realistic-looking image generation, they have a significant drawback compared to other models. GANs are notoriously difficult to train, requiring careful tuning of hyperparameters and large amounts of computational

[1] https://www.adobe.com/products/photoshop.html

[2] https://openai.com/index/dall-e

resources. Additionally, there are a variety of issues that can arise while training the models, which can hinder their performance.

Variational Autoencoders

Variational Autoencoders (VAEs) are another class of generative models that have significantly advanced image generation techniques. A VAE consists of two main components: the encoder and the decoder. The encoder maps the input data, such as an image, into a latent space representation of the input data. This latent space is designed to represent the data in a way that retains its most meaningful characteristics. The decoder then reconstructs the original input from this latent space representation, aiming to minimize the reconstruction error. By doing so, VAEs can generate new, similar images by sampling from the latent space, effectively learning the underlying distribution of the input data and producing realistic variations of the original images.

Diffusion Models

GANs dominated the image generation space for a long time, until the emergence of diffusion models, which enabled the creation of better-looking and highly realistic images while also being easier to train. Diffusion models represent a different approach to image generation compared to GANs. Diffusion models, in essence, try to remove noise from a noisy image in order to restore it to the original image while following the prompt that the user provided.

The training process involves taking an input image and progressively applying multiple steps of noise to it. The model's objective is to predict and subtract the added noise at each step until the original image is restored, while being guided by a prompt describing what the original image should look like. Over time, the model learns to reproduce an image that is described in the prompt from a very noisy input image. When used for image generation, the model can be given an image to guide the generation process, or no image can be used for the input, and instead the model uses random noise to produce an image following a prompt.

Currently, the most popular diffusion models belong to the Stable Diffusion[3], DALL-E, and Midjourney[4] families. While DALL-E and Midjourney models are accessible only through APIs, the older Stable Diffusion models are open-source and have been extensively customized and enhanced by the community. This has created a rich ecosystem of tools and extensions, further advancing the capabilities and applications of diffusion-based image generation.

[3]https://stability.ai/stable-image

[4]https://www.midjourney.com

Training Techniques

Diffusion models offer several advantages that make them highly versatile and effective for image generation tasks. One significant advantage is their ability to be easily modified and fine-tuned using techniques like Low-Rank Adaptation DreamBooth [RLJ+22] or (LoRA) [HSW+21].

DreamBooth is a model training technique designed for fine-tuning pre-trained diffusion models by updating the entire model based on just a few images of a specific subject or style. This technique works by associating a unique word or identifier in the prompt with the input images used during training, enabling the model to learn and replicate the distinctive features of the subject or style, such as a specific person or a particular art style of an artist. This approach is a significant advancement over previous methods, which typically required large datasets and substantial computational resources to achieve comparable levels of ”personalization.” DreamBooth makes it possible to customize diffusion models to produce highly specific and personalized outputs without requiring a lot of computational resources.

Model merging is another technique utilized by stable diffusion models. This technique involves combining multiple pre-trained models to create a single, more versatile model that inherits the strengths and unique features of both models. Because of the open-source models like Stable Diffusion, the community has been able to experiment with and refine this approach, leading to models that can generate a wider variety of styles and subjects with enhanced fidelity.

Another big advancement made in image generation is LoRA. LoRA enables efficient adaptation of pre-trained diffusion models to new tasks by updating only a small subset of parameters, reducing the amount of resources required to adapt or retrain the model. This allows users to customize models for specific applications without the need for extensive retraining. LoRA made a big impact in the image generation space because many people did not have the computational resources needed to retrain the whole Stable Diffusion model, making image generation more accessible to a wider population. Furthermore, it is also possible to merge LoRa models with other models, making the whole process of model training so much more customizable and easier overall.

With so many improvements in image generation, the output image quality has become much better and consequently much harder for humans to identify as fake. This enhanced realism, combined with the reduced computational resource requirements, has made it much easier for malicious people to utilize image generation for unlawful activities.

1.2. Detection Methods

Most of the existing methods for generated image detection initially utilized convolutional neural networks (CNNs) for image identification. However, recent advancements have demonstrated that Vision Transformer (ViT) models outperform CNNs in this task. ViT-based models excel in capturing relationships across the entire image and have shown better accuracy and robustness compared to CNN models, making them the current state-of-the-art approach for this task.

As image generators evolve, it becomes harder to identify their generated images as fake. This is to be expected because the quality of the generated images increases. The increase in quality means that errors previously made by image generators, such as blurring, inconsistent shadows, incorrect rendering of human hands, and other high detail areas, become less common. Meaning that generated image detection methods that heavily rely on the errors made by the generators become less accurate. And this will be more the case with image generators in the future. While many detectors have achieved high accuracy with using the errors in the spatial domain of the image for generator detection, these methods do not perform well with newer diffusion generators such as Stable Diffusion or Midjourney. This is because the newer models have almost eliminated the errors that the detectors exploit to detect generated images. Although one recent research [QSX+24] managed to exploit inconsistencies in human eye reflections to differentiate between real and fake images. However, just like how different eye colors have been fixed, so too will this defect disappear from new models. In addition, these detectors do not work with landscape images, which have a lot of undefined features unlike human faces, hands, or other objects. So in the near future, these methods will become obsolete because image generators have already eliminated most of their mistakes and will only continue to improve. That is why a lot of research articles are focused on detecting generated images based on other features that are more likely to persist as models improve.

One of the more popular detection methods involves analysing the frequency domain of images. Multiple studies [YDF19; WBZ+23; ZKC19; ZXL+24] have observed that, unlike traditional photography, image generation follows a specific process that leaves distinct artifacts in the frequency domain. Although this is not entirely correct, as there are studies [AGV18; KFK+18; Pen20] that use noise left by camera lenses and sensors to differentiate between different camera models, but many researchers do not consider the noise that exists in photographs. By using the frequency domain features, researchers have trained various neural network models to recognize the patterns left by generative networks and differentiate between real and generated images with high accuracy. However, a drawback of this method is that each generative network creates its own distinct pattern. So if a detector is trained with images generated by only one model, it will not be able to

detect images generated by other models. Many articles [YHC+22; OLL23; WBZ+23; WWZ+20; ZHL+23; ZWC+22; ZWH+22; ZXL+24] observe this as an issue, and some suggest new methods that are able to detect images generated by unseen models to some extent.

Although many research articles consider multiple model accuracy, they state their results as the highest accuracy achieved on generative models that the detector is trained on. This means that the detectors have to have prior knowledge about which model the image was created by, which is not representative of real-world scenarios. However, some research [OLL23; QSX+24; WWZ+20] has observed that detectors trained on GAN models are able to detect images from some of the other GAN models with quite a high accuracy. This is also true for models trained on diffusion model generated images when detecting images generated by other diffusion models [WBZ+23; XWM+23]. These results suggest that generative models, which have similar image generation techniques and produce similar patterns within the images, and a detector trained on only a handful of generators should be able to detect images generated by unseen generators, as long as their architectures are similar to the ones in the training batch. This is further analysed by several studies [YDF19; YHC+22], which have observed that training a GAN model on a different dataset or even fine-tuning it changes the ”fingerprint” that the GAN leaves on the images. Meaning that images generated by models that have been re-trained on a different dataset would be able to evade detection by detectors that do not account for this. To overcome this, the authors introduce a new method called DNA-Det, which identifies globally consistent fingerprints left by GAN type models. DNA-Det is made of two techniques: pre-training on image transformation classification and patchwise contrastive learning. The architectural fingerprints identified by this method remain recognizable even if the models are fine-tuned or retrained, unlike traces left by model weights that vary regionally. Experiments demonstrated by the authors show that their method greatly outperforms other methods when tested on different model configurations and training datasets.

[OLL23] suggests that the inability to detect images from unseen generators is because the models are trained to only detect images that show patterns of being ”fake.” As described in [OLL23] ”the classifier doesn't seem to look for features of the real distribution when classifying an image as real; instead, the real class becomes a ‘sink class’ which hosts anything that is not GAN's version of fake image.” To solve this problem, the authors of [OLL23] ”propose to perform real-vs-fake image classification using features that are not trained to separate fake from real images.”, meaning to use a network trained for a similar task of differentiating between images based on low-level details without the specific task of fake image detection. In their research, a CLIP:ViT-L/14[5] model is used, modified with an additional linear layer, and only this layer is trai-

[5]https://huggingface.co/openai/clip-vit-large-patch14

ned for binary classification between real and fake. Another study [HCM+24] has also used CLIP[6] in their research and suggests removing textual embeddings from CLIP by ”integrating a “forget-to-spell” model: an orthogonal linear projection designed to minimize textual content in CLIP’s latent space.” This is because CLIP may unintentionally insert textual information into the output, as observed in [MTB22].

One study [ZXL+24] proposes a universal method for generated image detection that works for images generated by all types of generative networks. Based on the observation that ”Generative models leave different artifacts between the poor and rich texture regions” [ZXL+24], this method compares the inter-pixel correlation between rich and poor texture areas within the image. Because cameras work differently from generative networks, in the way that for images created with cameras, the rich and poor texture area inter-pixel correlation is very similar. Their method consists of cropping the image into multiple patches, which are then separated into low and rich texture areas, and two images are reconstructed. One of the reconstructed images consists only of patches of rich texture and another of only low texture areas. The resulting images are fed into a classifier to determine if the image is fake or real, achieving very high accuracy over a lot of different models.

To address the challenge of detecting fake images from unseen models, the authors of [ZHL+23] introduce a generalized fake image detection framework based on gated hierarchical multi-task learning (GHML). The framework integrates a global artifact learning task and a block-wise spatial correlation learning task, employing techniques like region masking augmentation and jigsaw puzzles with color jitter operations to enhance generalization and prevent overfitting. Experiments on the ForenSynths dataset demonstrate that the method performs quite well in distinguishing fake images generated by various GANs and diffusion models when compared to other methods.

The research article [WBZ+23] addresses the challenge of identifying images generated by diffusion models, which have shown to be difficult for traditional detectors to detect, even when trained on data including specific diffusion models, as also observed in [CCZ+23]. The authors introduce a new image representation called Diffusion Reconstruction Error (DIRE). DIRE measures the error between an input image and its reconstruction by a pre-trained diffusion model [SME20], based on the observation that diffusion-generated images can be more accurately reconstructed compared to real images.

[XWM+23] proposes an image detection method for images generated by diffusion networks. And their method performs much better than other proposed methods used for GAN generated image detection when trained on the same dataset. They also noticed that networks trained on diffusion models could not detect images generated by GAN models and vice versa.

[6] https://openai.com/index/clip

”under the assumption that it is difficult to synthesize high-quality, high-frequency components in local regions.”[ZWC+22] have proposed a universal generated image detection method that works by analysing high-frequency component rich areas. They trained their model on a dataset made from images generated by ProGAN and tested it on images generated from multiple GAN models.

To increase accuracy over unseen generators, some recent research has suggested that training models specifically for the task of distinguishing generated images from real ones may not be the most effective approach. Studies, such as those presented in [MGP24; OLL23; QSX+24; ZWH+22], propose that utilizing a feature space not explicitly designed for differentiating between real and generated images can yield better results. The authors of [MGP24; OLL23] argue that models pre-trained within a more generalized feature space can enhance their ability to learn distinguishing features between real and generated images more effectively. Further re-training these models specifically for the task of generated image detection would enable them to outperform models that were explicitly trained for the purpose of fake image detection from the start. The empirical results from their research support this hypothesis, demonstrating superior performance in detecting generated images across different generative models.

When training generated image detection models on large amounts of images, there is a probability that some of the data might be labeled incorrectly. Therefore, the authors of [QSX+24; ZWH+22] proposed an unsupervised training technique where the data is assigned noisy labels. The biggest advantage of this technique is that it allows the image detection network to be trained on a dataset that has a high amount of incorrectly labeled data. This is important because when using big datasets for training a neural network, there is always a chance of data being mislabeled, which would heavily affect the model’s performance. Instead of using labeled data, these studies use a backbone network for feature extraction and separate the data by distance in the feature space. But like many other methods, this method suffers from low accuracy when detecting images from unseen generators.

Current methods for detecting generated images are focused on detecting images of existing generators and are posing challenges when applied to new, unseen models. So, instead of focusing on generated images, the authors of [BLY+23] propose a new approach that relies only on real images for training, thereby eliminating the dependency on generated images. The authors observed that the noise between real images is very similar, and generated image noise is very different. By mapping real images to a dense subspace within the feature space, the model is designed to detect generated images as outliers that fall outside this subspace. This strategy offers several advantages, including significantly reduced training data requirements and improved generalization to new

generative models. Moreover, it maintains robustness against various post-processing techniques applied to images, making it highly applicable in real-world scenarios.

Table 1: The detection accuracy comparison between different approaches. DIRE-D denotes DIRE [WBZ+23] detector trained over fake images from ADM. DIRE-G denotes DIRE detector trained over the same training set (ProGAN) as others. Among all detectors, the best result and the second-best result are denoted in boldface and underlined, respectively. Data is from [ZXL+24].

Generator	CNNSpot	FreDect	Fusing	GramNet	LNP	LGrad	DIRE-G	DIRE-D	UnivFD	PatchCraft
ProGAN	**100.00**	99.36	**100.00**	99.99	99.95	99.83	95.19	52.75	99.81	**100.00**
StyleGan	90.17	78.02	85.20	87.05	<u>92.64</u>	91.08	83.03	51.31	84.93	**92.77**
BigGAN	71.17	81.97	77.40	67.33	88.43	85.62	70.12	49.70	<u>95.08</u>	**95.80**
CycleGAN	<u>87.62</u>	78.77	87.00	86.07	79.07	86.94	74.19	49.58	**98.33**	70.17
StarGAN	94.60	94.62	97.00	95.05	**100.00**	99.27	95.47	46.72	95.75	<u>99.97</u>
GauGAN	<u>81.42</u>	80.57	77.00	69.35	79.17	78.46	67.79	51.23	**99.47**	71.58
Stylegan2	86.91	66.19	83.30	87.28	**93.82**	85.32	75.31	51.72	74.96	<u>89.55</u>
whichfaceisreal	**91.65**	50.75	66.80	86.80	50.00	55.70	58.05	53.30	<u>86.90</u>	85.80
ADM	60.39	63.42	49.00	58.61	<u>83.91</u>	67.15	75.78	**98.25**	66.87	82.17
Glide	58.07	54.13	57.20	54.50	83.50	66.11	71.75	**92.42**	62.46	<u>83.79</u>
Midjourney	51.39	45.87	52.20	50.02	69.55	65.35	58.01	<u>89.45</u>	56.13	**90.12**
SDv1.4	50.57	38.79	51.00	51.70	89.33	63.02	49.74	<u>91.24</u>	63.66	**95.38**
SDv1.5	50.53	39.21	51.40	52.16	88.81	63.67	49.83	<u>91.63</u>	63.49	**95.30**
VQDM	56.46	77.80	55.10	52.86	85.03	72.99	53.68	**91.90**	85.31	<u>88.91</u>
wukong	51.03	40.30	51.70	50.76	86.39	59.55	54.46	<u>90.90</u>	70.93	**91.07**
DALLE2	50.45	34.70	52.80	49.25	92.45	65.45	66.48	<u>92.45</u>	50.75	**96.60**
SDXL	53.03	51.23	55.60	64.53	87.75	71.30	55.35	<u>91.28</u>	50.73	**98.43**
Average	69.73	63.28	67.63	68.43	<u>85.28</u>	75.11	67.90	72.70	76.80	**89.85**

The results of different methods can be observed in Table 1. The methods tested are CNNSpot[WWZ+20], FreDect [FES+20], Fusing [JJK+22], GramNet [LQT20], LNP [BLY+23], LGrad [TZW+23], DIRE [WBZ+23], UnivFD [OLL23], and PatchCraft [ZXL+24]. All of the models are trained on a dataset made up of 360k real images from LSUN [YSZ+16] and 360k fake images generated with ProGAN. When evaluating the performance of these detection methods, it can be seen that PatchCraft performs the best overall, having the best average accuracy and having the highest or second highest accuracy on most generators. While the LNP method did not achieve the highest accuracy on most generators, it has a consistent high accuracy with the second highest average accuracy. DIRE-D method, while not performing well on GAN type models, performed better than PatchCraft on diffusion networks. Although DIRE-D was trained on a different dataset, so those results are to be expected.

1.3. Image Datasets

In the realm of synthetic image detection, the choice of dataset plays a crucial role in shaping the performance and reliability of the detection models. Various studies have utilized a range of publicly available datasets, while other studies have created their own.

CelebA [LLW+15] and FFHQ [T K19] are two of the most commonly used datasets, especially in facial recognition and manipulation detection research. CelebA contains a large number of celebrity images with detailed attribute annotations, and FFHQ offers high-resolution images with a wide range of facial attributes.

ImageNet [DDS+09] serves as a foundational dataset for many deep learning applications, including synthetic image detection. Due to its vast array of labeled images covering numerous categories, it is often used to pre-train models that are subsequently fine-tuned for the specific task of detecting fake images.

Specialized datasets like ForenSynths and FaceForensics++ [RCV+19] cater specifically to the needs of forensic analysis and deepfake detection. ForenSynths includes images generated by a variety of GAN models, while FaceForensics++ offers a diverse collection of altered videos created using various face-swapping methods.

Broader datasets like COCO (Common Objects in Context) [LMB+15] and LSUN (Large-scale Scene Understanding) [YSZ+16] expand the scope of synthetic image detection beyond facial images to include common objects and scene types. COCO has a rich variety of objects in everyday contexts. LSUN focuses on large-scale scene understanding, providing images of different environments such as bedrooms and churches, which help with testing the model's generalization to other types of content.

Some studies develop their own datasets to more accurately assess the performance of their methods. The authors of [WBZ+23] have created a diffusion generated image dataset, DiffusionForensics. This dataset includes images generated by eight different diffusion models, trained on the ImageNet and LSUN datasets.

There are, however, several limitations with these and other popular datasets that need to be accounted for during detector training. Firstly, some datasets, having been created several years ago, contain images that may not match the quality produced by current state-of-the-art generative models. This discrepancy can lead to detection methods that are less effective when applied to newer, more sophisticated AI-generated images. Moreover, the primary goal of developing detection methods for AI-generated images is to identify those that are challenging for humans to differentiate from real images. If a dataset has a lot of images that are easily identifiable as generated,

the high accuracy achieved on these datasets will not accurately reflect the detector's performance in real-world scenarios. Additionally, many datasets are limited in their variety, often containing images of a single type of object or environment. This lack of diversity can result in detectors that are not robust to changes in image content, reducing their effectiveness when applied to a broader range of images.

Another significant issue is the potential for overfitting. When detectors are trained extensively on a specific dataset, as observed in [YDF19; YHC+22], they may learn to recognize the peculiarities and artifacts of that dataset rather than general characteristics of AI-generated images. This overfitting makes the detectors less effective when exposed to images from different sources or newer models that generate images with different characteristics or that were trained on different data. Furthermore, the rapid advancement in generative models means that datasets quickly become outdated. What was once a challenging dataset can become obsolete as generative models improve, requiring continuous updates and the creation of new datasets that better represent the capabilities of current models.

Due to these limitations and other reasons, some studies [HFC+24; WBZ+23], have created their own datasets to achieve better performance in AI-generated image detection. For instance, the "WildFake" dataset [HFC+24] is designed to address the growing need for reliable AI-generated image detection by offering a diverse collection of images sourced from various open-source communities. This dataset includes a wide array of object categories and styles, ensuring a comprehensive representation of real-world images. Moreover, unlike many existing datasets, WildFake includes images generated by a variety of models, including GANs and diffusion models, which is crucial for evaluating the generalization capabilities of detection algorithms across different generative techniques. Table 2. shows the superiority of detectors trained on the WildFake dataset compared to the ArtiFact [RPS+23] dataset. Detectors were evaluated using generated images from DiffusionForensics, GenImage [ZCY+23], DiffusionDB [WMM+22], ArtiFact, and WildFake, using metrics such as accuracy (ACC), average precision (AP), and Area Under the ROC Curve (AUC).

Image Augmentation Techniques

Most studies referred to in this work apply some form of augmentation to their data during training, including common techniques such as translation, scaling, and flipping. These augmentations enhance the variability of the training data, which in turn helps improve the generalization capabilities of the models. Some articles specifically examine the impact of different strengths of Gaussian blur and JPEG compression on model accuracy. For instance, the authors of [WWZ+20]

Table 2: Evaluation of ResNet50 and ViT architecture detectors, trained on the ArtiFact and WildFake datasets. ACC(%), AP(%), and AUC (%) are reported. Data is from [HFC+24].

Training Detectors and Datasets	Testing Dataset					Avg
	DiffusionForensics	GenImage	DiffusionDB	ArtiFact	WildFake	
ResNet50-ArtiFact	85.4/94.9/76.7	76.5/84.8/82.9	64.1/69.9/68.1	**97.2/99.5/99.3**	85.4/94.9/76.7	81.7/88.8/80.74
ResNet50-WildFake	**87.2/96.6/83.4**	**80.9/89.9/89.3**	**96.3/99.2/99.2**	68.0/84.7/75.3	**99.6/99.9/99.9**	**86.4/94.1/89.42**
ViT-ArtiFact	84.2/96.1/82.5	78.5/88.1/85.0	68.4/75.3/73.2	**96.8/99.6/99.5**	84.2/96.1/82.5	82.4/91.0/84.4
ViT-WildFake	**95.8/99.1/97.2**	**88.6/83.6/89.7**	**99.3/99.8/99.9**	62.2/81.9/68.8	**99.1/99.9/99.9**	**89.0/92.84/91.1**

conducted extensive testing with various augmentations on different detectors and concluded that augmentations generally improve model performance. However, they noted exceptions in datasets involving super-resolution and deepfakes, where certain augmentations did not yield the same benefits.

A rational observation is made in [OLL23], that using a strong Gaussian blur with a sigma value of 2 or higher degrades image quality to the extent that it compromises the usefulness of the image, which would not be worth it to merely evade detection. This suggests that training a model to detect images with such degraded quality is not worth it because these kinds of images would not appear in real world scenarios.

1.4. Gaps in Research

Many studies that analyse the features left in the frequency domain by image generators do not account for the fact that different camera models also leave a different fingerprint on the image, as analysed in [AGV18; BLY+23; KFK+18; Pen20]. Camera noise originates from the post-processing algorithms applied during the conversion of raw sensor data into displayable image data. It is not clear if the noise left by cameras is ”weaker” or more easily identifiable than noise left by image generators, and there seems to be no studies done on that specific topic. However, it can be theorized that classifiers can detect camera noise, as images from unseen classifiers are often classified as real, as observed in [OLL23]. This could be because fake data in datasets is typically generated by a limited number of image generators, resulting in only a few distinct noise patterns. On the other hand, real images are captured by a variety of camera models, each introducing different noise patterns. Thus, when an image from an unseen generator with an unknown noise pattern is introduced, it may be misclassified as real due to the diverse noise patterns present in real images. Although it is also possible that an image from an unseen generator is misclassified because the classifier simply cannot detect their noise pattern due to only learning the patterns that it knows are fake.

1.5. Further research methodology

In this section, I will outline the most proficient methodologies and datasets that will be further researched in the master's thesis. Potential improvements based on gaps identified in the current literature will also be proposed here.

Methodology

Based on the current research, it is seen that models trained on very large amounts of data perform better than other models on images from unseen generators. For this reason, CLIP will be used as the classifier network for the task of generated image detection. Furthermore, using forensic methods, such as described in [ZXL+24] will be considered, as they have shown to increase accuracy on unseen image generators.

Dataset

Although studies have shown that models trained on larger datasets perform better, many existing datasets contain outdated images that do not match the quality achievable by current generative models. Using such data can reduce the accuracy of models in real-world scenarios, as they may not generalize well to higher-quality images produced by state-of-the-art generators. Consequently, methods that achieve high accuracy on those datasets might not work well when trained on higher-quality data. Therefore, for the research in the master's thesis, it would be best to use datasets that have recent, high-quality images. If no suitable datasets are available, either a new dataset will be created to meet these requirements, or a large dataset like WildFake [HFC+24] will be used, as it is composed of a lot of popular datasets.

Image augmentation

A lot of studies agree that image augmentation increases the robustness and generalizability of detection models. Therefore, common augmentations like rotations, translations, scaling, and flipping, as well as Gaussian noise or JPEG compression artifacts, will be used to increase data variety. Furthermore, every augmentation will be applied with reasonable intensity, where it does not degrade model performance, but will still allow the model to work on a wider array of data.

2. Research solution

The solution of this study is designed to rigorously investigate the feasibility of training a universal neural network model to distinguish AI-generated images from real photographs. The approach encompasses three primary stages: data collection, model selection and training, and evaluation. Each stage is meticulously detailed below to ensure reproducibility and clarity, addressing the challenges of dataset diversity, model generalization, and robust evaluation.

2.1. Problem Analysis

2.1.1. The Problem of AI-Generated Image Detection

In recent years, the emergence of highly sophisticated generative models such as diffusion-based networks, generative adversarial networks (GANs), and transformer-based image generators has significantly diminished the line between real and synthetic images. These models are capable of producing high-fidelity visuals that are often indistinguishable from real photographs to the human eye. As these systems become more accessible, they also become more attractive tools for misinformation, content manipulation, and deepfake creation. This presents a pressing need to develop reliable methods for detecting AI-generated content.

However, unlike earlier iterations of generative models that introduced artifacts or failed to replicate realistic textures, current generation models such as MidJourney, Ideogram, and Recraft are optimized to mimic the imperfections of natural images, such as image blur and lens distortions. Therefore, previous artifact-based detection techniques have become increasingly obsolete. This creates a moving target for detection methodologies: models trained on one type of generator may fail when presented with outputs from newer models or models of a different architecture.

Another aspect of this challenge is the variability in image sources. AI-generated images might be saved in different formats, undergo additional compression, or be edited after generation. Each of these factors further obfuscates any patterns that could be used for classification. As a result, the detection task must not only account for the source model but also for all post-processing steps applied to the images. This introduces a domain adaptation problem, where the distribution of real-world inputs does not perfectly match the training data.

2.1.2. Existing Detection Methods and Their Limitations

Various detection methods have been proposed in the literature. Early approaches focused on pixel-level noise analysis, frequency domain analysis, or identifying GAN-specific artifacts such

as checkerboard patterns from deconvolution operations. These were later extended by more sophisticated deep learning methods that use convolutional neural networks (CNNs) to automatically learn discriminative features.

One such notable approach is proposed in the work [ZXL+24], which utilized texture inconsistency across high-detail and low-detail regions of an image to detect GAN-based fakes. The intuition was that real cameras tend to introduce uniform noise patterns regardless of image complexity, whereas generative models introduce noise depending on the amount of detail in areas of the image. While this method was effective for older generation models, it has shown to be less reliable for current diffusion models. This can be observed in the noise patterns shown in Fig. 1. The image reveals no clear differences in the noise distributions between low and high-texture areas in modern AI-generated images, which are present in the image generated with an older model. Therefore, the models will not be trained on this proposed method.

Fig. 1: Images and their noise patterns. From left to right: FLUX generated image, real image, Recraft generated image

Another class of methods involves training large-scale binary classifiers that distinguish between real and fake images. While these models, particularly when fine-tuned on specific data, have achieved high accuracy on seen data, their generalizability remains poor. Models trained on StyleGAN images, for example, fail to detect diffusion-generated content due to the architectural differences of these models. This highlights the inherent overfitting risks and the challenge of

creating universal detectors.

The most promising solutions proposed in recent years involve using large-scale vision transformers like CLIP (Contrastive Language-Image Pre-Training), which can take advantage of extensive pretraining and generalize better across tasks. These models, originally trained for text-image alignment, have demonstrated a strong ability to adapt to downstream classification tasks, including synthetic image detection, especially when fine-tuned appropriately. However, they still exhibit performance degradation when the test data diverge significantly from the training distribution, as seen in this study.

2.1.3. Characteristics of AI-Generated Images

A core component of the detection task lies in understanding what makes an AI-generated image distinguishable. Common characteristics often include:

- Texture uniformity: Early AI-generated images had texture inconsistencies that made them detectable, especially in hair, skin, or background regions. Current models minimize this artifact through advanced denoising steps.
- Object boundaries: Imperfect segmentation and unnatural blending between foreground and background remain a subtle cue in some generated images.
- Semantic inconsistencies: Generated images may feature structural anomalies (e.g., distorted hands or nonsensical reflections), but recent models trained on large datasets and with improved prompt tuning capabilities often eliminate these issues.
- Color distribution: Some generators exhibit a unique color palette or oversaturation. While this trait can be model-specific, it provides limited generalization potential.

It is worth noting that many of these indicators are either too subtle for consistent model training or are eliminated through post-processing or prompt refinement. This necessitates a robust feature extraction mechanism, capable of abstracting and comparing global visual patterns across domains.

2.1.4. Model Generalization Challenge

From an analytical standpoint, the challenge of generalizing across different generative models shares similarities with domain generalization and transfer learning tasks. When training on

a narrow domain (e.g., images from a single generator), models tend to overfit to superficial features specific to that domain. Without domain-invariant feature learning, the model cannot maintain performance on out-of-distribution samples, such as those from new generators or those modified through JPEG compression and resizing.

This work takes a domain-aware approach by using separate validation datasets, including images generated from unseen diffusion models (Ideogram and MidJourney) and a more challenging generator (Recraft), whose architecture likely deviates slightly from the training data. The performance gap across these generators serves as an empirical indicator of generalization strength.

Moreover, increasing robustness often comes at the cost of accuracy on seen data. This study also explores a trade-off between classification precision and generalization ability. By modifying input resolution and applying augmentation strategies that emulate real-world post-processing, the study aims to simulate a broader distribution of possible image scenarios.

2.2. Design of the Solution

The primary goal of this work is to investigate whether a single neural network-based model can reliably distinguish between AI-generated and real photographic images, even when the AI-generated content originates from previously unseen generators. As highlighted in the analytical section, a major challenge in this task is ensuring the model's ability to generalize. Therefore, the design of the solution prioritizes robustness over in-distribution performance.

2.2.1. Data Collection

To enable robust training and evaluation of neural network models, a comprehensive dataset was curated, comprising 2,060 images: 1,180 AI-generated and 880 real photographs. The dataset was designed to balance diversity and reliability, capturing a wide range of image styles, resolutions, and generative techniques to reflect real-world scenarios.

2.2.1.1. Real Photographs

Real images were sourced from reputable image-hosting platforms: PxHere, Pixabay, Pexels, and Unsplash. These platforms were selected due to their stringent content moderation policies, which minimize the risk of including AI-generated images mislabeled as real. Images were chosen to represent diverse categories, including landscapes, portraits, urban scenes, and objects, ensuring broad coverage of visual content. A manual verification process was employed, where each image was inspected for authenticity markers (e.g., natural lighting, camera noise) to further reduce mi-

slabeling risks. Approximately 220 images were collected from each platform, resulting in 880 real photographs.

2.2.1.2. AI-Generated Images

AI-generated images were collected from platforms hosting generative model outputs: Civitai[7], Leonardo[8], Ideogram[9], MidJourney, and Recraft[10]. Civitai was the primary source due to its extensive repository of images generated by various models, including customized Stable Diffusion variants. Images from Leonardo, Ideogram, MidJourney, and Recraft were included to capture outputs from state-of-the-art diffusion models and other proprietary architectures. The training dataset included 880 AI-generated images from Civitai and Leonardo, while 300 images from Ideogram, MidJourney, and Recraft were reserved for the unseen dataset to evaluate generalization. Images were selected to match the diversity of real photographs, covering similar categories and resolutions.

2.2.1.3. Dataset Composition

The dataset was balanced to ensure equal representation of real and AI-generated images in the training set, with 880 images per class. The unseen dataset comprised 300 AI-generated images (100 from each of Ideogram, MidJourney, and Recraft) to test model performance on novel generators. Table 3 summarizes the dataset composition, and examples of each image group can be seen in Fig. 2.

Table 3: Dataset composition for training and evaluation.

Category	**Source**	**Training Set**	**Testing Set**	**Total**
Real Photographs	PxHere, Pixabay, Pexels, Unsplash	720	160	880
AI-Generated (Training)	Civitai, Leonardo	720	160	880
AI-Generated (Unseen)	Ideogram, Midjourney, Recraft	0	300	300
Total		1,440	620	2,060

[7]https://civitai.com
[8]https://leonardo.ai
[9]https://ideogram.ai
[10]https://www.recraft.ai

Fig. 2: First row shows fake images from the training dataset. Second row shows fake images from unseen dataset. Third row shows real images from the dataset

2.2.1.4. Challenges and Mitigation

A key challenge was ensuring the authenticity of real images, given the proliferation of AI-generated content on online platforms. To mitigate this, only images with verifiable metadata (e.g., EXIF data indicating camera models) were included when available. For AI-generated images, diversity across generative models was prioritized to prevent overfitting to specific architectures.

2.2.2. Data Preprocessing and Augmentation

To prepare the dataset for training and enhance model robustness, images underwent preprocessing and augmentation. These steps were critical to normalize inputs, preserve relevant features, and prevent overfitting.

2.2.2.1. Preprocessing

Images were preprocessed to ensure compatibility with model input requirements while retaining discriminative features. The preprocessing pipeline included:

- Resizing: Images were resized to 512x512 pixels for Configurations 2 and 4, and 1024x1024 pixels for Configuration 3, using bilinear interpolation to minimize feature loss. Resizing ensured uniformity while balancing computational efficiency and detail preservation.

- Normalization: Pixel values were normalized to the range [0, 1] and standardized using mean and standard deviation values from the ImageNet dataset, aligning with the pre-training data of CLIP and DINOv2 models.
- Cropping: For Configurations 1, multiple 224x224 crops were extracted from each image during training to increase data variability and capture local features. Crops were randomly selected to simulate real-world variations.

2.2.2.2. Augmentation

Data augmentation was applied to enhance model generalization and robustness to real-world image variations (e.g., compression artifacts, scaling). The following techniques were used, with parameters randomly sampled within specified ranges:

- Scaling: Images were randomly scaled between 0.5x and 2.0x their original size to simulate resolution variations.
- JPEG Compression: Random JPEG compression with quality levels between 75 and 95 was applied to mimic real-world image degradation.
- Gaussian Blur: A Gaussian blur with a kernel size of 1 to 3 was applied to test model resilience to noise reduction.
- Flipping and Rotation: Images were randomly flipped horizontally and rotated by a multiple of 90 degrees.

These augmentations were chosen based on prior research [WWZ+20], which demonstrated their effectiveness in improving detection model performance. However, strong Gaussian blur (sigma > 2) was avoided, as noted in [OLL23], to prevent degrading image quality beyond realistic scenarios.

2.2.3. Model Selection

Three neural network models were selected for this study: CLIP-ViT-B-32[11], CLIP-ViT-L-14, and DINOv2[12]. These models were chosen for their state-of-the-art performance in image classification and their ability to capture global image features, as demonstrated in [OLL23].

[11] https://github.com/openai/CLIP
[12] https://dinov2.metademolab.com/

2.2.3.1. CLIP-CiT Models

CLIP (Contrastive Language-Image Pre-training) models, developed by OpenAI, leverage Vision Transformer (ViT) architectures pre-trained on a large-scale dataset of image-text pairs. CLIP-ViT-B-32 and CLIP-ViT-L-14 differ in size and complexity:

- CLIP-ViT-B-32: A smaller model with 32x32 patch sizes, suitable for faster training and lower computational requirements.
- CLIP-ViT-L-14: A larger model with 14x14 patch sizes, offering higher capacity to capture fine-grained features.

Both models were chosen due to their ability to generalize across diverse image domains, as shown in [OLL23], and their pre-trained weights, which reduce training time. The input size was modified to 1024x1024 in Configuration 3, with positional embedding interpolation to preserve pre-trained knowledge.

2.2.3.2. DINOv2

DINOv2, developed by Meta AI, is a self-supervised Vision Transformer model trained on large-scale image datasets. It was selected for its robustness to varying input resolutions and its ability to capture global and local image features. DINOv2 was tested with a 1024x1024 input size to assess its performance compared to CLIP models.

2.2.3.3. Rationale

Vision Transformers were prioritized over convolutional neural networks (CNNs) due to their superior performance in capturing global image relationships, as noted in [OLL23]. Pre-trained models were used to leverage learned features, reducing the need for extensive training data and computational resources.

CLIP is not inherently a binary classifier; rather, it embeds visual and textual inputs into a shared latent space for similarity comparison. However, its deep visual encoder has proven to be highly generalizable when repurposed for image classification tasks. The design uses this property by appending a lightweight classification head—a single linear layer with a sigmoid activation—to the CLIP image encoder.

2.3. Training Procedure

The training process was conducted using Google Colab with an L4 GPU, leveraging PyTorch for model implementation and the Pillow library for image processing. The procedure involved hyperparameter tuning, layer freezing, and iterative training to optimize model performance.

2.3.1. Hyperparameter Configuration

The AdamW optimizer was used with the following hyperparameters:

- Learning Rate: 1e-6 for CLIP models, reduced to 5e-8 when performance plateaued; 1e-5 for DINOv2, reduced to 5e-7.
- Weight Decay: Varied between 0.01 and 0.1 to control overfitting, with higher values used in Configuration 3.
- Batch Size: 16 to 32 images, depending on the configuration, with in one configuration multiple crops per image to increase effective batch size.
- Epochs: Models were trained for 4–15 epochs, depending on the configuration, with early stopping if validation loss did not improve for three epochs.

Loss was computed using the cross-entropy function, averaged across multiple crops per image in Configurations 1 and 2.

2.3.2. Layer Freezing

With some configurations to preserve pre-trained weights, initial training froze all but the final layers of CLIP models. Layers were progressively unfrozen as training progressed, with the learning rate reduced to fine-tune deeper layers. DINOv2 used a similar strategy, with fewer layers frozen due to its self-supervised pre-training.

2.3.3. Training Workflow

The training workflow is summarized in the following pseudocode:

2.3.4. Configuration 1

Both CLIP-B-32 and CLIP-L-14 models were trained on multiple 224x224 crops taken from augmented images at their original resolution. This approach preserved noise patterns from cameras

Algorithm 1 Pseudocode of model training

Input: Dataset D (real and AI-generated images), Model M (CLIP or DINOv2)
Output: Trained model M'

1. Initialize M with pre-trained weights
2. Freeze all layers except the final classification layer
3. Set hyperparameters: learning_rate = 1e-6, weight_decay = 0.01, batch_size = 32
4. For each epoch (1 to 15):
   (a) Apply augmentations (scaling, JPEG compression, Gaussian blur, flipping)
   (b) For each batch in D:
      i. Extract multiple 224x224 crops (Configuration 1)
      ii. Compute forward pass and cross-entropy loss
      iii. Update model weights using AdamW optimizer
   (c) Evaluate validation loss on seen and unseen datasets
   (d) If validation loss plateaus:
      i. Reduce learning_rate by factor of 0.5
      ii. Unfreeze additional layers
5. Save model M' with best validation accuracy

and generators, which are critical for detection. The models trained for only 4 epochs and achieved almost identical results, with CLIP-L-14 achieving 95% accuracy on seen generators and images generated with Ideogram and MidJourney, but achieved only 45% accuracy on Recraft images. However, the models trained this way are not very robust and would not be applicable in many real-life scenarios, because they incorrectly classify images that have undergone stronger scaling than the one used during training. This was attributed to the reliance on resolution-specific features, which varied between real and generated images.

The training curve of the CLIP-B-32 model can be seen in Fig. 3.

2.3.5. Configuration 2

To improve robustness, images were scaled down to the same size of 512x512 before applying augmentations and then preprocessed to a size of 224x224. Both CLIP-B-32 and CLIP-L-14 models were trained with pre-trained weights, and layer freezing was applied to preserve the pre-trained weights, with progressive unfreezing and lowering of the learning rate as training progressed. The models trained for around 15 epochs. The CLIP-L-14 model achieved around 93% accuracy on seen generators, around 75% accuracy on images generated with Ideogram and Midjourney, and

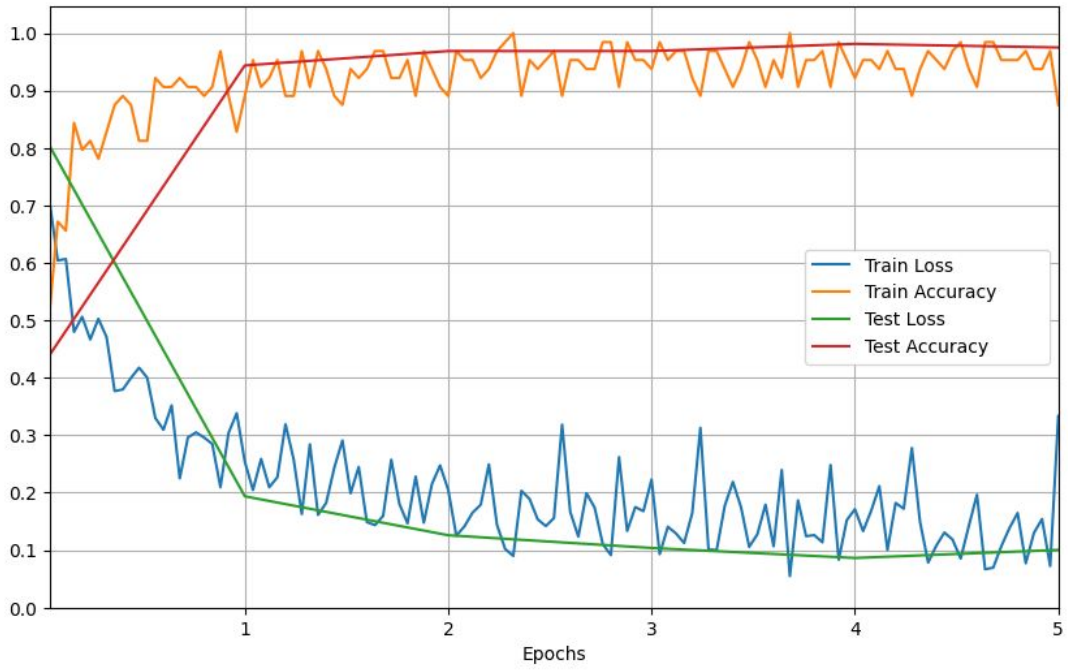

Fig. 3: The training curve of CLIP-ViT-B-32

around 35% accuracy on images generated with Recraft. The CLIP-B-32 model performed worse, by around 4% on average, which is to be expected, as it is a smaller model. While these results are worse than the ones achieved in the previous configuration, the models trained this way are more robust to images of any size. However, because the images are scaled down so much, all the noise left by the post-processing algorithms of cameras and image generation procedures is no longer present in the images, so the model is likely not classifying the image by noise, but by other features, such as color or texture features. Furthermore, without layer freezing, the models do not train and achieve accuracy close to 50%, showing that pre-training, even if for a different task, improves the models' ability to fit to new data.

The training curves of the models can be seen in Fig. 4 and Fig. 5.

2.3.6. Configuration 3

To mitigate information loss from scaling down images to 224 by 224 resolution, the input size was modified to 1024x1024 for CLIP models. And positional embedding interpolation was used to maintain some of the information from pre-training. However, the models did not outperform the results achieved in previous configurations. The models achieved only 85% accuracy on seen generators despite 99% training accuracy, indicating overfitting. Increasing weight decay reduced overfitting but also lowered validation accuracy, suggesting that either more data or more aggressive augmentations were needed.

Because CLIP models are not well suited for input size modification, a DINOv2 model, whi-

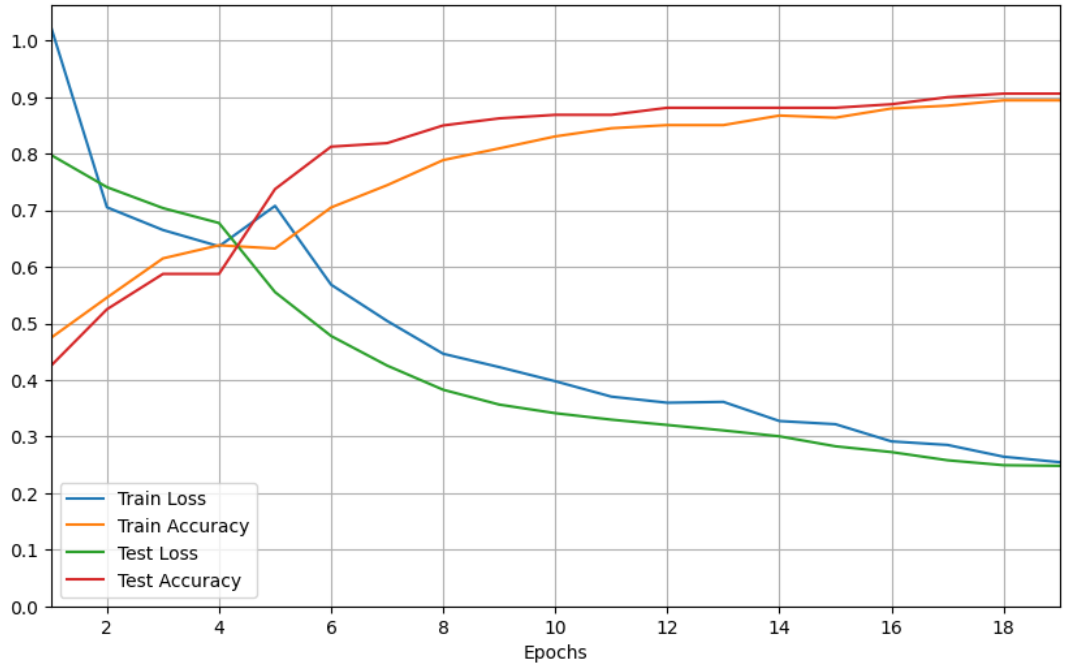

Fig. 4: The training curve of CLIP-ViT-B-32

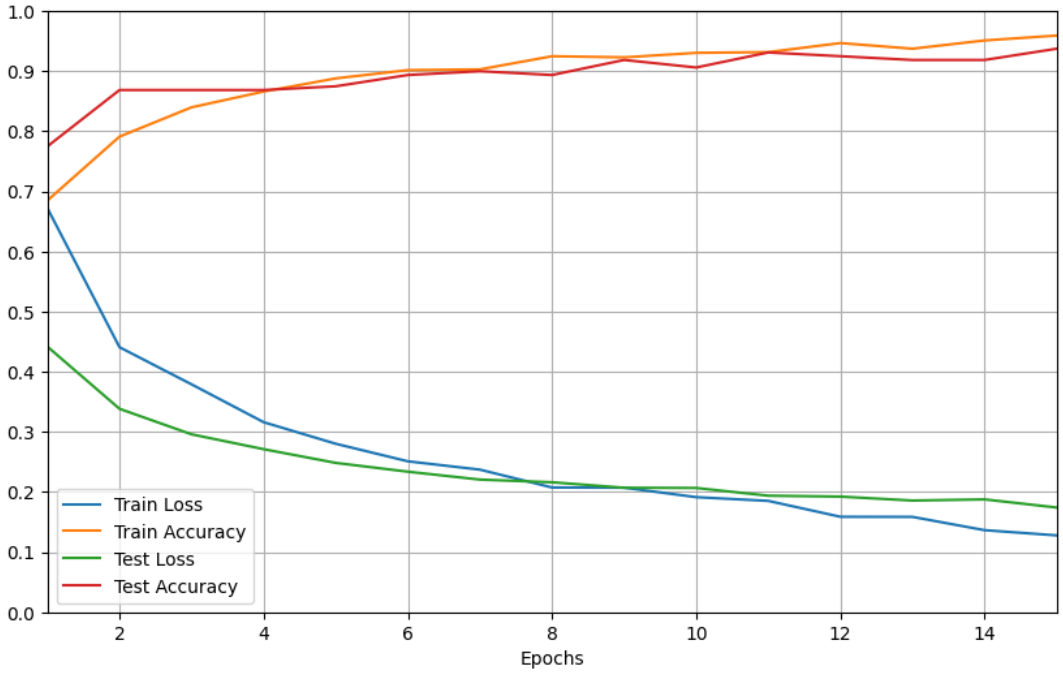

Fig. 5: The training curve of CLIP-ViT-L-14

ch is better suited for different image sizes, was also trained and tested with an input size of 1024. However, the model did not achieve higher than 60% accuracy and performed worse than CLIP models with modified input sizes, likely due to its sensitivity to fine-tuning parameters not optimized for this task.

2.3.7. Configuration 4

To address the poor classification accuracy on images generated by Recraft, the training dataset was expanded to include images generated by Ideogram and MidJourney. The CLIP-L-14 model was fine-tuned with layer freezing, achieving 91% accuracy on seen generator images and 63% on images generated with Recraft.

The training curve (Fig. 6) shows improved generalization, but performance on Recraft remains lower than desired, likely due to its unique architectural features. Accuracy achieved on images generated by Recraft is represented by ”Unseen Test Accuracy” curve.

Summary of the different configurations can be seen in Table 4.

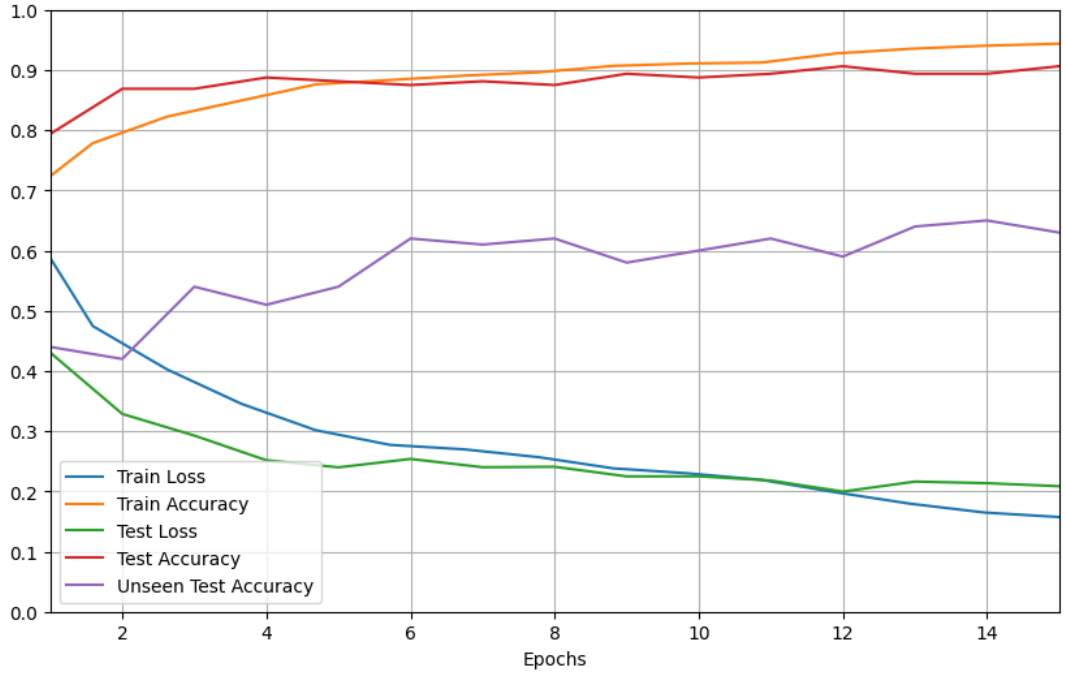

Fig. 6: The training curve of CLIP-ViT-L-14

2.3.8. Model Performance Testing

Model performance was evaluated using accuracy and loss metrics on both seen and unseen datasets. Loss was calculated using cross-entropy loss, averaged across crops or single predictions. The seen dataset included images from Civitai and Leonardo, while the unseen dataset comprised images from Ideogram, MidJourney, and Recraft. Additional metrics were not calculated because the classes are balanced.

The testing dataset was split from the training set (320 out of 1,760 images) to monitor training progress. The unseen dataset was tested separately to assess generalization. Each image was evaluated using the average prediction across multiple crops (Configuration 1) or a single forward pass (Configurations 2, 3, and 4). Models were tested on 100 images per generator (Ideogram,

Table 4: Different model training configurations

Configuration	Input Size	Augmentation	Layer Freezing	Epochs	Models Tested
1	Original (cropped to 224x224)	Scaling, JPEG, blur, flip, rotate	None	5	CLIP-B-32, CLIP-L-14
2	512x512 (preprocessed to 224x224)	Scaling, JPEG, blur, flip, rotate	Progressive	15	CLIP-B-32, CLIP-L-14
3	1024x1024	Scaling, JPEG, blur, flip, rotate	Progressive	10	CLIP-B-32, CLIP-L-14, DINOv2
4	512x512 (preprocessed to 224x224)	Scaling, JPEG, blur, flip, rotate	Progressive	15	CLIP-L-14

MidJourney, Recraft) to evaluate model generalization and analyze architectural differences of the generators.

Results

The study's findings provide valuable insights into the challenges and possibilities of creating a universal model to distinguish AI-generated images from real ones. The key results are summarized as follows:

1. Curated a dataset comprising a total of 2,060 images: 880 real photographs, 880 AI-generated images for training, and 300 AI-generated images from generators not used during training.
2. The CLIP-ViT-L-14 model, fine-tuned with augmentation and layer freezing, was implemented as a binary classifier, achieving 93% accuracy on seen generators (Civitai, Leonardo) and 75% on unseen diffusion-based generators (Ideogram, MidJourney) in Configuration 2.
3. The model struggled with Recraft images (35% accuracy in Configuration 2, 63% in Configuration 4), indicating limitations in universal detection due to architectural differences.
4. The model trained for 15 epochs, each epoch taking approximately 10 minutes on a Google Colab L4 GPU.

Performance Metrics

The table below summarizes performance across configurations:

Table 5: Model training results

Configuration	**Model**	**Seen Accuracy (%)**	**Ideogram/ MidJourney Accuracy (%)**	**Recraft Accuracy (%)**	**Training Time (min)**
1	CLIP-B-32	95	94	44	40
1	CLIP-L-14	95	95	45	50
2	CLIP-B-32	89	72	31	120
2	CLIP-L-14	93	75	35	150
3	CLIP-B-32	79	61	30	80
3	CLIP-L-14	85	65	30	100
3	DINOv2	60	55	25	150
4	CLIP-L-14	91	N/A	63	150

Analysis of Results

The high accuracy on seen generators (up to 95%) demonstrates the models' ability to learn distinctive features of known generative models. The 75-95% accuracy on Ideogram and MidJourney images suggests partial generalization to diffusion-based unseen generators, likely due to shared architectural similarities with training data. However, the low accuracy on Recraft images (35–63%) highlights significant challenges in generalizing to models with unique or proprietary architectures.

Configuration 1 achieved high accuracy but lacked robustness to scaling, making it impractical for real-world scenarios. Configuration 2 balanced performance and robustness, though scaling eliminated noise-based features completely. Configuration 3's overfitting indicates that larger input sizes require more data and more computational power. Configuration 4 improved Recraft performance by including similar unseen generators in training, but the gap persisted, suggesting Recraft's artifacts differ significantly from diffusion models.

Comparative Analysis

Comparing the models, CLIP-ViT-L-14 consistently outperformed CLIP-ViT-B-32 by 3–5% across configurations, likely due to its larger capacity and finer patch size. DINOv2 performed poorly (maximum 60% accuracy), possibly due to its self-supervised pre-training being less aligned with the binary classification task. Configuration 2 offered the best balance of accuracy and robustness, making it the most practical for real-world deployment. The low Recraft accuracy across all configurations highlights the need for broader training data diversity, as discussed in [WBZ+23].

These results emphasize the feasibility of high accuracy on known and similar unseen generators (e.g., diffusion-based) but expose significant challenges in generalizing to architecturally distinct generators like Recraft. The inclusion of additional generators in training (Configuration 4) improved performance, suggesting that dataset diversity is critical for universal detection models.

Discussion

The findings of this study provide critical insights into the challenges of developing a universal neural network model for distinguishing AI-generated images from real photographs. The results highlight both the potential and limitations of Vision Transformer (ViT)-based models, particularly in generalizing to unseen generative architectures. This section interprets the findings, discusses their implications, addresses limitations, and proposes directions for future research.

Interpretation of Findings

The CLIP-ViT-L-14 model, particularly in Configuration 2, achieved high accuracy (93%) on images from known generators (Civitai, Leonardo) and moderate accuracy (75%) on unseen diffusion-based generators (Ideogram, MidJourney). This performance aligns with prior research [WBZ+23; XWM+23], which noted that diffusion models produce similar artifacts, enabling detectors trained on one diffusion model to generalize to others. The shared architectural characteristics of diffusion models, such as iterative noise reduction, likely result in consistent frequency domain patterns that ViT models can exploit.

However, the model's low accuracy (35% in Configuration 2, 63% in Configuration 4) on Recraft-generated images underscores a significant challenge: generalizing to generators with distinct architectures. Recraft's proprietary model, which may not rely on diffusion techniques, produces images with fewer detectable artifacts, as evidenced by the noise pattern analysis in Figure 2. This finding supports [ZXL+24], which observed that newer generators minimize texture differences between rich and poor regions, rendering traditional detection methods less effective. The improved Recraft accuracy in Configuration 4, after including Ideogram and MidJourney in training, suggests that dataset diversity can partially mitigate this issue, but not fully resolve it. 7 The superior performance of CLIP-ViT-L-14 over CLIP-ViT-B-32 and DINOv2 can be attributed to its larger capacity and finer patch size, which enable better capture of global and local image features. DINOv2's poor performance (60% maximum accuracy) may stem from its self-supervised pre-training, which prioritizes general feature extraction over task-specific discrimination. This highlights the importance of aligning pre-training tasks with the target application, as noted in [OLL23].

Limitations

Several limitations impact the study's findings and their generalizability:

- Dataset Diversity: The dataset, while diverse, was limited to 2,060 images. This constrained the model's exposure to the full spectrum of generative architectures, particularly newer models like Recraft. Larger datasets should be considered to improve generalization.
- Computational Constraints: Training on Google Colab with an L4 GPU limited the model complexity and batch size. More powerful hardware could enable exploration of deeper architectures or larger input sizes without overfitting.
- Rapid Evolution of Generative Models: The fast pace of advancements in generative technologies, as discussed in [LHB+23], means that detection models risk becoming obsolete. Recraft's superior image quality exemplifies this challenge.
- Feature Identification: The study could not identify universal features distinguishing all AI-generated images, as hypothesized. The absence of consistent artifacts across generators, as noted in [OLL23], complicates universal detection.

Implications

The study's findings have significant implications for combating misinformation and ensuring digital authenticity. High accuracy on known and diffusion-based generators suggests that ViT-based models can be deployed in platforms like social media to flag AI-generated content, reducing the spread of deepfakes and manipulated images. However, the poor performance on Recraft indicates that current models are not yet reliable for universal deployment, particularly against emerging generators. Organizations implementing detection systems must continuously update training datasets to include new models, as demonstrated by Configuration 4's improvement.

The research contributes to the growing body of knowledge on AI-generated image detection by validating the efficacy of ViT architectures, consistent with [OLL23]. It also highlights the limitations of frequency domain-based detection, as newer generators reduce detectable artifacts.

Conclusions

This study investigated the feasibility of developing a universal neural network model to distinguish AI-generated images from real photographs, addressing a critical challenge in maintaining digital authenticity. The findings provide valuable insights into the potential and limitations of universal detection models, with implications for both practical applications and theoretical advancements. The key conclusions are as follows:

1. It is harder to detect images generated with Recraft because they have distinct features that greatly differ from images generated by other generative models.

2. There are no common features that are consistent throughout images generated by different generators.

3. It is not possible to create a universal classification model because features produced by different generative models are unique in how they differ from photographs.

While a truly universal model remains elusive, the study demonstrates that high accuracy is achievable for known and similar generators when trained on diverse datasets. This has practical implications for deploying detection systems in digital platforms to combat misinformation, provided training data is continuously updated to reflect new generative technologies.

Recommendations for Future Research

To advance the development of universal detection models, researchers should focus on the following:

- Larger and More Diverse Datasets: Incorporate datasets which include images from a broader range of generators. This would enhance model exposure to diverse artifacts and improve generalization.

- Hybrid Architectures: Explore hybrid CNN-ViT models or ensemble approaches to combine local feature extraction (CNNs) with global relationship modeling (ViTs), potentially improving detection of subtle artifacts.

- Unsupervised Learning: Adopt unsupervised techniques, as proposed in [QSX+24; ZWH+22], to handle noisy labels and reduce dependency on curated datasets. This could improve robustness to mislabeled or novel data.

- Real-Time Detection: Develop lightweight models optimized for real-time deployment on resource-constrained devices, ensuring practical applicability in online platforms.
- Continuous Learning Systems: Implement continual learning frameworks to adapt models to new generative models without retraining from scratch, addressing the rapid evolution of generative technologies.

These directions aim to bridge the gap between current detection capabilities and the evolving landscape of generative models, fostering more robust and universal detection systems.

In conclusion, while universal detection of AI-generated images is challenging due to the rapid evolution and diversity of generative models, this study demonstrates that robust detection is feasible for known and similar architectures with appropriate dataset design and model training. By addressing the identified limitations and pursuing the proposed research directions, future work can move closer to reliable solutions for maintaining authenticity in digital environments.

References

[AGV18] E. Athanasiadou, Z. Geradts and E. Van Eijk. Camera Recognition with Deep Learning. *Forensic Sciences Research*, 3(3):210–218, 2018-10. ISSN: 2096-1790. DOI: 10.1080/20961790.2018.1485198. eprint: https://academic.oup.com/fsr/article-pdf/3/3/210/46756930/fsr_3_3_210.pdf. URL: https://doi.org/10.1080/20961790.2018.1485198.

[BLY+23] Xiuli Bi, Bo Liu, Fan Yang, Bin Xiao, Weisheng Li, Gao Huang, and Pamela C. Cosman. Detecting generated images by real images only. https://arxiv.org/abs/2311.00962, 2023.

[CCZ+23] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano and L. Verdoliva. On The Detection of Synthetic Images Generated by Diffusion Models. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5, 2023. DOI: 10.1109/ICASSP49357.2023.10095167.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, p. 248–255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[FES+20] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz. Leveraging frequency analysis for deep fake image recognition. https://arxiv.org/abs/2003.08685, 2020.

[HCM+24] J. Huang, C. Chen, A. Mishra, B. C. Kwon, Z. Liu, and C. Bryan. Asap: interpretable analysis and summarization of ai-generated image patterns at scale. https://arxiv.org/abs/2404.02990, 2024.

[HFC+24] Y. Hong, J. Feng, H. Chen, J. Lan, H. Zhu, W. Wang, and J. Zhang. Wildfake: a large-scale challenging dataset for ai-generated images detection. https://arxiv.org/abs/2402.11843v1, 2024.

[HSW+21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: low-rank adaptation of large language models. https://arxiv.org/abs/2106.09685, 2021.

[YDF19] N. Yu, L. Davis and M. Fritz. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, p. 7555–7565, 2019. DOI: 10.1109/ICCV.2019.00765.

[YHC+22] T. Yang, Z. Huang, J. Cao, L. Li, and X. Li. Deepfake network architecture attribution. https://arxiv.org/abs/2202.13843, 2022.

[YSZ+16] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: construction of a large-scale image dataset using deep learning with humans in the loop. https://arxiv.org/abs/1506.03365, 2016.

[JJK+22] Y. Ju, S. Jia, L. Ke, H. Xue, K. Nagano and S. Lyu. Fusing Global and Local Features for Generalized AI-Synthesized Image Detection. *2022 IEEE International Conference on Image Processing (ICIP)*, p. 3465–3469, 2022. DOI: 10.1109/ICIP46576.2022.9897820.

[KFK+18] A. Kuzin, A. Fattakhov, I. Kibardin, V. Iglovikov, and R. Dautov. Camera model identification using convolutional neural networks. https://arxiv.org/abs/1810.02981, 2018.

[LHB+23] Z. Lu, D. Huang, L. Bai, J. Qu, C. Wu, X. Liu, and W. Ouyang. Seeing is not always believing: benchmarking human and model perception of ai-generated images. https://arxiv.org/abs/2304.13023, 2023.

[LLW+15] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. https://arxiv.org/abs/1411.7766, 2015.

[LMB+15] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, et al. Microsoft coco: common objects in context. https://arxiv.org/abs/1405.0312, 2015.

[LQT20] Zhengzhe Liu, Xiaojuan Qi and Philip H.S. Torr. Global Texture Enhancement for Fake Face Detection in the Wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 8057–8066, 2020. DOI: 10.1109/CVPR42600.2020.00808.

[MGP24] A.G. Moskowitz, T. Gaona, and J. Peterson. Detecting ai-generated images via clip. https://arxiv.org/abs/2404.08788, 2024.

[MTB22] J. Materzynska, A. Torralba, and D. Bau. Disentangling visual and written concepts in clip. https://arxiv.org/abs/2206.07835, 2022.

[OLL23] U. Ojha, Y. Li and Y. J. Lee. Towards Universal Fake Image Detectors that Generalize Across Generative Models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 24480–24489, 2023. DOI: 10.1109/CVPR52729.2023.02345.

[Pen20] C. J. D. Penedo. Predict the model of a camera. https://arxiv.org/abs/2004.03336, 2020.

[QSX+24] T. Qiao, H. Shao, S. Xie and R. Shi. Unsupervised Generative Fake Image Detector. *IEEE Transactions on Circuits and Systems for Video Technology*:1–1, 2024. DOI: 10.1109/TCSVT.2024.3383833.

[RCV+19] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner. FaceForensics++: Learning to Detect Manipulated Facial Images. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, p. 1–11, 2019. DOI: 10.1109/ICCV.2019.00009.

[RLJ+22] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: fine tuning text-to-image diffusion models for subject-driven generation. https://arxiv.org/abs/2208.12242, 2022.

[RPS+23] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, and S. A. Fattah. Artifact: a large-scale dataset with artificial and factual images for generalizable and robust synthetic image detection. https://arxiv.org/abs/2302.11970, 2023.

[SME20] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. https://arxiv.org/abs/2010.02502, 2020.

[T K19] T. Aila T. Karras S. Laine. A style-based generator architecture for generative adversarial networks. https://arxiv.org/abs/1812.04948, 2019.

[TZW+23] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu and Yunchao Wei. Learning on Gradients: Generalized Artifacts Representation for GAN-Generated Images Detection. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 12105–12114, 2023. DOI: 10.1109/CVPR52729.2023.01165.

[WBZ+23] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen and H. Li. DIRE for Diffusion-Generated Image Detection. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, p. 22388–22398, 2023. DOI: 10.1109/ICCV51070.2023.02051.

[WMM+22] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau. Diffusiondb: a large-scale prompt gallery dataset for text-to-image generative models. https://arxiv.org/abs/2210.14896, 2022.

[WWZ+20] S.-Y. Wang, O. Wang, R. Zhang, A. Owens and A. A. Efros. CNN-Generated Images Are Surprisingly Easy to Spot… for Now. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 8692–8701, 2020. DOI: 10.1109/CVPR42600.2020.00872.

[XWM+23] Q. Xu, H. Wang, L. Meng, Z. Mi, J. Yuan and H. Yan. Exposing fake images generated by text-to-image diffusion models. *Pattern Recognition Letters*, 176:76–82, 2023. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2023.10.021. URL: https://www.sciencedirect.com/science/article/pii/S0167865523002933.

[ZCY+23] M. Zhu, H. Chen, Q. Yan, X. Huang, et al. Genimage: a million-scale benchmark for detecting ai-generated image. https://arxiv.org/abs/2306.08571, 2023.

[ZHL+23] Y. Zhou, P. He, W. Li, Y. Cao and X. Jiang. Generalized Fake Image Detection Method Based on Gated Hierarchical Multi-Task Learning. *IEEE Signal Processing Letters*, 30:1767–1771, 2023. DOI: 10.1109/LSP.2023.3336570.

[ZKC19] X. Zhang, S. Karaman and S.-F. Chang. Detecting and Simulating Artifacts in GAN Fake Images. *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, p. 1–6, 2019. DOI: 10.1109/WIFS47025.2019.9035107.

[ZWC+22] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum and C.-C. Jay Kuo. A-PixelHop: A Green, Robust and Explainable Fake-Image Detector. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 8947–8951, 2022. DOI: 10.1109/ICASSP43922.2022.9747901.

[ZWH+22] M. Zhang, H. Wang, P. He, A. Malik and H. Liu. Exposing unseen GAN-generated image using unsupervised domain adaptation. *Knowledge-Based Systems*, 257:109905, 2022. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2022.109905. URL: https://www.sciencedirect.com/science/article/pii/S0950705122009984.

[ZXL+24] N. Zhong, Y. Xu, S. Li, Z. Qian, and X. Zhang. Patchcraft: exploring texture patch for efficient ai-generated image detection. https://arxiv.org/abs/2311.12397v3, 2024.