GAUNTLET: An Explainable Model for Detecting AI-Generated Images

Angelo Galavotti
University of Bologna
Master's in Artificial Intelligence
angelo.galavotti@studio.unibo.it

Lorenzo Galfano
University of Bologna
Master's in Artificial Intelligence
lorenzo.galfano@studio.unibo.it

Abstract

As AI-based image generation continues to advance, distinguishing between human-crafted and AI-generated content is becoming increasingly challenging. This poses significant risks, as this content can be exploited in malicious contexts.
GAUNTLET tackles this issue by providing an explainable system for detecting AI-generated images. In this document, we illustrate the features of GAUNTLET and analyze its results, while comparing the different explainability tools used. In addition, we present the training techniques employed and explain how they allowed us to leverage the capabilities of the model.
To demonstrate its practical applicability, we developed a web application that utilizes this model, enabling users to upload images and receive detailed insights into whether the content is AI-generated. This use case underscores the potential of GAUNTLET in addressing real-world challenges.

1 Introduction

As AI-based content generation continues to advance, distinguishing between authentic images and synthetic ones has become a significant challenge. The rapid evolution of generative models, particularly diffusion-based image generation, has introduced hyper-realistic images that are often indistinguishable from reality. Furthermore, such models have become increasingly more accessible, lowering the technical and financial barriers for individuals or groups with malicious intents. This raises important concerns about the misuse of AI-generated content and its possible implications in misinformation, identity fraud, and monetary scams.

In response to these challenges, we developed GAUNTLET, a framework aimed at detecting AI-generated images while emphasizing explainability. Our goal is to give users clear insights into the decision-making process, helping them recognize the visual cues and patterns that set images created by generative models apart from those created by humans. In particular, with this project we aim to equip users with the knowledge to identify AI-generated content on their own, rather than relying entirely on the model's conclusions.

To better showcase the ultimate goal of our project, we developed a web app that demonstrates a practical use case of the model. This interactive tool provides a hands-on experience, allowing to explore the model's capabilities and better understand its potential applications in real-world scenarios.

2 Background

2.1 The rise of generative AI

The rapid advancements in generative artificial intelligence have led to the proliferation of highly realistic synthetic images, often indistinguishable from real ones to the human eye. While this content can showcase remarkable progress in creative applications, they can potentially be a vector for malicious intents.

AI-powered fraud, such as deepfake images and voice impersonations, has become alarmingly effective. For instance, in 2023, U.S. consumers reported over $10 billion in losses due to scams, the highest annual figure recorded by the Federal Trade Commission[1]. Scammers have exploited AI-generated content for fake charity appeals and fraudulent investment schemes, capitalizing on the technology's ability to manipulate trust and deceive targets.

Businesses are also feeling the impact, with 76% of fraud and risk professionals indicating that their organizations have been targeted by AI-driven scams. Among consumers, younger demographics are disproportionately affected. Nearly 22% of individuals[2] in these groups fall for AI-enhanced fraud, despite 73% of respondents expressing confidence in their ability to identify such scams.

The growing threat of AI-powered scams highlights just how urgent it is to develop better ways to detect and understand these risks. Now more than ever there is a need for tools that can allow people to feel equipped to navigate this new reality with confidence.

2.2 Related Works

Several recent works, such as the one presented by Moskowitz et al. (2024), have explored methods to detect AI-generated content using the Contrastive Language-Image Pre-training (CLIP) architecture, which connects images and text in a shared embedding space (1). Their approach demonstrated performance comparable to, or exceeding, that of models specifically designed for this purpose.

Similarly, Xu et al. (2024) proposed the exploitations of deep trace representations and dual-branch interactive feature fusion (2). In particular, a Visual Transformer (ViT) is designed to learn the deep representations of the global trace information (i.e. the artifacts found in the AI generated image). After that, a low-level feature extraction module incorporating a channel-spatial attention block is also employed to enhance the learning of said trace representations.

2.3 Explanability tools

Explainability tools play a crucial role in bridging the gap between AI models and human understanding, and are a key aspect of this project. In particular, the explanations we adopted in GAUNTLET offer a visual representation of the focus areas of the models, and enable users to identify the distinguishing features of synthetic versus authentic images more easily.

2.3.1 Ablation-CAM and Score-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) (3) is a widely recognized explainability technique that offers visual insights into the decision-making processes of deep learning image models. By leveraging the gradients of a target class with respect to the final convolutional layer, Grad-CAM generates heatmaps that emphasize the regions of the input image most relevant to the model's prediction. While Grad-CAM is effective in many applications, it relies heavily on the backpropagation of gradients, and as such can output noisier heatmaps. Many variants of Grad-CAM have been developed throughout the years, and as such it has evolved into something more akin to a family of explainability techniques. Notably, some of these variantas avoid gradient computation altogether. In particular:

- Ablation-CAM operates by systematically ablating (i.e. removing or disabling) specific neurons in the feature maps of the convolutional layer and observing the impact on the model's output.

[1] `https://www.ftc.gov/reports/consumer-sentinel-network-data-book-2023`

[2] `https://sift.com/index-reports-ai-fraud-q2-2024?aliId=`

- Score-CAM utilizes the model's score as a measure of importance for each activation map. It perturbs the input image using different activation maps and observes how much each one contributes to the model's confidence in its prediction.

By skipping gradient computation, Ablation-CAM and Score-CAM produce more robust and well-defined heatmaps, making them particularly effective for classifiers that rely on capturing finer details. In addition, since they do not require backpropagation, Ablation-CAM and Score-CAM are more convenient to integrate into the evaluation phase of the project pipeline. For these reasons, we opted to use these tools over vanilla Grad-CAM. While we recognize that they require higher computational cost and memory in comparison, they remain feasible within the project's constraints.

In GAUNTLET, ScoreCAM and AblationCAM are set up so that the highlighted region corresponds to the area that affected the prediction of the specific label that was predicted by the network. In other words, if the model classifies an image as "FAKE", ScoreCAM will output an heatmap showing the patches of the image which caused it to predict the image as fake.

2.3.2 LIME

Local Interpretable Model-agnostic Explanations (LIME) (4) is another key tool for understanding the predictions of complex models. Unlike Grad-CAM, LIME is model-agnostic and works by perturbing the input data and observing changes in the model's output to determine which parts of the input contribute most to the prediction. For image data, LIME segments the image into superpixels and evaluates the importance of each region by modifying them (e.g., by masking or altering their content). This approach allows LIME to identify specific patches of an image that strongly influence whether the model classifies it as real or fake.

3 Datasets

3.1 CIFAKE

The CIFAKE dataset (5) is a pivotal dataset used for assessing models that detect AI-generated images. It consists of 120,000 32x32 samples, equally divided between authentic and synthetic categories. The synthetic images are generated using the Stable Diffusion 1.4 model (6), and are intentionally crafted to closely resemble real pictures. On the other hand, human crafted images of the datasets are derived from the CIFAR-10 dataset. The samples present in CIFAKE involve a diverse set of subjects, including objects, animals, and scenes.

3.2 Real vs. AI Art

The "Real vs. AI Art" dataset (RVAA) (7) is a more diverse and heterogeneous collection of 975 high-resolution images. It combines pictures sourced through web scraping and procedurally generated AI content, and it offers a broader spectrum of subjects, including animals, landscapes, and abstract visuals.

Unlike CIFAKE, this dataset presents significant variability in resolution and composition, as well as the model they were generated with, including Midjourney and DALL-E. Furthermore, some samples present images that are only in part AI-generated (e.g. an AI generated image with legible text, or put side-by-side with a crafted image). For these reasons, it poses a significant challenge for AI detection models.

3.3 MIXED

As the reader may have guessed, this dataset consists of a mixture of samples from the RVAA and CIFAKE dataset. To ensure balance, we sampled 1,200 images of CIFAKE (about 1% of its size) to match the smaller RVAA dataset. This particular experiment was designed to evaluate whether the model could learn and generalize features from both datasets, handling differences in resolution and scale effectively. In addition, we created a variant in which the RVAA samples are augmented using the same specifications as the one found in section 4.3.

4 Training setup

4.1 Models

The ResNet (8) is a well-established architecture known for its strong feature extraction capabilities, thanks to its deep layers and the use of residual connections. By addressing the vanishing gradient problem, residual connections enable the model to effectively learn rich, hierarchical representations of input data. ResNet has been widely validated on a variety of computer vision tasks, making it a dependable choice for general-purpose feature extraction. Moreover, its pretrained weights, available from large-scale datasets like ImageNet, provide an excellent starting point for transfer learning. This speeds up convergence and boosts performance, particularly on datasets with a limited amount of labeled samples (e.g. RVAA).

The main model adopted in this project consists in a fine-tuned ResNet-50, which is a deeper version of ResNet comprised of 50 layers. It is important to note that this decision was based on extensive testing and evaluation of alternative models, details of which are provided in Appendix A.

As a baseline, we designed and trained a custom CNN, which consists of three convolutional layers with increasing channel sizes (32, 64, 128), each followed by batch normalization, ReLU activation, and max pooling. The extracted features are then flattened and passed through three fully connected layers (512, 256, and the final output layer) to generate class predictions. This architecture balances simplicity and performance, providing a solid foundation for evaluating our method.

4.2 Training strategy

The strategy used for fine-tuning played a vital role in optimizing the model's performance. In particular, we adopted a method called "Linear Probing followed by Fine-Tuning" (9), which essentially consists of two steps. In the first stage, a new classification head was attached and trained, while the feature extractor layers were frozen. In the second stage, the feature extractor was unfrozen and fine-tuned, so that it could exploit the classification head of the previous step.

The hyperparameter configuration remained mostly consistent across the datasets. We used the Adam optimizer with a learning rate of $3 * 10^{-4}$, while for CIFAKE we opted for AdamW to speed up training. The batch sizes were tailored based on the size of each dataset: 128 for CIFAKE, 32 for the RVAA dataset, and 64 for the MIXED dataset. These settings were chosen following an extensive experimentation process, where we conducted a grid search to compare the losses achieved with various configurations. The same hyperparameter configuration was also employed for the respective baseline CNNs.

4.3 Data augmentation

To try and squeeze more performance out of the RVAA model, we tried to perform data augmentation as the dataset features a rather small number of samples. Essentially, we augmented about 80% of the training set, by applying random rotations of 90 degrees and flipping the image. These transformations were also applied to the 80% of the RVAA samples in the training set of the MIXED dataset, in another data augmentation experiment.

4.4 Evaluation

For the evaluation, a comprehensive set of metrics was computed to assess the models' performance in detecting AI-generated images. Accuracy and the confusion matrix were calculated first to provide an overview of the models' ability to distinguish between real and fake images.

Additionally, precision, recall, and F1-score were computed to evaluate the model's performance from a classification perspective.

5 Results

In this section, a quantitative and qualitative analysis of the results will be presented. The CNN network will be referred to as the "baseline" throughout the analysis, as it serves as the starting point for our experiments. In contrast, the ResNet-50 is presented with two possible designations: "*step1*" and "*step2*." These refer to the stages of the fine-tuning process during training. Specifically, *step1* represents the stage where only the classification head was fine-tuned, while *step2* indicates the point where both the classification head and the feature extractor were fine-tuned. The ResNet-50 architecture is also evaluated with an AUG variant, which incorporates data augmentation techniques during training.

5.1 CIFAKE results

The result in Table 1 shows a significant bump in performance from step1 to step2, with the model reaching an amazing 98% accuracy. This confirms that the fine-tuning strategy effectively enhanced the model's performance. On the other hand, the baseline performed much better than the step1 training model, as its higher capacity allowed for it to encapsulate the distinct features of images much better than just training the classifier. These results indicate that the CIFAKE dataset is well structured and, with its high number of samples, it is bound to yield very good results.

Models	Accuracy	Precision	Recall	F1-Score
CIFAKE_baseline	95.5	0.94	**0.97**	0.95
CIFAKE_ResNet-50_step1	82.7	0.83	0.83	0.83
CIFAKE_ResNet-50_step2	**98.0**	**0.97**	**0.97**	**0.97**

Table 1: Results on the CIFAKE dataset. The 2-step fine-tuning strategy achieves a very high accuracy score.

5.2 RVAA results

The RVAA dataset posed a unique challenge due to its inherent heterogeneity and the presence of ambiguous images that can sometimes be difficult for even humans to distinguish between fake and real. Despite this, the network demonstrated strong adaptability to these challenges, as evidenced by the results presented in Table 2. The baseline model achieved an accuracy of 76.7%, while the ResNet-50 model reached 81.5% accuracy after the two-step training process, showing a 3% improvement over the performance with only the first training step. Data augmentation allowed us to leverage the full potential of the dataset: we managed to achieve 86.3% accuracy score using the step2 model on the augmented dataset.

Models	Accuracy	Precision	Recall	F1-Score
RVAA_baseline	76.7	0.73	0.73	0.73
RVAA_ResNet-50_step1	78.8	0.76	0.75	0.75
RVAA_ResNet-50_step2	81.5	**0.87**	0.74	0.80
RVAA_AUG_ResNet-50_step1	80.1	0.79	0.75	0.77
RVAA_AUG_ResNet-50_step2	**86.3**	0.85	**0.83**	**0.84**

Table 2: Results on the RVAA dataset. Data augmentation is beneficial to the network, and allow us to leverage the full potential of the dataset.

5.3 MIXED results

Using the MIXED dataset, the ResNet-50 model demonstrated notable performance, achieving an accuracy of 90.3% and a higher, 91.4% accuracy using data augmentation. Due to the RVAA dataset being under-represented, only this subset was augmented, using the transformations outlined in Section 4.3. Conversely, the CIFAKE images were upscaled to match the dimensions of the RVAA dataset. The result figures are shown in Table 3.

Models	Accuracy	Precision	Recall	F1-Score
MIXED_baseline	83.1	0.92	0.86	0.89
MIXED_ResNet-50_step1	85.5	0.91	0.89	0.90
MIXED_ResNet-50_step2	90.3	**0.97**	0.90	0.93
MIXED_AUG_ResNet-50_step1	82.3	0.84	0.93	0.88
MIXED_AUG_ResNet-50_step2	**91.4**	0.93	**0.95**	**0.94**

Table 3: Results on the MIXED dataset. Data augmentation appears to slightly improve performance.

5.4 Evalutation and comparison of explanations

As mentioned in Section 2.3, we evaluated three explainability techniques: ScoreCAM, AblationCAM, and LIME. The outcomes varied depending on the backend and layers targeted by the models. Both ScoreCAM and AblationCAM produced higher-quality explanations compared to LIME.

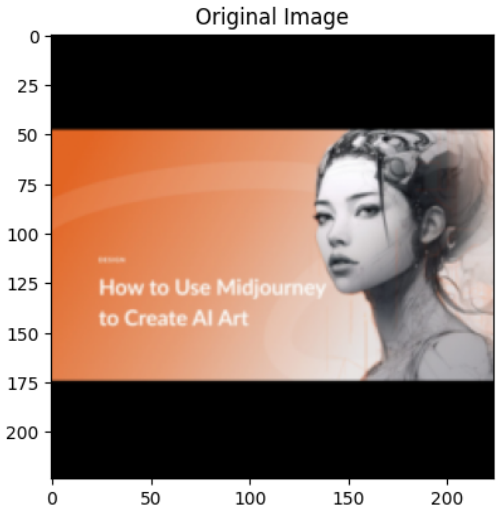

Figure 1: Original input image from RVAA

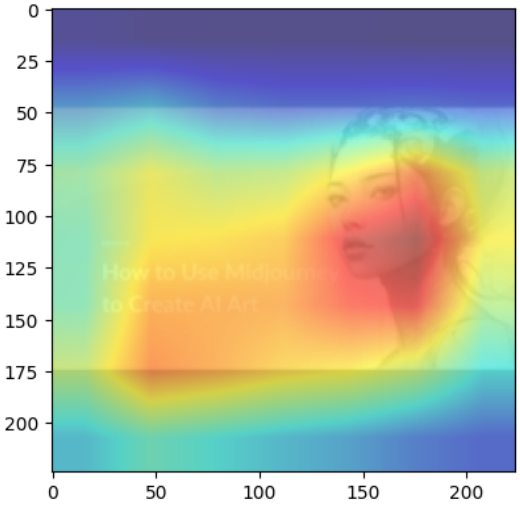

Figure 2: ScoreCAM on layer 4

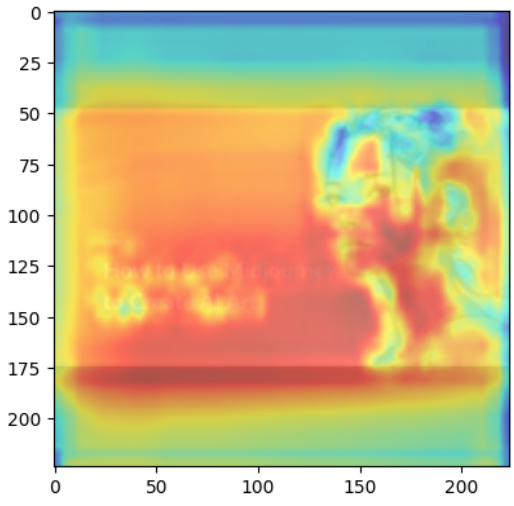

Figure 3: ScoreCAM on layer 2,3,4

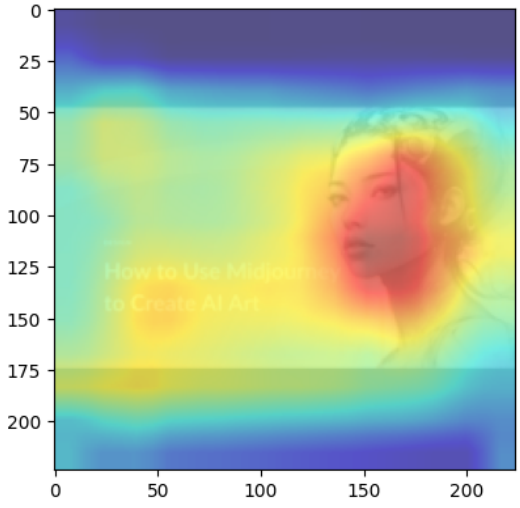

Figure 4: AblationCAM on layer 4

When using ScoreCAM and AblationCAM, we targeted layers 2, 3, and 4 collectively, as well as layer 4 individually. The former method generated more defined heatmaps overall, however this was not

always a positive. This is can clearly be seen in Figure 2 and 3. In this case, the model misclassified a fake image as "REAL". This error was understandable given the challenging features present, such as well-defined text, commonly associated with human-crafted images. The heatmap for Figure 2 highlighted the text as a key feature for the "REAL" prediction, consistent with the observation that generative models struggle to produce high-quality text. In contrast, in Figure 3, where layers 2, 3, and 4 are targeted, the heatmap, although more defined, is much harder to interpret. Between ScoreCAM and AblationCAM, results were broadly similar. However, ScoreCAM sometimes provided heatmaps that appeared more aligned with the model's decision-making process.

In contrast, LIME explanations were significantly less intuitive. The method often highlighted seemingly random patches of the image, making it difficult to discern the reasoning behind predictions. Given GAUNTLET's reliance on detailed and coherent explanations, LIME proved unsuitable for this application. We attribute this behavior to the fact LIME does not perform well with binary classifiers, and is instead more suitable for multi-category datasets.

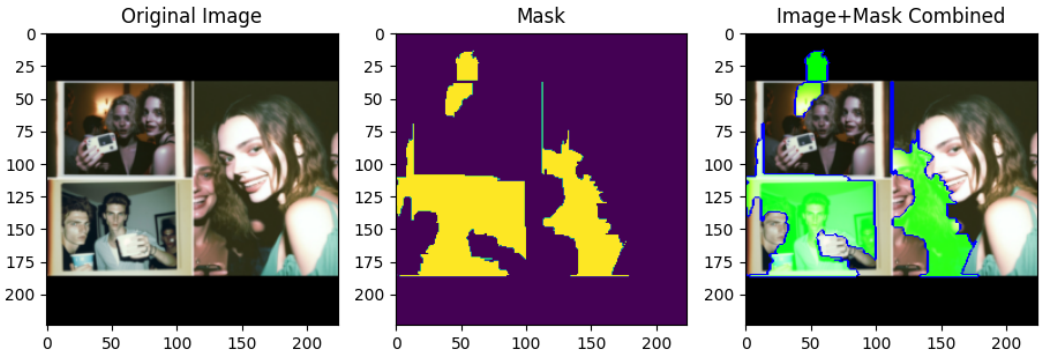

Figure 5: Example of LIME results

Dataset resolution also influenced explanation quality. Explanations for RVAA and MIXED datasets were generally better than those for CIFAKE. The low 32x32 resolution of CIFAKE images often resulted in broken explanations: either no parts of the image were highlighted, or the entire image was. This limitation affected both LIME and GradCAM variants. In contrast, the higher resolutions of RVAA and MIXED allowed for more meaningful visualizations.

5.5 Evaluation of the model with AI generated images

We evaluated the model on a set of images generated by various generative AIs, including Midjourney, DALL-E 3, and Stable Diffusion 1.4, to assess its performance in real-world scenarios. Overall, the second-step MIXED model demonstrated the most consistent predictions, and reliably detected real images across the sample set. However, all models faced challenges with images generated by Midjourney, as these were notably more realistic compared to those produced by the other generative models.

6 Web Application

We developed a web app[3] to showcase a possible use case of GAUNTLET, using React for the frontend and Flask for the backend. The app dynamically loads a list of available model checkpoints, allowing users to select the model they wish to use for predictions. Additionally, the applications provides a set of explanation methods that the users can choose: they are the same as the one we used for GAUNTLET (i.e. LIME, ScoreCAM, AblationCAM).

Upon making a query, the app processes the input through the selected model and returns a comprehensive visualization, including a heatmap or segmentation map illustrating the model's focus areas.

[3]A video demo of the web application can be found at the url `https://www.youtube.com/watch?v=THj-Gn8MYkw`

Alongside the visual explanation, the app displays the confidence scores for the REAL and FAKE classes, offering a clear and interactive demo of the model's capabilities.

7 Final Remarks

In this experiment, we evaluated the utility of building an explainable framework, beginning with accuracy assessment and extending to the visualization of explanations. Using a finetuned ResNet allowed us to achieve high scores in terms of both accuracy and f1, and proved to be the right choice for this task.

For visualization, we noted that explainability methods performed poorly on the standard CIFAKE dataset. This limitation likely stems from the small resolution of the images, which prevents the generation of meaningful explanations. In contrast, the results from the MIXED dataset, which included an upscaled version of CIFAKE, revealed more interpretable explanations.

Among the methods tested, AblationCAM and ScoreCAM provided much more clearer and interpretable visualizations to LIME. In comparison, LIME often struggled to produce meaningful explanations, likely because it is better suited for multi-class problems rather than binary classification.

7.1 Future Work

In terms of a preliminary test or proof-of-concept, we are satisfied with the results we achieved with the approach we used for GAUNTLET. Nevertheless, the overall results can still be improved in a number of ways.

Future work could focus on testing different kinds of network architectures. In particular, as shown by Moskowitz et al. (2024) (1), the use of a CLIP architecture can be very effective for detecting AI generated images. In a future extension of the project, we could attempt to apply said model to either CIFAKE or RVAA dataset. However, it must be noted that such architecture requires high computational power, and finetuning CLIP can be much more cumbersome than using a standard CNN-based model such as ResNet.

Another potential improvement lies in using a better quality dataset. The samples in CIFAKE were generated by a single model and RVAA had a small quantity of samples. For these reasons, these dataset are definitely not optimal. Instead, using a dataset such as WildFake (10), which contains images from a wide range of generative image models and has a very high number of samples (about 3 million images). It is worth mentioning that using high-resolution images and a large number of samples comes with a significantly higher computational cost. Therefore, balancing dataset quality and computational feasibility remains a key challenge for future improvements.

References

[1] A.G. Moskowitz, T. Gaona, J. Peterson (2024), *Detecting AI-Generated Images via CLIP* , https://arxiv.org/abs/2404.08788

[2] Qiang Xu, Xinghao Jiang, Tanfeng Sun, Hao Wang, Laijin Meng, Hong Yan (2024), *Detecting Artificial Intelligence-Generated images via deep trace representations and interactive feature fusion*, https://doi.org/10.1016/j.inffus.2024.102578

[3] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra (2016), *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*, https://arxiv.org/abs/1610.02391

[4] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin (2016), *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*, https://arxiv.org/abs/1602.04938

[5] Jordan J. Bird, Ahmad Lotfi (2023), *CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images*, https://arxiv.org/abs/2303.14126

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer (2021), *High-Resolution Image Synthesis with Latent Diffusion Models*, https://arxiv.org/abs/2112.10752

[7] Cash Bowman (2024), *AI Generated Images vs Real Images*, https://www.kaggle.com/datasets/cashbowman/ai-generated-images-vs-real-images

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015), *Deep Residual Learning for Image Recognition*, https://arxiv.org/abs/1512.03385

[9] Akiyoshi Tomihari, Issei Sato (2024), *Understanding Linear Probing then Fine-tuning Language Models from NTK Perspective*, https://arxiv.org/abs/2405.16747v1

[10] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, Weidi Xie (2024), *A Sanity Check for AI-generated Image Detection*, https://arxiv.org/abs/2406.19435

A Appendix A

A.1 Additional models

While searching for a suitable model to use with the datasets, we started by experimenting with modifications to the ResNet architecture on the CIFAKE dataset. Our first approach involved a modified ResNet, which we called ModResNet, where we added three additional linear layers to the standard ResNet-50 structure. We then took this a step further by incorporating Dropout into ModResNet to enhance its generalization capabilities.

Ultimately, this experiment proved to be unsuccessful: the performance showed no improvement whatsoever, while the added complexity of the network significantly increased training and inference times. As we have mentioned, these networks were tested exclusively on the CIFAKE dataset. Once their inefficiency became evident, they were discarded and not evaluated on other datasets.

Models	Accuracy	Precision	Recall	F1-Score
ModResNet_step1	85.6	0.86	0.85	0.86
ModResNet_step2	97.2	0.97	0.97	0.97
ModResNet_Dropout_step1	85.3	0.88	0.83	0.86
ModResNet_Dropout_step2	96.9	0.97	0.97	0.97

Table 4: Additional experiments done with the CIFAKE dataset.

B Appendix B

B.1 Testing a CIFAKE model on RVAA

To evaluate whether a model trained on CIFAKE could generalize to the RVAA dataset, another ResNet-50 model was trained using CIFAKE images, which were upscaled to 224x224 to match the resolutions of the RVAA dataset. It is important to note that upscaling the CIFAKE images lead to an exponential increase in the computational load when training the network, resulting in much higher training times. Afterwards, the newly trained model was evaluated on the RVAA dataset.

Despite these efforts, the network yielded poor results, achieving a precision of only 0.02 and a F1-score of 0.04. These findings confirm that the features learned from the CIFAKE dataset cannot be used on the RVAA dataset. In particular, the network predominantly labeled every image as "FAKE".

Models	Accuracy	Precision	Recall	F1-Score
CIFAKE_UPSCALED_step2	97.8	0.98	0.98	0.98
CIFAKE_UPSCALED_step2 on RVAA	55.4	0.02	0.53	0.04

Table 5: Results of the model trained on the upscaled CIFAKE on the RVAA dataset.