Adobe Final Report Submission

Team 86

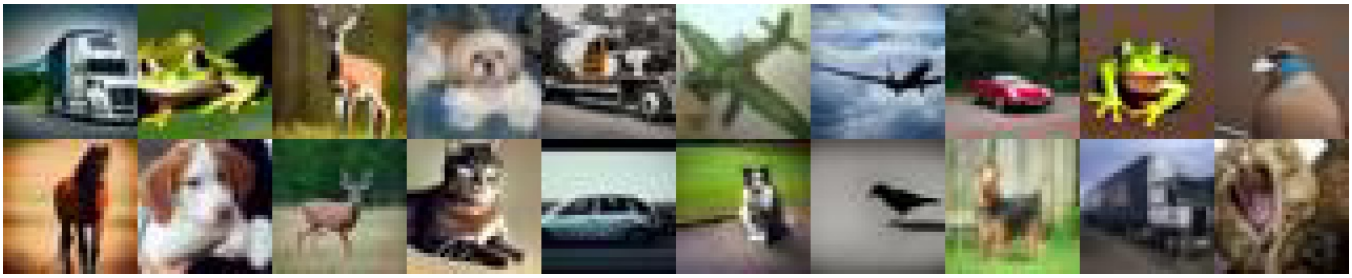

Figure 1: CIFAKE Dataset

Abstract

With the advent of generative models and their rapid improvement for tasks like image generation, detecting AI-generated images from real ones have become paramount. Most works in this domain have revolved around high-resolution images involving deep-learning models to cater to this task and lack explainability. In our approach, we have worked towards the development of a robust classifier for the detection of real vs fake images and providing comprehensive explanations for the same, which led to these conclusions.

ACM Reference Format:
Team 86. 2018. Adobe Final Report Submission. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The digital world is increasingly flooded with fake images, ranging from doctored photos created with tools like Photoshop to synthetic images generated by advanced machine learning models. With the rise of deep generative models, such as GANs [7], and diffusion models, like Adobe FireFly and Stable Diffusion [12], the ability to produce highly realistic fake images has become exciting and concerning. While these tools showcase the incredible potential for creativity and innovation, they also present serious risks, including misuse for malicious purposes. The challenge is compounded by the variety of synthetic content, from lifelike human faces to intricate, fabricated scenes. This growing complexity underscores the urgency of addressing the creation and detection of fake images to safeguard trust and integrity in the digital landscape.

The task of differentiating diffusion-generated images from real images was first introduced by Bird et al. [1] with the introduction of the CIFAKE dataset, consisting of 32x32 images based on the classes of CIFAR-10 [8] dataset using CompVis SD and an explainable model, which gives us an intuition as to why the image might be fake based on Gradient Class Activation Mapping. Since then, most works have focused on the creation of datasets of high-quality images using multiple text-to-image models [3] [19]. Efforts have also been made towards adversarial perturbations of these images to wrongly classify them [4] [6]. However, the development of explainable classifiers providing comprehensive explanations as to why an image has been determined as fake remains a task that needs further exploration. In this problem statement, we aim to develop a robust model for detecting real vs. fake images, which can look beyond adversarial perturbations and provide a comprehensive explanation.

2 Key Challenges

While the problem statement might initially appear straightforward, our experimentation with the 32×32 CIFAKE dataset revealed several critical challenges that significantly complicated the task. These challenges, spanning both technical and logistical dimensions, are outlined below:

1. Difficulty in Visual Discrimination: One of the primary obstacles was the inherent difficulty in distinguishing real and AI-generated images at such a low resolution (32×32) by the human eye. The differences between these images were imperceptible to human observers, making manual validation practically impossible. This limitation necessitated complete reliance on computational models for classification and validation, emphasizing the need for robust and accurate architectures.

2. Latency and Computational Constraints: Many state-of-the-art (SOTA) models evaluated in the study exhibited high inference times, and their parameters exceeded Adobe's expected limits. One of our primary objectives was to develop a solution that achieved high accuracy and was also computationally efficient and capable of operating on low-resource devices.

3. Lack of Adversarial Dataset Exposure: Another critical limitation was the absence of adversarially perturbed datasets in our initial evaluation pipeline. Despite achieving high accuracy on the clean CIFAKE dataset, the models exhibited poor performance when

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

exposed to adversarial attacks. This highlighted the necessity of creating synthetically perturbed datasets to assess and enhance the robustness of the models. We addressed this gap by generating adversarial datasets, as detailed in the subsequent sections of this report.

4. Artifact Identification Dataset Scarcity: The biggest challenge for Artifact Identification (in task 2) was the need for a well-defined dataset. With only a single example as a reference, it was challenging to establish a clear direction for model development. Manual labeling of images was infeasible due to the low resolution (32×32), which rendered human annotation unreliable. Additionally, even Large Language Models (LLMs) struggled to generate meaningful annotations when prompted with few-shot examples.

To address the lack of data for Artifact Identification, we employed a combination of diffusion models and advanced LLMs to generate and label synthetic datasets. This approach provided a foundational dataset for Task 2, enabling us to proceed with experimentation and evaluation. The creation of this dataset was instrumental in overcoming the bottleneck and ensuring the feasibility of Task 2.

These challenges underscored the complexity of the tasks, particularly in dealing with low-resolution datasets, ensuring adversarial robustness, and addressing the absence of task-specific datasets. Our efforts to tackle these issues, including adversarial dataset generation and synthetic dataset creation, are discussed in greater detail in subsequent sections, providing insights into our methodologies and strategies to overcome these obstacles.

3 Methodologies

3.1 Task 1: Detection of AI-Generated Images

Task 1 focuses on the binary classification of 32x32 pixel-sized images to determine whether they are AI-generated or real. The primary objective of this task was to enhance the accuracy of our models' classification and evaluate their robustness against a diverse range of adversarially attacked data.

To achieve this, we began by re-implementing two state-of-the-art (SOTA) methods from existing literature, including **DIRE**[16] and **AEROBLADE**[11].

3.1.1 ***DIRE: Diffusion Reconstruction Error.*** Diffusion Reconstruction Error (DIRE)[16] has emerged as a promising methodology for distinguishing between real and generated images by leveraging the reconstruction discrepancy observed in diffusion models. Specifically, DIRE quantifies the error between an input image and its reconstructed counterpart, asserting that diffusion-generated images are more effectively reconstructed than real images. This purported capability positions DIRE as a potential universal detector for diffusion-generated images, including those from unseen models and under various perturbations.

For the code, we have implemented the DDIM (Denoising Diffusion Implicit Models) process for image manipulation and error calculation:

- **DDIM Inversion**: Adds Gaussian noise to an image over multiple timesteps, progressively corrupting it.
- **DDIM Reconstruction**: Reverses the process by gradually removing the noise to reconstruct the image.
- **DIRE Calculation**: Measures the Diffusion Reconstruction Error (DIRE) by comparing the original image with its reconstructed version, calculating the mean absolute difference.

The code uses noise and scaling factors (alpha_t) at each timestep to control the diffusion process and compute the reconstruction error.

However, through extensive experimentation and rigorous evaluation, we observed DIRE's performance to be suboptimal, with the following metrics:

- **Accuracy**: 57.40
- **Precision**: 0.5855
- **Recall**: 0.5344
- **F1 Score**: 0.5588

While the original DIRE framework claims generalizability and robustness, our analysis revealed critical limitations. The evaluation in the DIRE framework was inadvertently biased due to a significant preprocessing discrepancy: the reconstruction errors (DIRE values) of real images were stored as lossy-compressed JPEGs, while those of generated images were saved as lossless PNGs. This inconsistency introduced substantial compression artifacts, which became a dominant factor in classification decisions. Consequently, the format-based artifacts disproportionately influenced the model's predictions rather than the inherent characteristics of real versus generated images.

Our experiments demonstrated that the DIRE-based detector's sensitivity to compression artifacts undermines its effectiveness as a robust detection methodology. The reliance on image format as an implicit feature for classification compromises the model's ability to generalize to datasets or conditions where compression artifacts do not align with those observed during training.

Given the DIRE framework's dependence on dataset-specific preprocessing, we conclude it is inadequate as a universal detector for diffusion-generated images. Future work should prioritize methods that eliminate preprocessing-induced biases and are invariant to factors such as file format and compression. Approaches that leverage intrinsic image features, independent of external artifacts, will likely yield more reliable and generalizable detection capabilities.

3.1.2 ***AEROBLADE:.*** We implemented **AEROBLADE**[11], a training-free method for detecting AI-generated images from real ones. This approach leverages a fundamental component of Latent Diffusion Models (LDMs): their reliance on autoencoders to compress and decompress images into a latent space, where the denoising process occurs. The method involves passing an image through a pair of autoencoders for compression and decompression and then calculating the perceptual loss between the original and reconstructed images. The key insight is that AI-generated images are created using an autoencoder decoder, and the perceptual loss calculated during this process tends to be lower for AI-generated images than for real images.

For our implementation, we utilized **Stability AI's sd-vae-ft-mse** as the variational autoencoder (VAE) and computed the perceptual loss using the **LPIPS** library, with **AlexNet** as the reference model.

We observed a clear distinction between the LPIPS loss of real and AI-generated images on high-resolution images, as illustrated

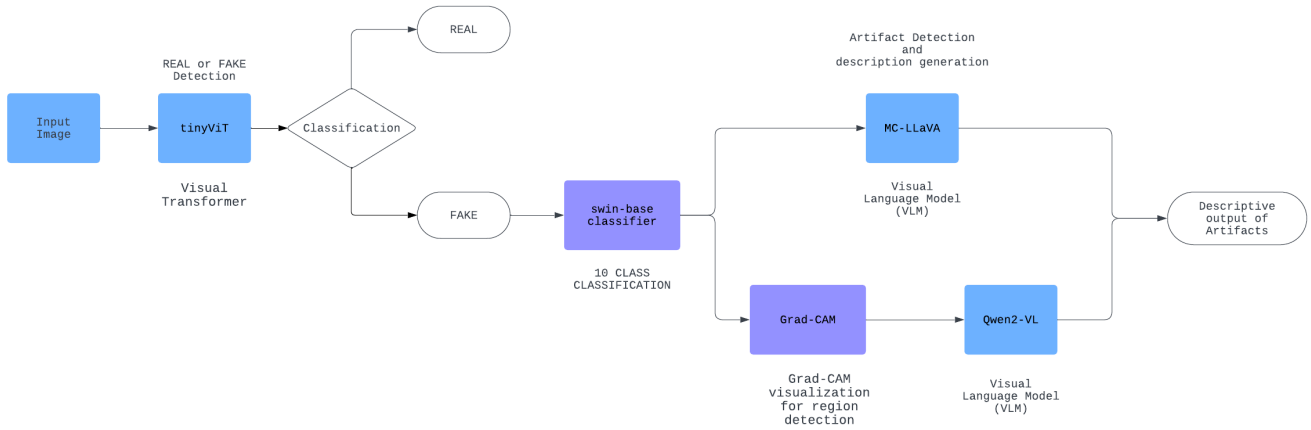

Figure 2: Flowchart showing the complete pipeline for artifact detection, classification and generation of interpretation explaination for a given image as input.

in *(results)*. However, when we applied the same method to the CIFAKE dataset, the perceptual loss values between real and AI-generated images became nearly indistinguishable, as shown in *(results)*. Moreover, the introduction of even slight perturbations or adversarial attacks severely degraded the performance of this method, further diminishing the ability to distinguish between real and AI-generated images.

While **AEROBLADE** showed promising results for high-resolution images, its efficacy on the CIFAKE dataset was significantly reduced, highlighting its vulnerability to perturbations. This limitation prompted us to explore alternative approaches to withstand smaller pixel-size images better.

3.1.3 ***RIGID:***. RIGID (Robust AI-Generated Image Detection)[2] does not require any training or prior knowledge of the generated images (e.g., which model is used for generation). The paper evaluated the detection performance of RIGID on a wide range of AI-generated image datasets and benchmarks. The results show that RIGID, albeit a training-free method, is often more effective than extensively trained classifiers.

While real and generated images often exhibit subtle differences in semantics and texture, these distinctions become increasingly difficult to discern by a human user as generation methods advance. Current training-based detectors attempt to extract these hidden differences through supervised learning. RADAR takes a different approach by exploiting the sensitivity difference between real and generated images to small perturbations. Adding noise perturbations causes the features of real images to change continuously, resulting in a smoother gradient. Conversely, generated images are more noise-sensitive, leading to a steeper change and gradient. Although the added noise is subtle, it can act as a probe for global features covering texture-rich and texture-poor regions of the image, which proves beneficial for generated image detection.

To accurately perceive how global features are affected by noise, we employ DINOv2 and ResNet as feature extractors. The input image was first resized and transformed, after which noise was added. The noise level was treated as a tunable parameter and adjusted based on performance. Cosine similarity was computed between the original and perturbed images, and a threshold was defined. Images with cosine similarity exceeding the threshold were classified as real, while those below the threshold were classified as fake. The RIGID approach demonstrated poor performance on the CIFAKE-10 dataset, with accuracy ranging between 40-50% under different threshold and noise parameter settings. DINOv2 and ResNet were employed as feature extractors, and multiple noise levels and threshold value configurations were tested. Despite these variations, the accuracy consistently failed to exceed 50%. As the similarity threshold increased, the model began classifying nearly all images as fake. This behavior highlighted the approach's limitations, ultimately necessitating exploring alternative methods.

3.1.4 ***FFT and Variational Autoencoders (VAE):***. The FFT transforms spatial-domain data into frequency-domain representations, revealing anomalies in fake images often generated by GANs. Key steps include preprocessing images to grayscale, applying FFT to extract frequency spectra, and using specific frequency bands or statistical measures for classification.

VAEs learn a lower-dimensional latent space for reconstructing input data. Trained on real images, VAEs highlight anomalies in fake images through higher reconstruction errors and deviations in latent representations. The architecture combines an encoder, a decoder, and a loss function incorporating reconstruction loss and KL divergence.

The detection pipeline integrates FFT and VAE. FFT is first applied to extract frequency-domain features from the input images, and these features are then passed through the VAE encoder to learn compact latent representations. Reconstruction errors from the decoder are analyzed to detect anomalies, which are used alongside latent features to classify images as fake or real.

The FFT+VAE approach achieved a detection accuracy of 94%. Real images, categorized as Class 0, achieved a precision of 96% and recall of 92%, while fake images, categorized as Class 1, achieved a precision of 93% and recall of 96%. The overall F1-score for both

classes was 0.94, with macro and weighted averages for precision, recall, and F1-score also at 0.94.

The FFT+VAE framework effectively identifies fake images by exploiting frequency-domain anomalies and reconstruction characteristics. Future improvements could include multi-scale frequency analysis or hybrid models incorporating spatial features.

3.1.5 ***ResNet-50:*** ResNet-50 is a deep convolutional neural network (CNN) architecture developed by Microsoft Research in 2015. It is a variant of the residual network (ResNet) architecture, where the "50" refers to the number of layers in the network. This architecture leverages residual connections, allowing gradients to flow more easily through the network by adding shortcut connections between layers. ResNet-50, with its 50 layers, has become a standard architecture for image classification tasks.

The input images were rescaled and resized for our experiments to 224×224, as this is the required input dimension for the ResNet-50 model. We employed a pre-trained ResNet-50 model, where all layers were frozen except for the final classification layer. To enhance the model's performance, several additional layers were added. Specifically, a dropout layer was introduced immediately after the last frozen layer, followed by a batch normalization layer. A flattening layer was applied to reshape the output, which was passed through a dense layer containing 64 neurons. Another dropout layer followed a second batch normalization layer, and the final dense layer with a single neuron was added. The activation function used in the first dense layer was ReLU, while the output layer used a sigmoid activation function to produce a binary classification output. The model was compiled using the Adam optimizer, with binary cross-entropy as the loss function.

The model was trained for three epochs, with performance evaluated on training and test datasets. After the first epoch, the training accuracy reached approximately 95%. By the third epoch, the **training accuracy improved to 97.35%**, while the **test accuracy reached 96.14%**. The precision and recall of the model were 97.38% and 97.32%, respectively. The model demonstrated strong performance, with no indications of overfitting during the training process.

The ResNet-50 architecture showed solid performance and was the first model to instill confidence in its ability to classify real and fake images. However, while it performed well on the initial tests, we were aware of the growing interest in transformer-based architectures, which showed promise in outperforming classical CNN models in various domains. This motivated us to explore transformer models further.

When tested on a perturbed dataset, the accuracy of ResNet-50 dropped significantly, falling to 86.13%. This performance drop was disappointing, as it highlighted the model's vulnerability to adversarial attacks. Nevertheless, this outcome catalyzed further exploration into more robust architectures, particularly those based on transformers, which could provide greater resilience to such perturbations.

3.1.6 ***CLIP+NN: Leveraging CLIP Encodings with Neural Network-Based Classification.*** In this study, we developed a classification model based on the **CLIP encoder**, a state-of-the-art architecture for learning joint visual and textual representations. CLIP (Contrastive Language-Image Pretraining) [10] is pretrained on a large corpus of image-text pairs, enabling it to align the high-dimensional image and text embeddings in a shared latent space. The architecture consists of two primary modules: (1) a vision encoder, implemented as a Vision Transformer (ViT) or Convolutional Neural Network (CNN), which extracts semantic features from images, and (2) a text encoder, typically based on transformers, which processes textual descriptions. Contrastive learning aligns these representations, facilitating robust zero-shot transfer across diverse downstream tasks.

For our implementation, we utilized the **CLIP vision encoder** as a feature extractor, leveraging its pretrained embeddings to enhance classification performance. Specifically, the vision encoder maps each input image to a 512-dimensional embedding, encapsulating its semantic content. These embeddings were subsequently passed through a lightweight neural network for classification.

Model Architecture. The proposed model follows a two-step pipeline:

- **Feature Extraction via CLIP Encoder**: Input images are processed by the pretrained CLIP vision encoder to obtain 512-dimensional embeddings, serving as compact and high-quality feature representations.
- **Neural Network Classifier**: The extracted embeddings are fed into a neural network consisting of four fully connected layers, with the following components:
  - Each layer is followed by **batch normalization** to stabilize and accelerate convergence during training.
  - **Dropout regularization** with a rate of 0.2 is applied after every alternate layer to mitigate overfitting.
  - The final layer consists of a single neuron with a **sigmoid activation function**, outputting probabilities for binary classification (real vs. fake).

The model was trained using the **Adam optimizer** with a learning rate of 1×10^{-4}, and the loss function employed was binary cross-entropy. The training process was terminated after **8 epochs**, determined via early stopping on validation accuracy to ensure convergence and generalization.

The model was evaluated on the **CIFAKE dataset**, where it achieved an **accuracy of 94.01%**, demonstrating strong discriminative capabilities in distinguishing between real and synthetic (fake) images. The results highlight the effectiveness of the pretrained CLIP embeddings, which provide robust semantic representations, enabling the lightweight neural network to achieve high classification performance with minimal computational overhead.

This architecture combines the strengths of **pretrained CLIP embeddings**, known for their versatility and generalization, with a simple yet effective neural network classifier. The approach significantly reduces the computational cost compared to end-to-end training of large models while maintaining competitive accuracy. This makes the proposed model highly suitable for resource-constrained environments and tasks requiring efficient real-vs-fake image classification.

Having achieved strong individual performances, we explored the possibility of creating a third model with comparable accuracy to facilitate an ensemble method. For this purpose, we finetuned **Google's Vision Transformer (ViT)**.

3.1.7 ***Vision Transformer (ViT):.*** The Vision Transformer (ViT) is a cutting-edge deep learning architecture that applies transformer models, originally designed for natural language processing, to image classification tasks. The specific model used in our experiments, `vit-base-patch16-224-in21k`, is pre-trained on the ImageNet-21k dataset, which consists of over 14 million images across 21,843 classes. This pre-trained ViT model operates at an input resolution of 224×224 pixels.

ViT processes images by dividing them into fixed-sized non-overlapping patches (in this case, 16×16 pixels), and each patch is linearly embedded into a fixed-dimensional vector. A classification token ([CLS]) is prepended to the sequence of patch embeddings, and absolute positional embeddings are added to retain spatial information. The sequence is then passed through multiple layers of a transformer encoder (a BERT-like architecture) to extract high-level feature representations. The [CLS] token is used for the downstream classification task.

To adapt the pre-trained ViT model to our specific task of classifying real and AI-generated images, we finetuned the model on the CIFAKE dataset. The fine-tuning was conducted for 2 epochs with the following hyperparameters:

- **Learning Rate**: 1×10^{-6}
- **Training Batch Size**: 64
- **Evaluation Batch Size**: 32

After the second epoch, the model achieved a training accuracy of **98.25%** and an F1 score of **0.982**, demonstrating a slight but significant improvement over previously explored architectures, such as ResNet-50 and CLIP-based models.

Following creating an adversarially perturbed dataset, the model's performance was evaluated under adversarial conditions. Unfortunately, the test accuracy on the perturbed dataset dropped significantly to **91.75%** This decline highlighted the model's vulnerability to adversarial attacks, indicating that the ViT architecture, while highly effective on clean data, requires further modifications or additional techniques to enhance its robustness against adversarial perturbations.

The Vision Transformer served as a strong baseline model for our task, surpassing the performance of classical convolutional architectures like ResNet-50 and modern hybrid approaches like CLIP. However, its susceptibility to adversarial perturbations underscored the need to explore more robust architectures or incorporate defense mechanisms. This observation motivated further experimentation to develop a model that could withstand adversarial attacks while maintaining high classification accuracy.

We then shifted our focus to enhancing the robustness of our model. Various attack strategies were applied, and our pre-trained models were evaluated on the resulting adversarially perturbed data. The results revealed a significant decline in performance when tested on these attacked images. Consequently, we began exploring methods to improve the overall robustness of our architecture against such attacks. This exploration led us to **Robust CLIP**.

3.1.8 ***Robust CLIP.*** In our pursuit of a robust model for adversarially perturbed datasets, we examined the work *Robust CLIP* [13]. This finetuned variant of CLIP is designed to enhance robustness against visual adversarial attacks through an unsupervised adversarial finetuning process. Given the suboptimal accuracy observed with the ViT (Vision Transformer) model on adversarial datasets, we anticipated significant improvements with Robust CLIP—a hypothesis validated through rigorous evaluation.

On the Final combined dataset, Robust CLIP demonstrated remarkable performance, achieving an **accuracy of 95.50%** and an **F1-score of 0.9549**. This surpassed the ViT (94.34% for combined dataset) model and established Robust CLIP as a highly effective solution for handling adversarial data while maintaining exceptional performance on clean, unperturbed datasets. These results underline the strength of adversarial finetuning in improving model resilience to perturbations.

To adapt this model for our specific use case, we designed a hybrid architecture that combined the robustness of the pre-trained Robust CLIP embeddings with a lightweight task-specific classifier. The architecture comprised:

- **Base Model**: The pre-trained Robust CLIP model generates 768-dimensional embeddings from input images.
- **Task-specific Classifier**: The architecture consisted of an initial linear layer reducing the embedding dimensionality to 512, followed by a GELU activation, a dropout layer (with a 50% dropout rate for regularization), and a final linear layer outputting a single probability value via a sigmoid activation.

While this architecture allowed for efficient task-specific adaptation, the computational cost of embedding generation combined with the dense layers resulted in high training and inference latency. Despite the remarkable performance, the overhead made it unsuitable for real-time or resource-constrained applications.

These limitations underscored the need for alternative solutions, prompting us to explore lightweight models with fewer parameters and reduced inference times, capable of maintaining competitive accuracy and robustness, particularly for adversarially perturbed datasets.

Before advancing further, we learned about a critical constraint: the model size for Task 1 would be evaluated against ResNet50's parameter size (23M). This led us to explore a more parameter-efficient alternative.

3.1.9 ***TinyViT [21M parameters]:*** One of the primary challenges in our experimentation thus far has been addressing the inference time and latency constraints of large-scale models such as CLIP, ViT, and Robust CLIP. While these models demonstrated high accuracy and robustness, their computational overhead rendered them suboptimal for latency-sensitive applications. We investigated the methodology proposed in the paper *TinyViT* [17] to overcome this limitation.

TinyViT introduces a novel family of lightweight and efficient vision transformers pre-trained on large-scale datasets using a fast distillation framework. Being just a 21M parameter model, the core innovation lies in transferring knowledge from large, pre-trained teacher models to smaller student models during the pretraining phase. This process employs distillation to sparsify teacher model

logits, storing them in advance to minimize memory and computational overhead. The resulting student transformers are scaled-down versions of the teacher models optimized for computational efficiency and parameter constraints.

Through a series of experiments, TinyViT proved highly effective for our task of adversarially robust image classification:

- **Initial Fine-Tuning on CIFAKE Dataset**: Fully fine-tuning TinyViT on the CIFAKE dataset yielded an accuracy of **98.67%**, showcasing its capacity to generalize well to clean datasets.
- **Evaluation on Adversarially Attacked Dataset**: When evaluated on the adversarially attacked dataset, the model achieved an accuracy of **92.87%** and a precision of **0.9953**. While the performance was promising, the results highlighted the need for further fine-tuning to enhance robustness against adversarial attacks.
- **Fine-Tuning with Adversarial Data and Generated Images**: Fine-tuning TinyViT on a combined dataset comprising adversarially attacked images and a small proportion of fake images generated by various diffusion models significantly improved its performance.
  - Accuracy increased to **98.43%** on adversarially attacked data and,
  - Further optimization yielded an accuracy of **97.39%** on all the Combined dataset.
- **Final Comprehensive Fine-Tuning**: The model was fine-tuned on a comprehensive dataset that integrated the original CIFAKE dataset, adversarially attacked data, and a diverse set of fake images generated using multiple diffusion models. This final training phase resulted in an outstanding accuracy of **98.61%**.

Additionally, TinyViT demonstrated impressive efficiency in terms of inference times. For a batch size of 32, the batch inference time was just **35ms per image**, while single-image inference was achieved in **75ms**. These results highlight its scalability for real-time applications without compromising on robustness or accuracy.

TinyViT emerged as the **optimal solution** for our Task 1: *"REAL" and "FAKE" Image Classification* due to its remarkable performance across both clean and adversarial datasets. With the added advantages of significantly reduced latency and computational overhead, TinyViT provides a **practical, scalable, and robust framework** for adversarially robust image classification tasks.

3.2 Adversarial Attacks

To evaluate the true robustness of our model, we subjected our dataset to various adversarial attacks. These attacks serve as a critical benchmark for the model, as they have demonstrated the ability to deceive even the most sophisticated classifiers. Therefore, these attacks were essential in assessing the effectiveness and resilience of our model. The attacks used to test the model's robustness are listed below:

3.2.1 ***PGD Adversarial Attack.*** The Projected Gradient Descent (PGD) [9] attack is a cornerstone method in adversarial machine learning research, widely employed to evaluate the robustness of models against adversarial perturbations. PGD is an iterative attack that constructs adversarial examples by applying small, imperceptible modifications to input data to maximize the model's prediction error. These perturbations are constrained within a predefined budget, ensuring that the adversarial examples remain visually indistinguishable from the original inputs. The PGD attack has become a standard benchmark due to its effectiveness in exposing vulnerabilities in deep learning architectures and its ability to evaluate model robustness against adversarial noise.

Our Implementation: In this study, we employed two variants of the PGD attack: one based on the L_2 norm and the other on the L_∞ norm. The configurations for these attacks were as follows:

- L_2**-based PGD Attack:** The perturbation budget (ϵ) was set to 1.0 for this variant. The ResNet50 model was utilized to generate adversarial datasets for the CIFAKE-10 dataset, ensuring diversity in adversarial examples while adhering to the L_2 constraint.
- L_∞**-based PGD Attack:** Two datasets were generated using this variant. For the ResNet50-based attack, an ϵ value of $\frac{16}{255}$ was employed. An L_∞-based adversarial dataset was also generated using a finetuned Vision Transformer (ViT) model, with $\epsilon = 0.03$ and a step size (α) of 0.01.

Objective: The primary purpose of generating these adversarial datasets was to comprehensively evaluate the robustness of our classifier under diverse attack configurations. By incorporating both L_2 and L_∞ norm-based attacks, we assessed the model's resilience to adversarial perturbations rigorously. The adversarial datasets created using these configurations provide valuable insights into the model's behavior under varying degrees of adversarial noise.

Illustration: An illustration of the L_∞ attack, as implemented using the ResNet50 model, is provided below to demonstrate the perturbation process and its effect on input data.

Figure 3: Visualization of the L_∞ attack on one of the image of CIFAKE-10 Dataset. The left panel shows the original image, while the right panel displays the adversarially perturbed image generated using the ResNet50 model. Note the subtle perturbations that remain visually imperceptible but significantly impact the model's predictions.

3.2.2 ***One-Pixel Attack.*** To rigorously evaluate the robustness of our classifier, we incorporated the **one-pixel attack**[14] into creating our adversarial dataset. The one-pixel attack is a highly targeted adversarial method designed to probe the vulnerabilities of machine learning models. Unlike conventional adversarial attacks that perturb a significant portion of the input image, the one-pixel attack modifies only a single pixel. Despite its minimal nature, this

attack can cause significant shifts in model predictions, exposing latent fragilities even in state-of-the-art deep learning architectures.

Minimal Perturbation: The attack operates with a single-pixel modification, ensuring the image remains virtually identical to its original form. This makes the attack particularly challenging to detect using conventional anomaly detection techniques.

Model Vulnerability Exploration: The profound impact of this localized perturbation highlights the model's sensitivity to pixel-level changes, providing valuable insights into the classifier's decision boundaries and reliance on specific image features.

Relevance to Real-World Scenarios: The one-pixel attack simulates highly localized and targeted adversarial scenarios, making it a valuable addition to adversarial robustness testing frameworks.

We assessed the classifier's ability to withstand extreme, localized adversarial manipulations by integrating the one-pixel attack into our adversarial dataset. The inclusion of this attack enabled us to:

- Evaluate the model's response to highly specific, minimal perturbations.
- Enhance the adversarial robustness of the classifier by addressing vulnerabilities exposed through this method.
- Broaden the scope of robustness evaluation to encompass attacks that exploit fine-grained vulnerabilities in input representations.

Incorporating the one-pixel attack was instrumental in developing a comprehensive adversarial robustness evaluation pipeline. Its simplicity yet significant impact underscores the necessity of accounting for such targeted perturbations when designing robust classifiers. This approach strengthened our model against localized adversarial attacks and contributed to its overall resilience across diverse adversarial scenarios.

Figure 4: The figure provides a visual demonstration of the one-pixel attack on one of the image of CIFAKE-10 dataset. The left panel displays the original image, while the middle panel illustrates the attacked image after applying the one-pixel modification. The right panel highlights the specific pixel that was altered to generate the adversarial example, showcasing the localized and minimal nature of this attack.

3.2.3 ***DCT Attack.*** To comprehensively assess the robustness of our models against adversarial perturbations, we conducted experiments using various attack strategies targeting both spatial and frequency domains. One notable frequency-domain attack implemented was the **Discrete Cosine Transform (DCT) Attack** [18], which manipulates the frequency representation of an image to introduce adversarial perturbations. This attack leverages the properties of DCT for effective disruption, providing valuable insights into model vulnerabilities.

The DCT attack operates using two key hyperparameters:

- **Epsilon (ϵ)**: Controls the perturbation magnitude.
- **Number of Iterations (`num_iters`)**: Specifies the number of iterations for perturbing the DCT coefficients.

The procedure for the DCT attack involves the following steps:

(1) **DCT Computation**: The Discrete Cosine Transform of the input image is computed, yielding its frequency-domain representation.
(2) **Iterative Perturbation**: Over a specified number of iterations (`num_iters`), random coefficients in the DCT representation are selected and perturbed by noise proportional to ϵ.
(3) **Constraint**: The value of `num_iters` is constrained by the total number of pixels in the image, as it dictates the maximum number of coefficients that can be modified.

Empirical evaluations revealed that setting `num_iters` to approximately $\frac{\text{total pixels}}{3}$ yielded the most effective attack results, balancing perturbation strength with computational efficiency.

To evaluate the impact of the DCT attack, a finetuned Vision Transformer (ViT) model was tested under varying levels of ϵ, analyzing its robustness across different perturbation magnitudes. The results are summarized as follows:

- **Low Noise Levels ($\epsilon = 0$ to 5)**: At minimal perturbation levels, the ViT demonstrated strong resilience, maintaining an accuracy of **96%**.
- **Moderate Noise Levels ($\epsilon = 5$ to 15)**: Increasing the noise magnitude caused a slight decline in performance, with accuracy reducing to **91%**.
- **High Noise Levels ($\epsilon > 20$)**: At significant perturbation levels, the model exhibited notable challenges. While the ViT consistently classified real images correctly, it struggled with fake image detection, achieving only **60%** accuracy on fake samples. This disparity reduced the overall accuracy to **80%**.

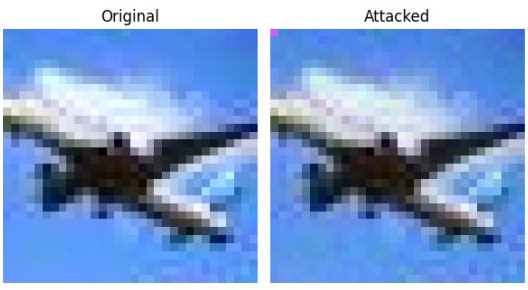

Figure 5: Visualization of the Discrete Cosine Transform (DCT) Attack

The DCT attack effectively exploits frequency-domain vulnerabilities, revealing critical insights into the robustness of deep learning models. The finetuned ViT demonstrated strong performance

under low and moderate noise levels but faced substantial challenges at higher perturbation magnitudes, particularly in distinguishing between real and fake images. These findings underscore the importance of incorporating frequency-domain perturbations in robustness evaluations and highlight the need for advanced defenses to mitigate such adversarial attacks.

3.3 Task 2: Artifact Identification and Explanation Generation

In this task, we were asked to generate interpretable explanations explaining why an image has been classified as AI-generated. In this, we have primarily faced two challenges:

- **Low resolution of the images :** We were tasked to generate an explanation for a particular image based on a low-resolution version of the original AI-generated image, where most of the intricate features that act as obvious giveaways for fake images are lost.
- **Choice of a model which can handle images of such low resolution :** Most out-of-the-box image encoding models have been trained on images of at least 129x128 dimensions and do not handle 32x32 images without resizing them to a higher dimension, which can potentially introduce even more artifacts.

To deal with these challenges, we took a "multi-model" approach, leveraging the capabilities of two strong but efficient Visual Language Models (VLMs)to aid us in this task: **1)** Qwen2-VL [15] and **2)** MC-LlaVa [1].

Why Qwen2-VL ? The choice of Qwen2-VL for this task stems from primarily two reasons :

- Its superior performance in visual language understanding is shown by its position in the VLM leaderboard.
- Its capability to handle multiple images simultaneously to generate a more nuanced output.

Why MC-LlaVa? As discussed before, most VLMs cannot handle low-resolution images out of the box, which results in many practical limitations of such a model. MC-LlaVa aims to solve one limitation, which aims to improve the model's capability to understand smaller subsections of a higher-resolution image better. To cater this model to our problem statement, we hypothesize that we can consider the 32x32 images to be a portion of higher-resolution images we have to reason. Since MC-LlaVa has been trained primarily for this task, we chose MC-Llava as one of our models.

Our approach for Task 2 can be primarily divided into Five parts:

- Swin-B class recognition
- Dataset Creation
- Obtaining Grad-CAM of the image from the model in Task 1
- Finetuning of Qwen2-VL and MC-LlaVa on the dataset
- Combining the outputs of Qwen2-VL and MC-LlaVa for a more holistic output.

Each step has been described in detail below.

3.3.1 ***Swin-B: Fine-Tuned for Artifact-Specific Classification.*** The Swin-B model, fine-tuned on the CIFAKE-10 dataset, was leveraged to perform artifact-specific classification with high precision. This task involved addressing the intricate and dynamic nature of artifacts, which were inherently dependent on the class-specific characteristics of the input images.

By utilizing its hierarchical feature extraction capabilities and shifted window-based self-attention mechanism, the Swin-B model effectively captured both global semantic information and fine-grained local patterns. This allowed it to deliver enhanced class-specific predictions, thereby augmenting the overall robustness and reliability of the classification pipeline.

The classification outputs from the Swin-B model were instrumental in enhancing prompt precision for downstream tasks, particularly in refining adversarial testing processes. The model's superior performance underscores its critical role in ensuring a robust and scalable framework for artifact detection and class-specific adversarial analysis.

3.3.2 ***Custom Dataset Creation*** . The following steps were carried out to ensure the dataset was comprehensive and of high quality:

- Fake images were generated using advanced diffusion models, including SD2,SD2.1, SD3, SD3.5 Turbo, Flux, and Pixart. These models were chosen for their ability to produce diverse and high-resolution synthetic images.
- To ensure an ample dataset for training, 500 images were generated for each class. Each model contributed 1000 images, resulting in a diverse and robust dataset for analysis and training.

Attribute Sampling and Class-Specific Considerations

- A systematic approach was adopted to sample various attributes relevant to each class. This ensured that the generated images exhibited realistic variations while maintaining the characteristics of the respective class.
- Class-specific constraints were rigorously applied to avoid unrealistic attributes. For example, attributes like *"leg"* or *"limb distortion"* were excluded from the *car* class to maintain semantic consistency and improve the model's learning.

Artifact Explanation Generation

- Pre-existing APIs were employed to generate detailed artifact explanations for the synthetic images. These APIs included:
  - GPT (various versions) provided general insights but lacked fine-grained vision capabilities.
  - Gemini 1.5 Flash demonstrated superior performance in identifying smaller and more intricate artifacts.
- While GPT struggled with tasks requiring detailed visual analysis, Gemini 1.5 Flash excelled by identifying subtle features and providing precise explanations.

Prompt for Artifact Explanation

The following prompt was carefully designed to elicit concise and accurate artifact explanations:

> *Analyze the provided image, and its corresponding Grad-CAM output which has been resized to 32x32. Focus primarily on the original image to identify and explain distinguishing artifacts that indicate it is fake. Use the Grad-CAM output for reference only when necessary.*

[1] https://huggingface.co/blog/visheratin/vlm-resolution-curse

Provide clear, concise explanations (maximum 50 words each) using the specified artifacts below. Include positional references like ́top left´ or ́bottom right´ when relevant. ***DO NOT*** *include any other sentences or artifacts in your response. Select only 6-7 relevant artifacts.* Write each artifact and explanation on a separate line, using the format: Artifact Name: Explanation. For example: Unrealistic eye reflections: Unnatural symmetrical light reflections in both eyes, suggesting generated elements. Over-smoothing of natural textures: Fur appears unusually smooth in the top right, lacking natural texture variation.
Notes: Explanations should remain under 50 words for clarity. **DO NOT** reference artifacts not listed or include extra commentary.
ONLY use the artifacts listed below:

This prompt ensured consistency across explanations and facilitated the generation of high-quality annotations. Including Grad-CAM as a supplementary reference enhances artifact detection by providing insight into areas of importance identified by the classification model.

Figure 6: A example of a AI-Generated Deer form our custom dataset

Final Dataset Preparation

- Artifact explanations for each image were generated using API calls guided by the carefully designed prompt. This process ensured uniformity and clarity in the annotations.
- The annotated dataset was curated to train the Vision-Language Model (VLM). These annotations served as a critical component for enabling the model to differentiate between real and fake images effectively and to identify specific artifacts that indicate forgery.

3.3.3 ***Generating GRAD-Cam for Fake Images.*** Grad-CAM is utilized to obtain visual explanations for the predictions made by the model from Task 1. Specifically, it helps highlight the regions of an image that the classifier focuses on when determining whether an image is fake.

These visualizations help in two significant ways:

- They enable a deeper understanding of the artifacts or patterns that contribute to classifying an image as fake, thus offering insights into the model's decision-making process.
- They serve as a valuable feature for finetuning vision-language models like Qwen2-VL. By leveraging the Grad-CAM visualizations, these models can better correlate the textual and visual cues, improving alignment and prediction accuracy.

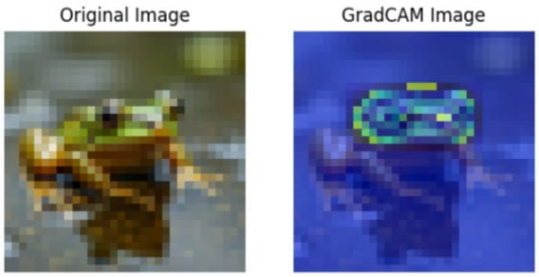

Figure 7: Grad-CAM visualization highlighting the regions in the image that contributed most to the model's prediction.

These visualizations offer supplementary insights into the artifacts or patterns influencing the model's decision.

3.3.4 Fine-tuning VLMs on the custom dataset. We leveraged the multi-image capability of Qwen2-VL and fine-tuned it to take in both the original image downscaled to 32x32 (low-resolution image) and the GRAD-Cam overlayed image and generate explanations based on the custom dataset. We decided to integrate GRAD-Cam along with the original image to get a more nuanced generation, which also takes into consideration the exact position of the potential anomaly (based on Grad-CAM) under consideration. - We parallelly fine-tuned MC-LlaVa on the custom dataset, leveraging its understanding of low-resolution images to generate explainable reasons as to why an image is potentially AI-generated. We used LoRA-based fine-tuning [5] adapters for both the models owing to its efficiency.

The hyper-parameters for model fine-tuning have been provided below:

- **LoRA Rank:** 64
- **LoRA Alpha:** 128
- **Target Module:** ["k_proj", "q_proj", "v_proj","out_proj"]
- **LoRA Dropout:** 0.1
- **Learning Rate:** 2e-5
- **Steps:** 3000

NOTE : LLaMA-Factory [2] was used to fine-tine Qwen2-VL

3.3.5 ***Combing the outputs of Qwen2-VL and MC-LlaVa.*** The outputs from Qwen2-VL and MC-LlaVa for a particular image are concatenated to form the final result. For artifacts which were generated by both models, we randomly kept the explanation generated

[2]https://github.com/hiyouga/LLaMA-Factory

by one of the models and dropped the other. The combination of outputs from both the models help us gain a holistic explanation of the reason as to why an image can potentially be AI-Generated.

4 Results

After rigorous experimentation across multiple models and approaches, we identified **TinyViT** as the optimal architecture for the detection of AI-generated images. For the artifact identification and explanation generation task, the *Multi-model arrangement of MC-LlaVa and Qwen2-VL paired with Grad-CAM* demonstrated superior performance, augmented by the integration of a specialized *Swin-B* for multiclass classification. This approach significantly enhanced the accuracy and interpretability of the second task.

The detailed methodologies and experimentation protocols have been described in the preceding sections. This section presents a comparative analysis of the performance of different models, justifying our selection of **TinyViT** based on its robustness and superior performance metrics across diverse datasets. The models were evaluated on the following datasets:

- **CIFAKE:** The original CIFAKE dataset provided by Adobe consists of a diverse set of real and AI-generated images.
- **Adversarial CIFAKE:** This dataset incorporates adversarial perturbations applied to CIFAKE images using methods such as Projected Gradient Descent (PGD) attack, Discrete Cosine Transform (DCT) attack and One-Pixel Attack.
- **Combined Dataset:** A comprehensive dataset aggregating CIFAKE, adversarially perturbed images, and synthetic images generated using advanced diffusion models, including SD2, SD3, SD3.5 Turbo, Flux, Pixart, and SDXL.

4.1 Performance Evaluation Across Models

Table 1 provides a comprehensive comparison of model performance across the three datasets. Accuracy was used as the primary metric to evaluate the detection capabilities of each model.

Table 1: Model Accuracy Across CIFAKE, Adversarial CIFAKE, and Combined Datasets (%)

Model	CIFAKE	Adversarial	Combined
DIRE	57.40	-	-
AEROBLADE	53.81	-	-
RIGID	46.34	-	-
FFT+VAE	94.12	-	-
ResNet50	96.14	86.13	89.06
CLIP + NN	94.01	86.33	89.01
ViT	98.25	91.75	94.35
RobustCLIP	-	97.23	95.50
TinyViT	**98.67**	**98.72**	**98.61**

The analysis in Table 1 underscores the superior performance of **TinyViT** across all datasets, demonstrating its ability to generalize effectively even under adversarial conditions.

4.2 Performance Metrics for TinyViT

To further analyze the robustness of **TinyViT**, we evaluated its precision, recall, and F1 score across the three datasets. Table 2 summarizes these metrics.

Additionally, TinyViT demonstrated remarkable efficiency in terms of inference times. For a batch size of 32, the batch inference time was just **35ms per image**, while single-image inference was achieved in **75ms** on **CPU**. These results underscore its capability for real-time and latency-sensitive applications without sacrificing performance or robustness.

These results highlight the robustness of **TinyViT**, particularly in handling diverse and adversarially perturbed data.

4.3 Artifact Identification and Explanation Generation

The artifact identification and explanation generation task was conducted using "multi-model" setting as described in Section 2.3, which effectively generated meaningful explanations for detected fake images.

During inference, we classify a given image using a Swin-Transformer-based multi-class classifier to find out the class to which the image belongs and ask the model to generate an explanation based on artefacts only related to the relevant class. We have compared our approach against 2 other baseline models fine-tuned on the custom dataset :

- Llama-3.2-11B-vision
- PaliGemma-3B

We illustrate the performance of each of these model of the example shown in Figure 8.

The explanations generated of the baseline model and our proposed methodology has been shown below.

Llama-3.2-11B-vision :

> *Multiple inconsistent shadow sources: Unrealistic lighting; highlights are too sharp and lack natural diffusion. Over-smoothing of natural textures: Fur lacks realistic texture and individual hair detail; appears overly smooth. Color coherence breaks: Colors are overly saturated, especially the deer's coat. Blurred boundaries in fine details: Deer is sharply in focus while background is excessively blurred.*

PaliGemma-3B :

> *Unnatural Lighting Gradients: Inconsistent lighting across the scene; shadows are unnatural and lack depth. Fake depth of field: Background elements lack natural depth of field blur, appearing blurry and artificial. Unnatural pose artifacts: The deer's position is unnatural and lacks natural balance. Over-smoothing of natural textures : Fur texture lacks realistic variation and detail; appears overly smoothed.*

Ours (Qwen2-VL + MC-Llava) with Grad-CAM

> *Over-smoothing of natural textures: There is a lack of detail in the deer's fur; it appears blurry and smudged. Unnatural colour transitions: Excessive, unrealistic colour saturation in the background.*

Table 2: TinyViT Performance Metrics Across Training and Testing Datasets

Trained on	Tested on	Accuracy	Precision	Recall	F1 Score
CIFAKE	CIFAKE	98.62%	0.984	0.9885	0.9862
CIFAKE	Adversarial CIFAKE	92.87%	0.9953	0.8615	0.9236
CIFAKE	Combined Dataset	94.70%	0.989	0.9042	0.9447
Adversarial CIFAKE	CIFAKE	97.88%	0.9646	0.994	0.9791
Adversarial CIFAKE	Adversarial CIFAKE	98.43%	0.9848	0.9839	0.9843
Adversarial CIFAKE	Combined Dataset	97.39%	0.9741	0.9737	0.9739
Combined Dataset	CIFAKE	98.67%	0.9867	0.9933	0.9868
Combined Dataset	Adversarial CIFAKE	98.72%	0.9851	0.9894	0.9872
Combined Dataset	Combined Dataset	98.61%	0.9827	0.9897	0.9861

Multiple light source conflicts: Unnatural, harsh lighting and lack of realistic shadowing on the deer.
Biological asymmetry errors: Antler's points are slightly askew, with unnatural asymmetry.
Fake depth of field: Background blur is artificially enhanced and lacks natural bokeh.
Misshapen ears or appendages: The deer's legs appear slightly distorted and out of frame.
Movie-poster like composition of ordinary scenes: The scene feels artificially composed, lacking organic placement.
Blurred boundaries in fine details: Slight pixelation visible on the edges of the image. Biological asymmetry errors: The deer's size appears disproportionate to the background trees.
Artificial depth of field in object presentation: The background trees lack detail and appear pasted on.
Impossible foreshortening in animal bodies: The deer's legs appear disproportionately short and thick.

This clearly highlights that the "multi-model" combination of Qwen2-VL and MC-LLaVa performs better, giving a more holistic explanation of why an image can be AI-generated.

Our overall pipeline had an inference time of approximately **12 seconds** per image (both models running parallel) on 80GB A100 GPU.

4.4 Conclusion

The extensive experimentation and analysis establish **TinyViT** as the most effective model for the detection of AI-generated images, achieving an accuracy of 98.61% on the combined dataset. Its precision of 0.9827 and superior performance on adversarially perturbed data further highlight its robustness and practical applicability in real-world scenarios. Additionally, the *<task2model>* demonstrated strong capabilities in artifact identification and explanation generation, providing interpretable insights into the detection process.

5 Discussion

5.1 Limitations of the Current Solution

Despite achieving notable accuracy, the current model is constrained by the need for low latency and minimal inference times, which precludes the deployment of computationally intensive state-of-the-art (SOTA) architectures. These SOTA models, if properly fine-tuned, could significantly enhance performance but remain impractical under the imposed resource constraints. Additionally, the low resolution of the CIFAKE dataset (32 × 32 pixels) presents a persistent challenge, limiting the model's ability to capture intricate generative artifacts. This constraint also complicates human verification processes, further impacting the robustness of the solution. Moreover, the absence of diverse adversarially perturbed datasets in the early development stages restricted comprehensive robustness evaluations, leaving vulnerabilities to real-world adversarial scenarios.

5.2 Observations Regarding the Data

Our experiments revealed notable trends in model performance. For Task 1, the introduction of perturbed data consistently led to a decline in classification accuracy, underscoring the vulnerability of the models to adversarial manipulations. This observation highlights the critical need for robust adversarial training methodologies to mitigate these effects.

In Task 2, the generalization of artifact detection was hindered by the scarcity of well-annotated datasets, necessitating the generation of diverse synthetic datasets to facilitate meaningful learning. These synthetic datasets were instrumental in overcoming data limitations and enabling the models to learn subtle artifact patterns effectively.

The low resolution of the input images (32 × 32 pixels) posed significant challenges, as it limited the model's ability to capture fine-grained details necessary for precise classification and artifact identification. However, techniques such as Grad-CAM provided valuable insights despite the resolution constraints. Grad-CAM visualizations revealed common patterns in the classifier's spatial focus, which were systematically leveraged to identify key artifacts indicative of generative inconsistencies. These patterns served as crucial features, allowing the models to compensate for the limitations imposed by the low-resolution input.

Figure 8: An example of an AI-generated image used for the illustration of model performance in Section 3.3.
(a) : Original Image (1024x1024)
(b) : Down-scaled Image (32x32)
(c) : Grad-CAM on the downs-caled Image

Overall, the integration of adversarially generated data and Grad-CAM-driven artifact identification played a pivotal role in stress-testing the models, exposing limitations, and enabling targeted refinements to enhance performance across both tasks. "'

5.3 Implementation Difficulties

The implementation of the solution was hindered by several technical and resource-related challenges. The adaptation of existing architectures, such as ResNet-50 and Vision Transformers (ViTs), to low-resolution inputs required extensive fine-tuning and architectural modifications. Ensuring robustness to adversarial attacks necessitated generating adversarial datasets using advanced attack strategies, such as Projected Gradient Descent (PGD) and one-pixel attacks, further complicating the pipeline. Additionally, the computational overhead of training robust architectures under resource-constrained conditions necessitated compromises in model complexity, impacting potential performance gains.

5.4 Potential Improvements

Future work can address these challenges by integrating lightweight transformer-based architectures, such as TinyViT, combined with robust adversarial training pipelines. Augmenting the dataset with multi-resolution and multi-modal inputs could enhance generalizability and robustness. Furthermore, employing knowledge distillation and multi-task learning can enable the adoption of SOTA models while maintaining operational feasibility. The application of frequency-domain feature extraction and adversarial defense mechanisms, such as input preprocessing and noise filtering, may also bolster resilience to adversarial attacks.

5.5 Broader Applications

The insights derived from this study open avenues for broader applicability. The real-vs-fake image classification models have significant implications in digital content authentication, combating misinformation, and upholding media integrity in journalism and legal domains. The explainability-driven artifact identification methods can be extended to medical imaging, where detecting anomalies with transparency is crucial. Additionally, the lightweight architectures and adversarial robustness techniques developed herein are well-suited for deployment in edge-computing environments, enabling real-time detection of fake content on social media platforms and other resource-constrained applications.

This work not only contributes to the advancement of AI-generated image detection but also establishes a scalable, explainable foundation for addressing similar challenges across a spectrum of real-world scenarios.

References

[1] Jordan J. Bird and Ahmad Lotfi. 2023. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. arXiv:2303.14126 [cs.CV] https://arxiv.org/abs/2303.14126

[2] Zhiyuan He, Pin-Yu Chen, and Tsung-Yi Ho. 2024. RIGID: A Training-free and Model-Agnostic Framework for Robust AI-Generated Image Detection. *arXiv preprint arXiv:2405.20112* (2024). https://doi.org/10.48550/arXiv.2405.20112 Accepted to CVPR 2024.

[3] Yan Hong and Jianfu Zhang. 2024. Wildfake: A large-scale challenging dataset for ai-generated images detection. *arXiv preprint arXiv:2402.11843* (2024).

[4] Yang Hou, Qing Guo, Yihao Huang, Xiaofei Xie, Lei Ma, and Jianjun Zhao. 2023. Evading DeepFake Detectors via Adversarial Statistical Consistency. arXiv:2304.11670 [cs.CV] https://arxiv.org/abs/2304.11670

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[6] Shuai Jia, Chao Ma, Taiping Yao, Bangjie Yin, Shouhong Ding, and Xiaokang Yang. 2022. Exploring Frequency Adversarial Attacks for Face Forgery Detection. arXiv:2203.15674 [cs.CV] https://arxiv.org/abs/2203.15674

[7] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. 2023. Scaling up GANs for Text-to-Image Synthesis. arXiv:2303.05511 [cs.CV] https://arxiv.org/abs/2303.05511

[8] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv preprint arXiv:1706.06083* (2017). https://doi.org/10.48550/arXiv.1706.06083

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models

From Natural Language Supervision. *arXiv preprint arXiv:2103.00020* (2021). https://doi.org/10.48550/arXiv.2103.00020

[11] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. 2024. AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error. *arXiv preprint arXiv:2401.17879* (2024). https://doi.org/10.48550/arXiv.2401.17879 Accepted to CVPR 2024.

[12] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. 2024. Fast High-Resolution Image Synthesis with Latent Adversarial Diffusion Distillation. arXiv:2403.12015 [cs.CV] https://arxiv.org/abs/2403.12015

[13] Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. 2024. Robust CLIP: Unsupervised Adversarial Fine-Tuning of Vision Embeddings for Robust Large Vision-Language Models. *arXiv preprint arXiv:2402.12336* (2024). https://doi.org/10.48550/arXiv.2402.12336

[14] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. 2017. One Pixel Attack for Fooling Deep Neural Networks. *arXiv preprint arXiv:1710.08864* (2017). https://doi.org/10.48550/arXiv.1710.08864 Published in IEEE Transactions on Evolutionary Computation, Vol.23, Issue 5, pp. 828–841, 2019.

[15] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv:2409.12191 [cs.CV] https://arxiv.org/abs/2409.12191

[16] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. 2023. DIRE for Diffusion-Generated Image Detection. *arXiv preprint arXiv:2303.09295* (2023). https://doi.org/10.48550/arXiv.2303.09295 A general diffusion-generated image detector.

[17] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 2022. TinyViT: Fast Pretraining Distillation for Small Vision Transformers. *arXiv preprint arXiv:2207.10666* (2022). https://doi.org/10.48550/arXiv.2207.10666

[18] T. Xiao, X. Deng, and W. Jiang. 2022. An invisible backdoor attack based on DCT-Injection. In *2022 IEEE International Conference on Unmanned Systems (ICUS)*. 1–6. https://doi.org/10.1109/ICUS55513.2022.9987040

[19] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. 2024. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems* 36 (2024).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009