ADOBE
Image Classification and Artifact Detection

Abstract

Deepfakes—manipulated media that closely mimic real content, pose significant threats to digital trust. Detecting these fakes becomes even more challenging when the images are very small in resolution. This report focuses on Deepfake detection and explanation generation of the detected artifacts on 32x32-sized images, altered by artifact injections from sources like Stable Diffusion, Adobe Firefly, and GANs. To address the challenge of small image size, we use a fine tuned Real-ESRGAN_x4 trained on real images from ImageNet to upscale them to 128x128, which are then fed into DenseNet121, achieving a classification accuracy of 98% and significant robustness against adversarial perturbations

To streamline the artifact detection and explanation process, the dataset is divided into two main classes: Biological and Mechanical. These classes are grouped based on shared properties for better categorization. Grad-CAM is applied to highlight artifact-affected regions, enabling more precise identification of manipulated areas. These images are then fed into a VLM to generate explanations. This combined approach enhances the performance of the Vision-Language Model (VLM) by focusing on critical features, significantly improving artifact detection accuracy.

1 INTRODUCTION

In recent years, the rapid advancement of artificial intelligence (AI) and machine learning technologies has led to the creation of highly sophisticated digital content, including deepfakes—manipulated videos, images, and audio that convincingly imitate real media. While deepfakes offer significant potential for creative and entertainment purposes, they also present profound challenges to trust and security in digital media. To combat this, researchers worldwide have contributed to creating pipelines and technologies to detect deepfakes. Deepfake detection using deep learning methods has been an area of extensive research for a long time, with the development of various models and techniques aimed at identifying subtle inconsistencies in manipulated media.

One of the key challenges in detecting deepfakes lies in the resolution of the images used for classification. For instance, 32x32 images have a low resolution, making it difficult for models to capture crucial details and context necessary for accurate classification. Super-resolution techniques offer a promising solution to this issue; however, they may introduce artifacts that distort the inherent properties of real images.

To minimize the introduction of such artifacts, it is more effective to train the super-resolution exclusively on real images. This approach enhances the models' ability to generate visually similar images that are free of any introduced artifacts, such that the classifications becomes more accurate without compromising their original characteristics.

2 DATASET

Train and Test Dataset: The CIFAKE dataset consists of two classes—Real and Fake images—each containing 50,000 images of size 32x32 for training and 10,000 images for testing. Both classes are further divided into 10 categories: Dog, Cat, Bird, Horse, Deer, Frog, Airplane, Ship, Car, and Truck. Artifacts such as unnatural lighting gradients, misshapen ears and appendages, texture bleed, and

others were prominent in the FAKE images. A total of 70 different artifacts were applied to the images.

3 LITERATURE REVIEW

The paper [4] demonstrates that a universal detector trained on a single CNN model (ProGAN) can generalize to detect images from diverse CNN architectures using robust data augmentation. It introduces the ForenSynths dataset and identifies common artifacts in CNN-generated images, enabling cross-model generalization. Despite strong results, challenges remain with heavily post-processed or out-of-distribution images.

AEROBLADE [1] is a novel, training-free method for detecting images generated by latent diffusion models (LDMs), leveraging autoencoder (AE) reconstruction errors. Unlike prior techniques, it exploits the inherent capability of AEs to reconstruct generated images with lower errors compared to real ones. Beyond detection, AEROBLADE provides qualitative insights, such as identifying inpainted regions, hence enhancing its utility. Its modularity makes it easily adaptable to new models, offering a scalable and effective solution to combat visual disinformation in the era of generative AI. Although the method achieves excellent detection accuracy, it performs best when access to the autoencoder (AE) of the generating latent diffusion model (LDM) is available, which may not always be feasible for proprietary models. Additionally, while it generalizes well across models using similar AEs, it can face reduced effectiveness for generated images with low complexity, such as logos, which are harder to detect. The method also exhibits sensitivity to image perturbations like compression or noise, though its robustness can be improved with careful tuning or alternative metrics. These challenges highlight the need for continued refinement and adaptation to address diverse real-world scenarios.

Diffusion Reconstruction Error (DIRE) [5] detects diffusion-generated images by measuring the reconstruction error between an image and its reconstruction using a pre-trained diffusion model. It leverages the fact that synthetic images are more accurately reconstructed, yielding a lower DIRE compared to real images. DIRE is computed as the L2-norm of the difference between the input image and its reconstruction. This method generalizes across diffusion models without retraining and is robust to distortions like Gaussian blur and JPEG compression. However, it may be less effective for models designed to minimize reconstruction errors or those using advanced post-processing. DIRE is also less suitable for other generative models like GANs due to differing generation processes.

The paper [3] investigates the use of Benford's Law for detecting GAN-generated images, proposing a novel approach that analyzes deviations in the statistical behavior of quantized DCT coefficients. Unlike deep learning methods requiring extensive training or handcrafted approaches with limited accuracy, this method offers a compact, architecture-agnostic feature extraction framework, achieving high detection accuracy even on low-power devices. Evaluated across diverse datasets, it outperforms state-of-the-art models, though challenges remain in handling JPEG recompression and highly realistic face images. The study highlights the potential of Benford's Law in multimedia forensics, paving the way for robust, resource-efficient GAN detection solutions.

Shadow detection and removal have progressed from traditional methods with hand-crafted features to CNN-based approaches like SP+M and DeshadowNet, which rely on paired shadow/shadow-free datasets. These models, however, struggle with limited generalization due to the restricted diversity and quality of real-world datasets. Unsupervised methods such as Mask-ShadowGAN attempt to overcome this but often yield unstable and undesirable results. SynShadow [2] addresses

these issues by synthesizing a large-scale, diverse dataset of shadow/shadow-free/matte triplets using a physically-grounded shadow illumination model. Despite its advantages, SynShadow assumes flat surfaces, external occluders, and single light sources, which limit its applicability to scenes with uneven surfaces, occluders within the camera view, or multiple light sources.

4 METHODOLOGY

4.1 Data Processing

The key challenge about detecting deepfakes in 32x32 sized images is the lack of details present in the images for models to train on. Super-Resolution is a plausible approach, and ultimately the one we have followed, but it risks the addition of artifacts that corrupt the inherent nature of real images.

4.2 Task 1

To enhance both accuracy and F1 score while maintaining robustness against adversarial attacks, we incorporated a super-resolution preprocessing step using Real-ESRGAN. This step mitigates the intensity of adversarial perturbations by enhancing image quality. The refined images are then processed using DenseNet, an architecture known for its resilience to adversarial data when appropriately trained. By pairing DenseNet with adversarially augmented training datasets, the model demonstrates exceptional performance, effectively balancing robustness and predictive accuracy.

4.3 Task 2

To detect artifacts and generate explanations, we first identify the class using DenseNet from Task 1 and leverage its Grad-CAM outputs to localize areas of interest where artifacts are likely present. By combining class information and artifact localization (e.g., distinguishing between facial or body regions), we narrow down potential artifact types, enabling more focused interactions with the vision-language model, LLaMA 3.2 Vision Instruct. Additionally, prompt templates and concise follow-up questions are employed to generate clear and contextually relevant explanations.

5 TASK 1: CLASSIFICATION

5.1 Super-Resolution

The loss of detail in 32X32 images introduced significant complexity. To address this, we used Real-ESRGAN_x4, a super-resolution architecture, to upscale the images to 128x128. This approach preserved essential details while simplifying the complexity of the classification process. The training of this model was done on real images from the ImageNet dataset. Several loss functions were taken into consideration to stabilize the training process.

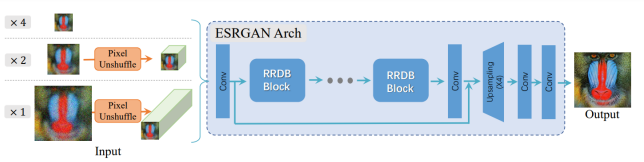

Figure 1: Real ESRGAN_X4

5.1.1 ***Perceptual Loss.*** In image processing, L1 loss is often used to measure the difference between high level semantic and textual information. It helps to generate visually realistic and similar results to the original image - The perceptual loss is defined as:

$$\mathcal{L} = \frac{1}{C_l H_l W_l} \sum i = 1^{C_l} \sum_{j=1}^{H_l} \sum_{k=1}^{W_l} \left\| \phi_l(I)ijk - \phi_l(\hat{I})ijk \right\|_2^2$$

where:

- $\phi_l(\cdot)$ represents the feature map extracted from the l-th layer of a pretrained network (e.g., VGG or ResNet),
- I is the ground truth image, and $\hat{I}$ is the predicted image,

- C_l, H_l, and W_l are the number of channels, height, and width of the feature map at layer l,
- $\|\cdot\|_2^2$ is the squared L2 norm.

5.1.2 ***Adversarial Loss***. We defined adversarial training by making modifications that allowed the discriminator to distinguish between the original high-resolution images taken from ImageNet and the super-resolved images derived from 32x32.

Loss Function:

$$\min_G \max_D \mathcal{L}\text{GAN}(D,G) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1-D(G(z)))]$$

where:

- $D(x)$: Probability assigned by the discriminator that x is a original high resolution image,
- $G(z)$: Image generated by the generator from a random noise vector z,
- p_{data}: Distribution of original high resolution image data
- p_z: Distribution of the random noise vector z.

5.1.3 ***L1 Loss***. In image processing, L1 loss is often used to measure the difference between two images on a pixel-by-pixel basis. It helps to generate high-resolution images by minimizing pixel-wise errors in low-resolution to high-resolution mapping tasks. The L1 loss is defined as:

$$\text{L1 Loss} = \frac{1}{H\times W\times C}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{C}\left|I_{ijk}-\hat{I}_{ijk}\right|$$

where:

- H is the height of the image,
- W is the width of the image,
- C is the number of color channels,
- I_{ijk} is the pixel value of the ground truth image at position (i,j) in channel k, and
- $\hat{I}_{ijk}$ is the corresponding pixel value in the predicted image.

5.2 DenseNet121

We employed a pretrained DenseNet121 model and modified the last layer by adding a fully connected (FC) layer to perform binary classification, distinguishing between real and fake images. The model was trained on the CIFAKE dataset, and then fine-tuned on adverserial attacks. The upscaled images (128x128) from ESRGAN_x4 were then fed into this model. The output from the model was passed through a sigmoid function, where the classes were labeled as 1 for fake and 0 for real. This approach resulted in a classification accuracy of 98% on the training dataset, demonstrating the effectiveness of combining super-resolution techniques with the DenseNet121 architecture.

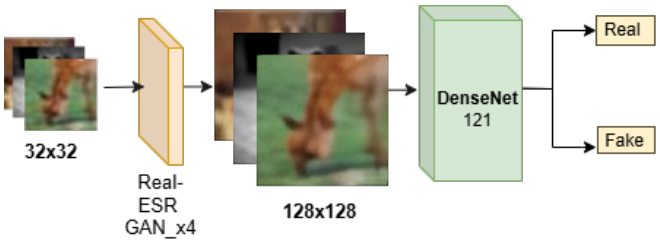

Figure 2: The Classification Pipeline

6 TASK 2: ARTIFACT DETECTION

To approach this task effectively, a comprehensive strategy was devised prior to implementation. The dataset was divided into two main classes:

- `Biological (Animal)`: Dog, Cat, Bird, Horse, Deer, and Frog
- `Mechanical`: Airplane, Ship, Car, and Truck

The Biological class was further categorized into two subgroups:

(i) Face

(ii) Body

For artifact detection in the biological class, Grad-CAM was first applied to the fake 128x128 super-resolved images to detect region of highest activation, i.e where the model thought the artifacts were. A rectangular box was drawn around the regions of highest activation. These marked images were then

fed as input into a Vision-Language Model (VLM) to classify them into either the Face or Body subcategory.

For artifact detection in the mechanical class, the

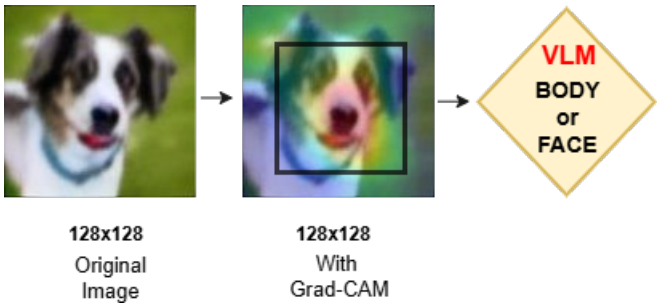

Figure 3: Focused portion of an image using Grad-CAM overlaid on the image

128x128 super-resolved image was simply passed into the VLM.

6.1 Techniques Evaluated

Due to the lack of labeled data, we initially attempted to generate images with specific artifacts using a diffusion model. However, we found that the artifacts could not be replicated in the same way as they appeared in the fake images from the CIFAKE dataset. This made the idea of using supervised training to detect artifact categories unfeasible.

We categorized the artifacts into groups such as lighting issues and biological anomalies, which simplified the task of classifying images. Following this, we evaluated the use of Vision-Language Models (VLM) for artifact detection. However, when asking the VLM to check for multiple artifacts together or to analyze entire artifact categories, it led to high inference times and often resulted in incorrect outputs.

To address this, we explored a decision tree-based approach to narrow down the possible artifacts and request explanations. We created classification models that could classify images as either mechanical or animal-related. However, this approach yielded unsatisfactory results due to adversarial attacks. We found that prompting the VLM to identify these classes was better and more accurate

Finally, we also tested the explanation generation process using VLM. This approach significantly increased the inference time and produced a higher number of incorrect results. Instead, we opted for a follow-up question strategy, which proved to be more efficient and yielded better results.

6.2 Approach Outline

To detect artifacts in super-resolved 128×128 images, a structured decision-based approach was implemented.

Instead of directly feeding the images and artifact information into a Vision-Language Model (VLM), the task of artifact detection was divided into subtasks. These subtasks were handled using either the VLM or specialized classifier models, ensuring enhanced precision and explainability. Certain artifacts were intentionally ignored due to the resizing of images to a lower resolution (32×32), which caused a loss of critical information about those artifacts.

6.2.1 ***Decision-Tree-Based Workflow.*** A decision tree framework was developed to streamline the artifact detection process:

(1) **Initial Classification** The first step involved classifying an image into one of the CIFAR-10 classes using the VLM.

(2) **Region of Interest Identification** The image was then passed through the Task 1 classifier model. Grad-CAM was applied to identify areas where the last layer of the classification model focused its attention. Bounding boxes were drawn around the regions of highest activation and passed through the VLM to determine whether the focus was on the face or the body.

(3) **Artifact Classification** Using the localized focus regions, the list of potential artifacts was narrowed down, significantly accelerating the subsequent analysis.

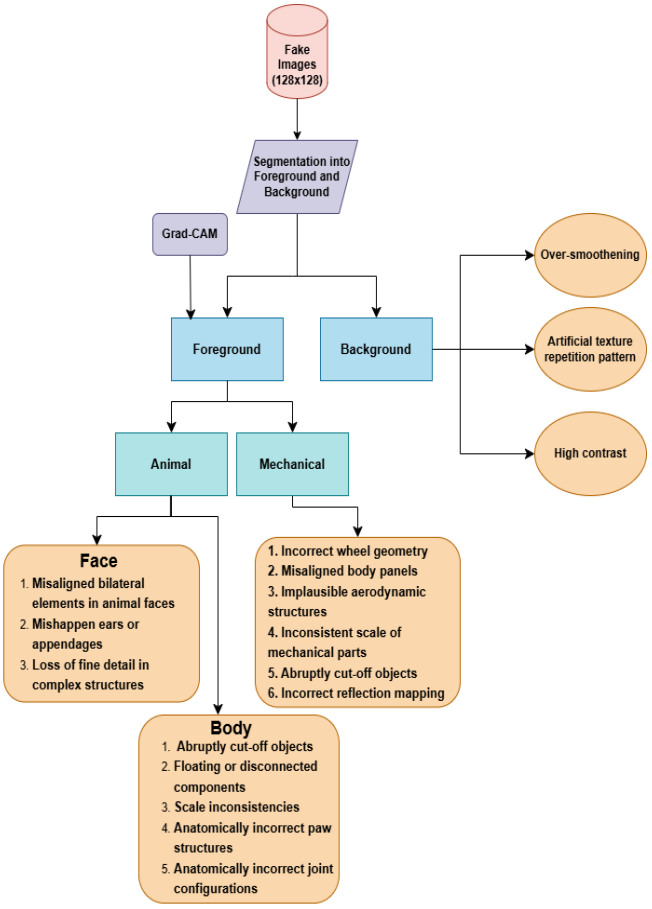

Figure 4: Rule Based Approach for VLM

(4) **Additional Artifact Classification** Artifacts such as texture repetition patterns, which the VLM struggled to detect effectively, were identified using mathematical models which will be explained in detail next. Additionally, the backgrounds of the images were extracted through a fine-tuned C-GAN trained on manually masked data. These models provided a more reliable method for detecting such specific artifacts, complementing the capabilities of the VLM.

6.2.2 ***Explanation Generation.*** To provide explanations for detected artifacts, a follow-up questioning approach was implemented:

- Predefined explanation templates were created for each artifact type.
- The VLM was presented with a set of possible causes for each artifact and tasked with selecting the most appropriate option.
- This approach enabled the model to generate coherent and contextually relevant explanations based on the predefined templates.

By combining Grad-CAM for region localization, a decision-tree methodology for classification, and predefined templates for explanation generation, this multi-stage approach significantly enhanced the efficiency and reliability of artifact detection and interpretation in super-resolved images.

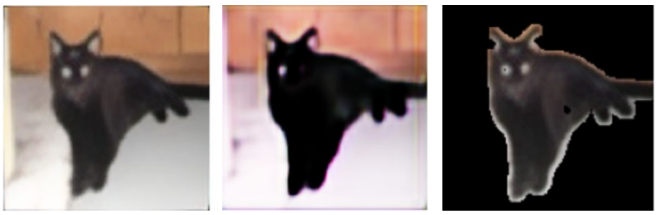

Figure 5: Showcase of masking of an image : Left to right - Original, masked foreground, masked background

6.3 Artifact analysis of Image Background

6.3.1 ***Repetitive Texture patterns.*** In our research, we explored three methodologies for detecting repetitive texture patterns in images: autocorrelation, Laws' Texture Energy, and Fourier Transform (FT). The goal was to distinguish between real and artificially generated images, particularly in datasets with synthetic textures. Among these, Fourier Transform demonstrated superior performance, as evidenced by histogram-based graphical analysis.

Fourier Transform Approach The Fourier Transform analyzes the frequency domain representation of an image, making it well-suited for detecting repetitive or grid-like patterns. The specific steps used in the FT-based analysis are as follows:

2. Preprocessing: Each image was converted to grayscale, and a 2D Fourier Transform was applied to compute the frequency domain representation.

3. Magnitude Spectrum: The transformed data was shifted to center the low-frequency components and log-transformed to enhance visibility while avoiding numerical issues.

4. Peak detection: To detect patterns, a dynamic threshold was set based on the mean (μ) and standard deviation (σ) of the magnitude spectrum:

$$\text{Threshold} = \mu_{\text{magnitude spectrum}} + 3 \cdot \sigma_{\text{magnitude spectrum}}$$

The pixels of the spectrum that exceeded this threshold were counted as *peaks*, and their total count formed the peak pixel score.

5. Classification: To classify an image as showing artificial repetitive patterns, the peak pixel score was compared against a second threshold specific to each CIFAKE-10 class:

$$\text{Peak Pixel Threshold} = \mu_{\text{real}} + x \cdot \sigma_{\text{real}}$$

The parameter x was adjusted empirically to ensure that approximately 30% of fake images in each class exceeded the threshold.

6. Statistical Analysis Distribution graphs of **peak pixel score** for all 10 classes each class containing 800 real images from ImageNet and 5000 super-resolved fake images. Below are the histograms for 4 classes. The rest of the 6 classes also show similar distribution graphs. As evident, the distribution is approximately normal, there is minimal overlap between peak pixel score of real and fake images and the threshold is set according the specific distribution of a class.

7. Threshold Optimization The threshold for each class was determined through trial and error by visually analyzing histograms of the peak pixel scores. This process ensured a clear distinction between the distributions of real and fake images, with the selected x value maximizing separation. The adaptive nature of x allowed for flexibility in handling class-specific variations in image characteristics. Finally, the average of all thresholds was taken to make inference easier while still maintaining accuracy.

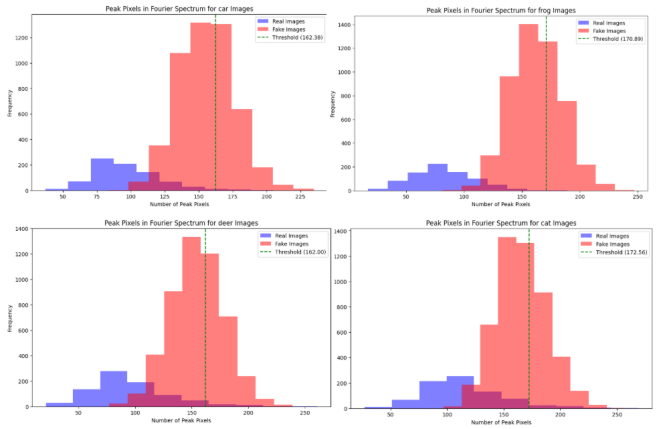

Figure 6: Histograms of peak pixel scores for 4 classes: comparison between real and super-resolved fake images.

8. Comparison with Other Methods

Autocorrelation: Autocorrelation analyzes the self-similarity of an image as it is shifted over itself. The mathematical formulation for 2D autocorrelation is:

$$R(\Delta x, \Delta y) = \sum_{x,y} I(x, y) \cdot I(x + \Delta x, y + \Delta y)$$

where $I(x, y)$ represents the intensity at pixel (x, y), and $(\Delta x, \Delta y)$ are the shifts. While autocorrelation highlights periodic structures, the resulting scores for real and fake images showed significant overlap, limiting its utility.

9. Laws' Texture Energy: This method uses convolution masks to extract textural features. For a given image $I(x, y)$, texture energy is computed using specific masks (e.g., edges, ripples):

$$E(x, y) = \sum_{i,j} |I(x + i, y + j) \cdot M(i, j)|$$

where $M(i, j)$ is the convolution mask. Despite its ability to capture local textural features, this method yielded scores with minimal separation between real and fake images, suggesting limited sensitivity to artificial patterns.

10. Results and Conclusion Fourier Transform outperformed other methods, as demonstrated by the histograms of peak pixel scores. Real images

generally exhibited lower scores due to the absence of strong periodic patterns, while fake images showed higher scores. The class-specific thresholding strategy ensured approximately 30% of fake images were correctly identified, reinforcing the method's reliability for detecting synthetic textures.

In summary, Fourier Transform, with its frequency-domain analysis and adaptive thresholding, emerged as the most effective approach for identifying artificial repetitive textures. Its results provide a robust basis for classifying real and fake images in texture-focused datasets.

6.3.2 ***Artificial smoothness.*** Oversmoothening in fake images is a common artifact that can degrade the quality of visual content by eliminating natural variations and details. Our approach is designed to focus on the background regions while ensuring that the primary object of interest is excluded from the analysis by carefully masking it.

Our approach relies on a oversmoothening-based analysis of the image background to identify oversmoothening artifacts. The methodology involves the following key steps:

1. Masking the Object of Interest The object in the image is carefully masked to exclude it from the analysis. This ensures that the focus remains entirely on the background regions, avoiding any interference from object details.

2. Artificial Smoothening Computation The method computes the oversmoothening magnitude of the background using Sobel operators, which calculate the rate of intensity change in both horizontal and vertical directions. This gradient magnitude provides a measure of texture and detail in the background.

3. Thresholding for Low oversmoothening The computed oversmoothening values are normalized to a 0–255 range, and a threshold is applied to identify regions with low oversmoothening values.

4. Background Pixel Classification The analysis is restricted to non-black background pixels, ensuring that irrelevant or empty regions are excluded. This improves the precision of the results by focusing solely on meaningful areas of the background.

5. Ratio Calculation The ratio of low-oversmoothening pixels to the total non-black background pixels is calculated. This ratio serves as a quantitative measure to determine the extent of oversmoothing.

$$R_{\text{oversmooth}} = \frac{P_{\text{low}}}{P_{\text{total}}}$$

where:

- P_{low}: Number of low-oversmoothing pixels.
- P_{total}: Total number of non-black background pixels.

6. Threshold Selection The threshold for oversmoothening magnitude was decided after an extensive review of a large dataset of background images. This dataset included various textures and levels of smoothness, enabling us to identify a threshold that balances sensitivity and specificity. As a result, the approach can reliably detect oversmoothing without falsely flagging naturally smooth regions.

7. Performance and Validation Our method has been tested on a diverse set of background images, and it has demonstrated consistent accuracy in identifying oversmoothing artifacts. The results are visually validated using masks that highlight non-black pixels and low-gradient regions. These visualizations ensure the reliability of the analysis and provide clear evidence of oversmoothing wherever detected.

Conclusion This gradient-based approach offers an effective and precise solution for detecting oversmoothing in image backgrounds. By focusing exclusively on the background and carefully masking the object of interest, we ensure that the results are both accurate and relevant. The use of a data-driven threshold further enhances the robustness

of the method, making it a reliable tool for quality assessment in image processing tasks.

6.3.3 ***High-Contrast Detection.*** High-contrast detection is another essential component of our methodology, designed to identify regions with exaggerated intensity variations. This approach operates on masked images, allowing for the analysis of both foreground and background regions. By detecting unnatural contrast, it aids in identifying artifacts or irregular enhancements in the image.

1. Preprocessing and Masking The process begins with careful masking to isolate the region of interest. Depending on the analysis requirements, either foreground or background masking is applied:

- **Foreground Masking:** The object of interest is extracted, and all surrounding regions are excluded.
- **Background Masking:** The background is retained for analysis while excluding any black cavities or other irrelevant regions using a cavity mask.

The masking ensures the analysis is focused only on relevant portions of the image, eliminating interference from extraneous regions.

2. Gradient and Intensity Analysis Once masked, the image is converted to grayscale to simplify intensity-based analysis. The detection process comprises two main steps:

(1) **Standard Deviation of Intensity:** The algorithm computes the standard deviation of pixel intensity values in the masked region. High standard deviation values suggest significant intensity variation, a hallmark of strong contrast.
(2) **Luminance Contrast:** The luminance contrast is calculated as the normalized difference between the maximum and minimum pixel intensities in the region. This metric further quantifies the contrast by evaluating brightness variation.

3. Threshold Selection and Calibration Thresholds for standard deviation and luminance contrast were determined through extensive analysis of a diverse dataset. By examining real-world and synthetic images with varying textures and contrast levels, thresholds were calibrated to balance sensitivity and specificity. This ensures reliable detection of high contrast while minimizing false positives.

4. Validation through Histograms To validate this approach, a graph plotting pixel intensity against frequency is generated for each analyzed region. These histograms provide a visual representation of intensity distribution. In cases of high contrast, the histograms exhibit instant sharp spikes across a wide range of pixel intensities, clearly indicating significant intensity variation.

The visualizations confirm the algorithm's accuracy in identifying high contrast, with the histogram spikes providing intuitive evidence of the findings.

5. Performance and Robustness The algorithm has been extensively tested on images with varied characteristics, including different masking conditions. The combination of standard deviation, luminance contrast, and histogram analysis ensures a robust and precise identification of high-contrast artifacts.

Conclusion This high-contrast detection methodology offers a reliable means to identify unnatural intensity variations in both foreground and background regions. With its flexible application, calibrated thresholds, and validation through histograms, the approach significantly enhances the artifact detection pipeline's overall effectiveness.

7 Limitations of the current solution

We are utilizing a rule-based approach to identify artifacts in images by leveraging concise, follow-up one-word responses from the VLM instead of generating full explanations. We then pre-define explanations based on these VLM responses. This

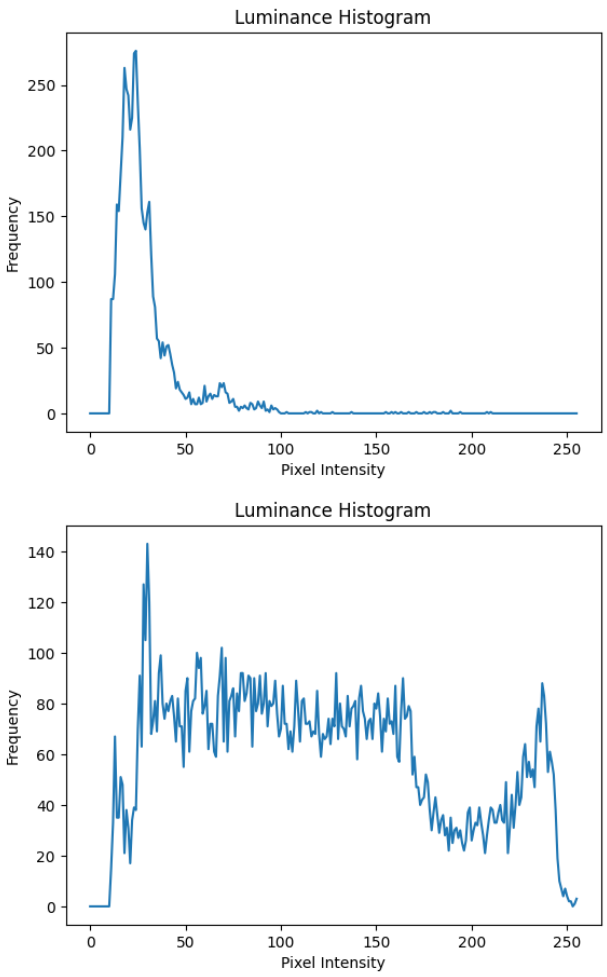

Figure 7: Histograms of pixel intensity vs. frequency for masked regions. Left: Low contrast. Right: High contrast, showing sharp spikes.

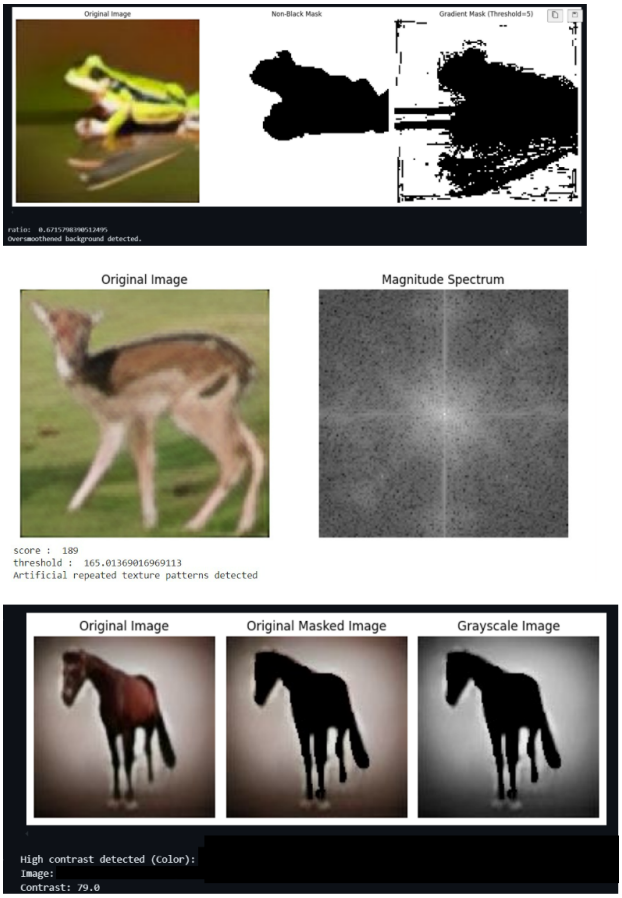

Figure 8: Over-smoothening, repetitive texture pattern and high contrast in fake images

Figure 9: Image classification and artifact detection

approach reduces the generalizability of the explanations for images. We also needed to determine thresholds for the mathematical functions used in the detection of background artifacts. However, due to the lack of labeled data, these thresholds were established through statistical analysis rather than rigorous testing. As a result, the accuracy of these methods remains uncertain.

8 Qualitative analysis of the Image background artifacts

The images above show the qualitative analysis of our approach for task 1 and task 2. These results showcase the practical usage of our solution in real world.

9 RESULTS

For `Task 1`, we achieved an accuracy of **97.9%** under the FGSM adversarial attack on the CIFAKE dataset, with an inference time of 150 ms on the

Metric	Score
Precision	0.9768
Recall	0.9779
F1-Score	0.9773
Accuracy	0.9790

Table 1: Evaluation metrics for the model.

Model	Parameters	Size
Real ESRGAN-X4	16.6 M	67.1 MB
DenseNet121	6.9 M	28.4 MB
Vision Language Model	11 B	21.3 GB
Classifier	TBD M	TBD

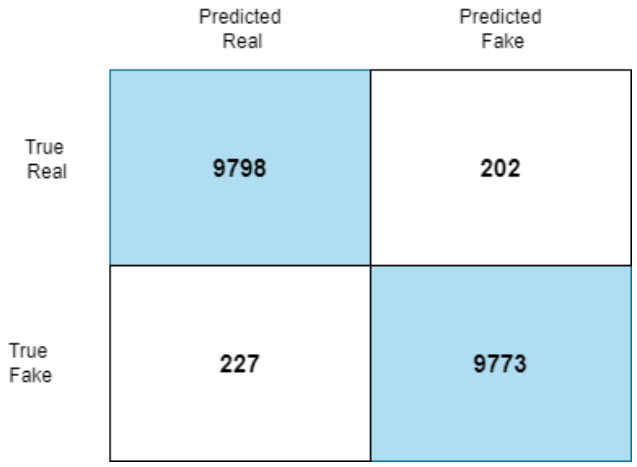

Figure 10: Confusion Matrix

CPU. In `Task 2`, results were obtained by localizing the region of interest using Grad-CAM, followed by a rule-based approach to generate explanations and detect artifacts, with further assistance from the Vision Language Model (VLM).

10 KEY CHALLENGES

One of the main challenges was the small size of the 32x32 images provided in the data set and the lack of labeled data for task 2. We generated artificial images that showcase the main artifacts of each of the 10 classes using Stable Diffusion v-4, but the generated images differed significantly from the CIFAKE-10 dataset. Artifacts prominent in the background of the images, such as over-smoothening, texture repetition patterns, and high contrast, could not be clearly detected by the Vision-Language Model (VLM), so we had to rely on a mathematical approach for these artifacts. However, this approach was challenging and research-intensive and the lack of data made it difficult to evaluate the accuracy of mathematical techniques. Additionally, designing prompts for the VLM was quite challenging as we needed to reduce inference time while still retrieving the most accurate information from the VLM. For the model used to separate the foreground and background of images, the initial results were unsatisfactory, leading us to manually generate training data by masking the super-resolved images.

11 FUTURE WORK

There are a lot of scope of improvements in this pipeline. The Super Resolution pipeline can be improved to potentially generate images that will be visually identical to the input image. This would increase the information available to the model and hence increase the prediction accuracy. The custom discriminator can also be worked on. We could train the classifier and the entire pipeline on further adversarial attacks to ensure robustness and generalization.

For Task 2 we can further improve our decision based pipeline. If given the time allowance for inference, we could have experimented further with the prompts and their effects on the final output. We could also observe the correlation between the temperature of the model, random seeds and the final inference. We could also experiment with different Vision Language Models because each has their own strengths. Overall there were not many improvements that we could have made, within the constraints that were imposed. But there is always improvement to be found.

References

[1] Denis Lukovnikov Jonas Ricker. 2024. Aeroblade: training-free detection of latent diffusion images using autoencoder reconstruction error. https://arxiv.org/abs/2401.17879.

[2] Toshihiko Yamasaki Naoto Inoue. 2021. Synthetic shadows for shadow detection and removal. Retrieved May 1, 2021 from https://arxiv.org/abs/2101.01713.

[3] Paolo Bestagini Nicolò Bonettini. 2020. Benford's law to detect gan-generated images. https://arxiv.org/pdf/2004.07682.

[4] Oliver Wang Sheng-Yu Wang. 2019. Cnn-generated images are surprisingly easy to spot... for now. https://arxiv.org/abs/1912.11035.

[5] Jianmin Bao Zhendong Wang. 2023. Dire for diffusion-generated image detection. https://arxiv.org/pdf/2303.09295.