Image Classification and Artifact Detection

Team-41

ACM Reference Format:
Team-41. . Image Classification and Artifact Detection. In ,. ACM, New York, NY, USA, 8 pages.

INTRODUCTION

In an age where AI can generate hyper-realistic images, distinguishing between what's real and what's artificially generated has become more challenging than ever. What if we could not only detect these artificial creations but also understand the reasoning behind that detection? This is the heart of our project: using the power of Faster Than Lies (FTL), and Vision-Language Models (VLMs) to not only unravel the mystery of real vs. fake images but also provide clear, human-understandable explanations behind every decision.

Our approach goes beyond just labeling 32*32 images. By using FTL, we ensure that the model's decision-making process is transparent and interpretable, shedding light on the reasons behind each classification. VLMs take this a step further, transforming technical findings into natural language, helping anyone—regardless of their AI expertise—grasp why an image is flagged as fake. With this powerful combination, our system offers more than just accuracy: it fosters trust and understanding in AI's role in discerning the authenticity of visual content in an increasingly synthetic world

KEY CHALLENGES

Developing an efficient model for classifying and explaining 32×32 resolution images presented a range of unique challenges. Most existing research focuses on high-resolution images, meaning that adapting these methodologies to such a low resolution required innovative approaches and extensive experimentation. Below, we outline the primary obstacles encountered during the development process.

Model Design for Low-Resolution Images

Classifying 32×32 pixel images was particularly challenging due to the limited amount of visual information they provided. Traditional classifiers designed for higher-resolution images, such as 256x256 and beyond, proved ineffective when applied to lower-resolution images. This necessitated the development of novel architectures and techniques. Achieving satisfactory performance required multiple rounds of experimentation and model refinement. We narrowed down to **CNNs and Vision Transformers**

Lightweight and Locally Deployable Model

Another critical challenge was designing a lightweight model that could be efficiently deployed and run on local devices, as specified in the problem statement. The model needed to meet the following requirements:

- **Parameters:** approximately 23 million
- **Inference time:** approximately 200 ms on CPUs (excluding model loading time)
- **Size:** approximately 98 MB

Achieving a balance between computational efficiency and high accuracy was paramount. In particular, it was crucial that the model maintained reliability while being deployable on devices with limited resources, such as edge devices or low-power systems. This challenge also required addressing real-time inference needs without sacrificing performance.

To meet these requirements, we explored various state-of-the-art models, including:

- **EfficientNet B0, B1, B2, B3:** These models are known for their efficiency in terms of parameters and computational requirements, making them strong candidates for low-resource environments.
- **ResNet32:** A lightweight convolutional network with fewer parameters compared to traditional deep networks, providing a good trade-off between speed and accuracy.
- **Vision Transformers (ViT):** Vision Transformers have shown excellent performance in vision tasks but typically require more computational resources. We explored various variants to determine if a lightweight version could fit our needs.
- **Top Models from Each Category:** We also investigated leading models from each domain, including lightweight convolutional neural networks (CNNs) and transformers optimized for vision tasks.

After extensive evaluation of these models, we ultimately selected **Faster than lies**, which has 29.8 million parameters and an inference time of approximately 175ms on an 8-core CPU. Despite having slightly more parameters than our initial target (23 million), it struck an optimal balance between speed and accuracy. Its performance on local devices was highly satisfactory, with inference times well below the required 200 ms, and its size (approximately 98 MB) was ideal for deployment in low-resource environments.

Explainability of Classifications

Given the presence of various artifacts in the dataset, providing explainable classifications was a major priority. Achieving this required the integration of a fine-tuned Vision-Language Model (VLM). However, due to the low resolution of the images, the performance of the VLM in generating accurate and meaningful explanations initially fell short. To overcome this limitation, extensive prompt engineering and fine-tuning with auxiliary datasets were required to improve the model's ability to explain its predictions.

Image Quality

Since the models had been downsampled to 32x32 it was very difficult for models to understand the features. So we tried upscaling and restoring the images using Non GAN and Non Diffusion techniques. We found SwinIR as one such approach that could restore the images to 128x128 with precision but this happened to add

Inter IIT Tech Meet 13.0,
.

edge based artifacts (when tested on a localisation approach we went ahead with, will be explained later in the paper). We even tried removing these artifacts but this further added complexities in more artifact identification than they had.

Figure 1: Image Quality since downsampled to 32x32

Dealing with Perturbations

Addressing the impact of perturbations on image classification was a major challenge in our study. Initially, we focused on analyzing test images to identify various types of perturbations, using methods such as frequency analysis and Local Binary Patterns (LBP). However, we soon realized that adversarial perturbations significantly impacted the model's performance, especially when these perturbations exceeded certain thresholds.

To tackle this issue, we followed a structured process:

(1) First, we analyzed the perturbations in a set of test images, evaluating their intensity based on noise, compression artifacts, and blur.
(2) Next, we extended this analysis to real-world images, as well as perturbed real images, to identify perturbations that exceeded a predefined threshold.
(3) Based on this analysis, we introduced similar perturbations to our training dataset, simulating realistic conditions and challenges.
(4) Finally, we retrained our model using this augmented dataset. By exposing the model to perturbed images, we allowed it to adapt to these variations and improve its robustness to such disturbances.

Choosing the Most Suitable Vision-Language Model

Selecting the right Vision-Language Model (VLM) for the task of low-resolution image classification and artifact explanation was a critical step in our research. It involved extensive research and experimentation with multiple models, each of which was carefully evaluated against the specific requirements of the task. These requirements included the ability to handle the unique challenges posed by 32×32 resolution images and the need to generate accurate and explainable classifications of image artifacts.

We initially explored several potential models, including PixTral, Palligemma2, Llama2, OV, LLaVA7B, LLaVA13B, and Qwen2 VL 7B. Each model was tested for its performance on low-resolution images and its ability to provide meaningful explanations for classifications. The models were also assessed for computational efficiency, as running the model locally on edge devices required a balance between accuracy and resource consumption.

Among these options, Qwen2 VL 7B stood out as the most promising candidate. After conducting a series of experiments, it became clear that Qwen2 VL 7B provided the best combination of performance, accuracy, and explainability. The model demonstrated a strong ability to handle the intricacies of low-resolution images while also providing insightful explanations for the detected artifacts. Furthermore, Qwen2 VL 7B was well-optimized for deployment in a local setting, which was essential for ensuring that the model could run efficiently on resource-constrained devices.

Model	Performance Metrics	Use Case Compatibility
PixTral (3B)	Accuracy: 79%, Speed: Fast	Low-resolution images and artifacts
Palligemma2 (4.2B)	Accuracy: 82%, Speed: Moderate	Text-heavy use cases
Llama2 OV (7B)	Accuracy: 83%, Speed: Moderate	NLP-heavy applications
Llava7B	Accuracy: 85%, Speed: Moderate	Vision-Language tasks
Llava13B	Accuracy: 87%, Speed: Slow	Vision-Language tasks, high accuracy
Qwen2 VL 7B	Accuracy: 90%, Speed: Fast	Ideal for low-resolution image classification and explanation
Qwen2 VL 7B 4-bit	Accuracy: 88%, Speed: Very Fast	Low-power devices, edge computing

Table 1: Comparison of Vision-Language Models

In conclusion, Qwen2 VL 7B was selected post testing process, where each potential model was evaluated based on its compatibility with the task's requirements. This model's ability to effectively handle low-resolution images and provide clear explanations made it the best choice for our system, and its performance was key to the success of our approach.

METHODOLOGY

The overall objective of the problem statement being to identify if an image is AI Generated that has been generated using Stable Diffusion, PixArt, GigaGAN and Adobe Firefly with adversarial perturbations introduced to prevent detection.

Dataset Collection. We were provided with an initial dataset of CiFake (1,200,000 diffusion images). To cover other types of AI-generated images as well, we looked for images generated by specific models. We included GigaGAN Conditioned, GigaGAN T2I COCO DiffNoised, PixArt, and the ArtiFact dataset. This brought the total to 1.2 million images, with nearly equal numbers of real and fake images, i.e., around 680k fake images and 520k real images.

Task-1:

Workflow.

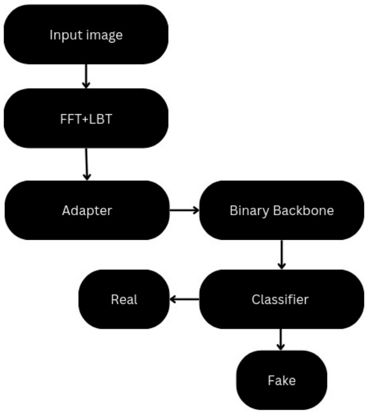


Figure 2: Workflow task 1

Model Selection. Based on 2.1 and 2.2 sections we initially began working with EfficientNet and Faster Than Lies.

Training. We had the dataset divided in 2 classes **Real and Fake**. We trained the models on CiFake to begin with. EfficientNetB1 displayed an accuracy of 94.1 %. Faster than Lies displayed an accuracy of 95.7 %. Post expansion of dataset we trained both the models parallely on following parameters:

- Initial Epochs: 20
- Train Images: 0.96 Million
- Test Images: 1.24 Million
- RandomHorizontal flip - 0.6
- RandomVertical Flip - 0.6 prob
- Random Erasing: 0.6 prob
- RandomGrayscale - 0.6
- RandomPerspective-0.6
- Random rotation - 30 degress
- Batch size 32 for Effiecient Net , 128 for Faster than Lies
- GPU used for EfficientNet: T4
- GPU used for Faster Than Lies: L4

We began with 20 epochs as the initial number which we constantly observed for overfitting. The EfficientNet model began overfitting at around 9 epochs and we did not let it go much beyond that so stopped training at 10 epochs. The Faster than lies model showed overfitting at 4th epoch so we stopped training at the end of 5th epoch.

Pertubation Identification: Since the problem statement also included adversarial perturbations introduced to prevent detection.

To tackle this issue, we followed a structured process:

(1) First, we analyzed the perturbations in a set of test images, evaluating their intensity based on noise, compression artifacts, and blur.
(2) Next, we extended this analysis to real-world images, as well as perturbed real images, to identify perturbations that exceeded a predefined threshold.
(3) Based on this analysis, we introduced similar perturbations to our training dataset, simulating realistic conditions and challenges.
(4) Finally, we retrained our model using this augmented dataset. By exposing the model to perturbed images, we allowed it to adapt to these variations and improve its robustness to such disturbances.

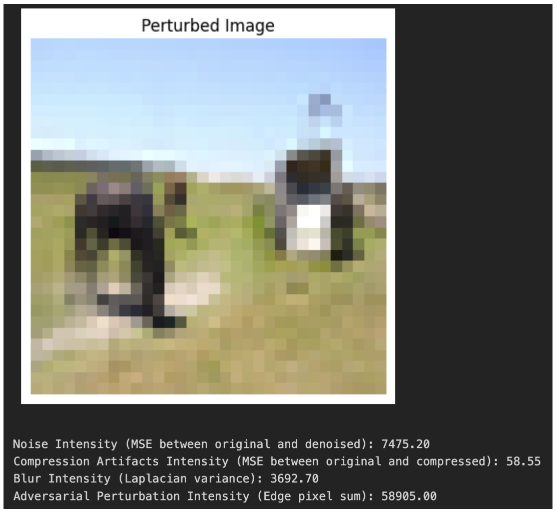
Perturbation Detection and Analysis. To accurately detect and analyze different types of perturbations, we developed a comprehensive set of methods. This included several image processing techniques designed to identify specific artifacts and distortions:

- **Noise Detection:** We used Total Variation (TV) denoising and mean squared error (MSE) to assess noise in the images.
- **Compression Artifacts:** These were simulated by resizing and upscaling images, with intensity measured through MSE.
- **Blur Detection:** We quantified blur using the Laplacian variance of the image.
- **Adversarial Perturbations:** These were detected by analyzing edge intensity, using the Canny edge detection algorithm.
- **Color Perturbations:** We analyzed shifts in the red, green, and blue channels by calculating their absolute differences.
- **Saturation and Contrast Changes:** We examined the variance in the saturation channel of the image after converting it to HSV color space.
- **Pixel Shuffling:** This was simulated by randomly permuting the pixels of the image and measuring the resulting distortion using MSE.
- **JPEG Artifacts:** These were introduced by compressing images at lower quality levels and comparing the decompressed images to the originals.
- **Resizing Artifacts:** We simulated resizing artifacts by reducing the image size and then resizing it back, measuring the distortion through MSE.
- **Edge Smoothing:** We detected smoothing by comparing the original image to one blurred using a Gaussian kernel.
- **Motion Blur:** This was simulated by applying a motion blur-like effect to the image, again quantified with MSE.
- **Pattern Injection:** We detected repeating patterns in the frequency domain using Fourier transforms.
- **Brightness Adjustments:** We measured the mean pixel intensity to identify any significant changes in brightness.

Augmenting the Dataset and Retraining

After identifying and analyzing the perturbations, we augmented the CiFake training dataset with the following synthetic pertubations: gaussian noise, salt and pepper noise, motion blur, pixelate, random hue saturation, random erase, adversarial noise(using mobilenetv2),quantization artifacts, mask based corruption. This allowed the model to see more realistic perturbations, including noise, compression artifacts, and adversarial effects, during the training process.

Once the training dataset was enriched with these perturbed images, we retrained the model. This retraining helped the model become more robust by teaching it to recognize and adapt to various types of perturbations. As a result, the model's performance improved significantly, allowing it to handle adversarial conditions more effectively.



```
Noise Intensity (MSE between original and denoised): 7475.20
Compression Artifacts Intensity (MSE between original and compressed): 58.55
Blur Intensity (Laplacian variance): 3692.70
Adversarial Perturbation Intensity (Edge pixel sum): 58905.00
```

Figure 3: Pertubation Analysis

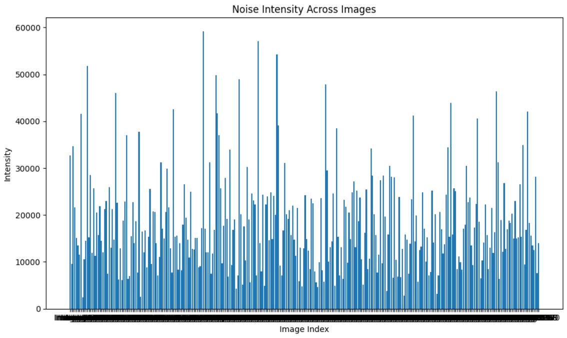


Figure 4: Pertubation Analysis

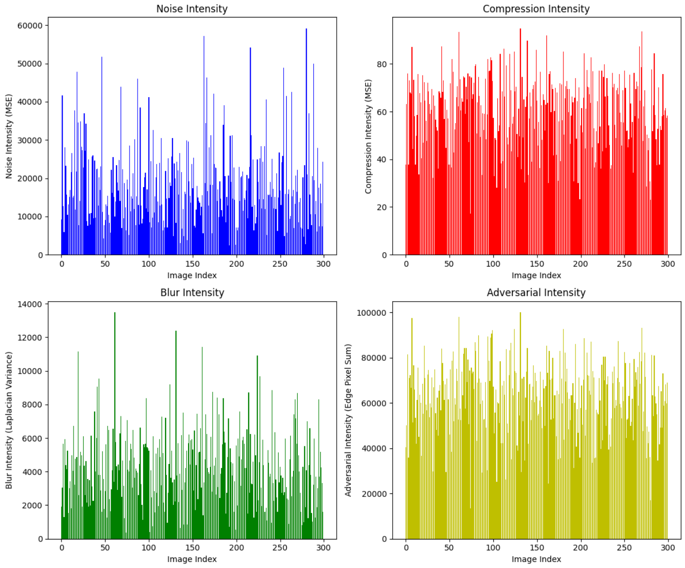


Figure 5: Pertubation Analysis

Task-2:

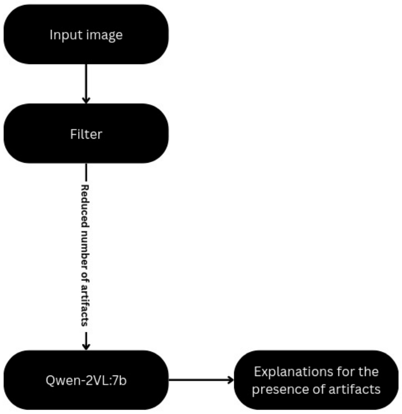

Input image
Filter
Reduced number of artifacts
Qwen-2VL:7b
Explanations for the presence of artifacts

Figure 6: Workflow for task2

Model Selection: Based on findings and experiments from Section 2.6 and Table 1, we decided to go ahead with Qwen2 VL 7B since it worked better for low resolution images and provided better explanation.

Localisation: We trained an Autoencoder on Real images so that it learns how to reconstruct real images. We then reconstructed all images in our dataset , found pixelwise loss on each image. The real part had less loss since it could be reconstructed properly whereas the parts with higher loss could not be reconstructed properly and were plotted on attention map. The areas of higher loss were highlighted and considered the localisation of the artifact regions. These localised images gave better visual understanding to the VLMs as well as the humans making it more understandable.

Figure 7

Implementation and Results. To detect and analyze the perturbations, we used a variety of image processing libraries, including OpenCV, NumPy, and SciPy. The results were visualized using

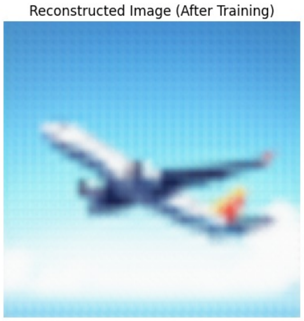


Figure 8

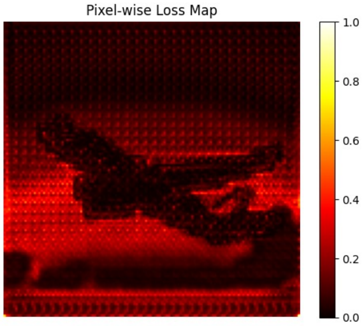


Figure 9

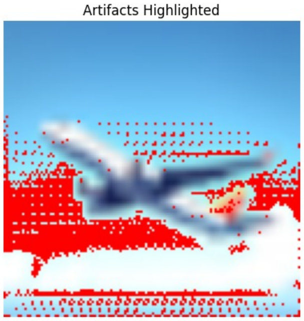


Figure 10

matplotlib, which allowed us to track and compare perturbation intensities across different images.

The retrained model showed substantial improvements, demonstrating a stronger ability to classify images even when they were distorted by various types of perturbations. illustrates how perturbation intensities varied across different test images, showcasing the model's enhanced robustness.

Artifact Filtering: Out of the total 70 artifacts , we grouped them into 8 categories namely

Artifact Groups and Descriptions.

Geometric and Structural Anomalies.

- Inconsistent object boundaries
- Discontinuous surfaces
- Non-manifold geometries in rigid structures
- Floating or disconnected components
- Asymmetric features in naturally symmetric objects
- Misaligned bilateral elements in animal faces
- Irregular proportions in mechanical components
- Impossible mechanical connections
- Inconsistent scale of mechanical parts
- Physically impossible structural elements
- Incorrect wheel geometry
- Implausible aerodynamic structures
- Misaligned body panels
- Impossible mechanical joints
- Anatomically impossible joint configurations
- Unnatural pose artifacts
- Biological asymmetry errors
- Excessive sharpness in certain image regions
- Unnaturally glossy surfaces

Texture and Surface Issues.

- Texture bleeding between adjacent regions
- Texture repetition patterns
- Over-smoothing of natural textures
- Artificial noise patterns in uniform surfaces
- Metallic surface artifacts
- Artificial enhancement artifacts
- Regular grid-like artifacts in textures
- Repeated element patterns
- Synthetic material appearance
- Artificial smoothness

Lighting and Reflection Problems.

- Unrealistic specular highlights
- Inconsistent material properties
- Multiple light source conflicts
- Missing ambient occlusion
- Incorrect reflection mapping
- Inconsistent shadow directions
- Glow or light bleed around object boundaries
- Incorrect Skin Tones
- Unnatural Lighting Gradients
- Dramatic lighting that defies natural physics
- Multiple inconsistent shadow sources

Anatomical and Biological Anomalies.

- Dental anomalies in mammals
- Anatomically incorrect paw structures
- Improper fur direction flows
- Unrealistic eye reflections
- Misshapen ears or appendages
- Anatomically impossible joint configurations
- Impossible foreshortening in animal bodies
- Exaggerated characteristic features

Perspective and Spatial Distortions.

- Incorrect perspective rendering
- Scale inconsistencies within single objects
- Spatial relationship errors

- Depth perception anomalies
- Fake depth of field
- Resolution inconsistencies within regions
- Artificial depth of field in object presentation
- Impossible mechanical joints

Image Quality Issues.

- Over-sharpening artifacts
- Aliasing along high-contrast edges
- Blurred boundaries in fine details
- Jagged edges in curved structures
- Random noise patterns in detailed areas
- Loss of fine detail in complex structures
- Systematic color distribution anomalies
- Color coherence breaks
- Unnatural color transitions
- Frequency domain signatures

Visual Artifacts from Synthetic Image Generation.

- Ghosting effects: Semi-transparent duplicates of elements
- Cinematization Effects
- Movie-poster like composition of ordinary scenes
- Unnatural pose artifacts

Occlusion and Object Cut-off Issues.

- Abruptly cut off objects
- Inconsistent object boundaries

We passed all the fake detected images through a CLIP Encoder ViT-B32 which did basic visual interpretation of the images and depicted which category can the artifacts belong to with a certain confidence score. We then shortlisted the 3 highest scoring categories and parsed it to the VLM (Qwen2 7B).

0.0.1 Processing the images. Qwen2 7B now looked for specifically these artifacts in the images. If detected , it did further visual analysis and created a textual interpretation of the artifact.

RESULTS

Inference Time

L40s was the GPU we used. The following were the inference times:

- Qwen2 VL 7B: 5.189s per image

On a 8 Core CPU:

- Image Localisation(Includes Reconstruction , Attention Mapping and Highlighting : 1s
- Faster than Lies: 175ms
- EfficientNetB0: 136 ms
- EfficientNetB0: 151 ms

Accuracy

:

- Efficient net b0: 81 Percent (two class)
- Efficient net b3 74.53 Percent (4 class)
- Faster Than Lies without pertubations: 94 Percent
- faster than Lies with pertubations: 96.5 Percent

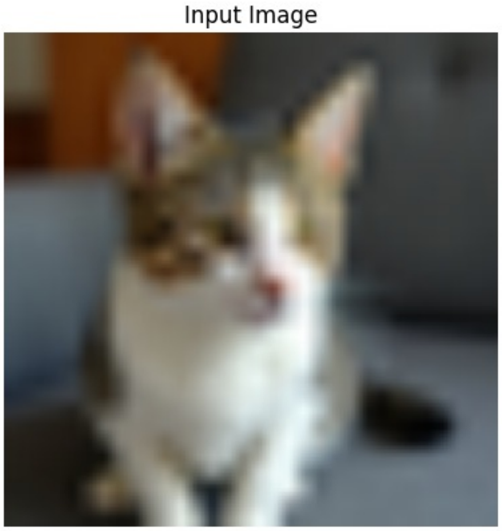


Figure 11: Input Image.

0.0.2 Qualitative Examples. **Detected: Fake** Top 3 detected artifact categories:

- Occlusion and object cutoff Issues
- Visual artifacts from synthetic image generation
- Occlusion and object cutoff Issues

Index: 18

Explanation:

- **Over-sharpening artifacts:** The image appears to have excessive sharpening, causing the edges and details to be overly pronounced.
- **Random noise patterns in detailed areas:** Noticeable noise patterns are present, particularly in the fur of the cat, disrupting the natural texture.
- **Aliasing along high-contrast edges:** There is aliasing along the edges of the cat's fur, especially around the ears and eyes, resulting in a jagged appearance.
- **Frequency domain signatures:** The image exhibits high-frequency noise patterns, which are visible in the fur texture.
- **Color coherence breaks:** Abrupt color transitions are observed, especially around the edges of the cat's fur, indicating a lack of smooth color transitions.

DISCUSSION

Limitations of current solution

- The explanations are dependednt on VLM
- Might be more AI image sources whoch we did not incorporate
- We might be missing on some pertubations
- The accuracy of the second solution's explanations cannot be tested on the current dataset provided to us.
- Our solution might not work on images smaller than this.
- Localisation has not fully been acheived , it can be implemented in a better way by even deeper analysis.

- Speed vs. Accuracy Trade-off: There is an inherent trade-off be- tween the speed of the classifier and its accuracy. Optimizing the classifier for faster performance negatively impacts its accuracy, and vice versa. This creates a limiting factor in finding the right balance between speed and accuracy.

Observations regarding the data

The test data had 185 fake images and 115 Real Images , all heavily pertubated. The results are attached in 41task1 and 41task2 json files. Common Artifacts:

- **Inconsistent shadow sources**: Shadows do not align with a single light source, a common artifact in AI-generated images.
- **Dramatic lighting**: Overly dramatic lighting that defies natural physics.
- **Incorrect reflection mapping**: Reflections do not match the lighting conditions.
- **Inconsistent shadow directions**: Shadows point in different directions, indicating inconsistent lighting.
- **Inconsistent material properties**: Mismatch in textures, e.g., rough vs. smooth fur or mixed materials (wood, metal).
- **Artificial smoothness**: Over-smoothing that removes fine details.
- **Texture repetition patterns**: Repetitive patterns in the texture due to artificial generation.
- **Incorrect perspective rendering**: Distorted proportions and unrealistic object alignment.
- **Scale inconsistencies**: Objects appear disproportionately large or small.
- **Loss of fine detail**: Compression or resolution issues causing loss of intricate details.
- **Random noise patterns**: Noise visible in detailed areas of the image.
- **Over-sharpening artifacts**: Excessive sharpening, creating exaggerated edges.
- **Artificial depth of field**: Unnatural depth of field effects.

Implementation Difficulties

Developing an efficient model for classifying and explaining 32x32 resolution images presented several challenges, as most existing research focuses on higher-resolution images. Key difficulties included:

- **Low-Resolution Image Classification:** Classifying low-resolution images was difficult due to limited visual information. Existing models designed for higher-resolution images were ineffective, requiring the development of new architectures and techniques.
- **Lightweight and Locally Deployable Model:** The model had to be computationally efficient while maintaining accuracy, making it suitable for mobile or embedded devices with limited resources.
- **Explainability of Classifications:** Low-resolution images limited the effectiveness of Vision-Language Models (VLMs) for generating explainable results. Significant effort went into prompt engineering and fine-tuning to improve explainability.
- **Handling Pertubations** Handling pertubations was a major task since they were intended to fool the classification model. A lot of research went into understanding the noises that might affect the detection process and in synthetically creating them.
- **Dataset Limitations:** Scarcity of annotated datasets for artifact detection hindered model fine-tuning. We had to curate and augment data to overcome this challenge.
- **Selection of Vision-Language Models:** Choosing the right VLM for low-resolution classification and artifact explanation involved extensive evaluation and testing.
- **Computational Constraints:** The lack of free GPU resources slowed training and fine-tuning, limiting our ability to quickly test different models or architectures.

Potential Improvements

Based on the challenges and opportunities identified during the implementation process, the following potential improvements could enhance the solution:

- **Dynamic Resolution Image Support:**
  While the model performs well for 32x32 resolution images, expanding the solution to handle higher-resolution images (e.g., 64x64 or 128x128) could improve classification accuracy. This would be particularly beneficial for cases where more detailed information is available.
- **Enhanced Model Efficiency:**
  Although the model is lightweight, exploring additional optimizations such as quantization or pruning could further reduce the model's size and inference time. These optimizations would be especially beneficial for deployment on edge devices with limited resources, enabling faster and more efficient operations on CPUs or low-power GPUs.
- **Improved Explainability:**
  Despite significant efforts in prompt engineering, enhancing the explainability of classifications could be achieved by integrating attention mechanisms or saliency maps. These techniques would allow for a better understanding of the important features used by the model in its decision-making process, making the model's predictions more interpretable, especially in real-world applications.
- **Data Augmentation and Expansion:**
  The scarcity of annotated datasets for artifact detection presents a limitation. Expanding the dataset by collecting more labeled data or leveraging semi-supervised learning techniques could substantially improve the model's robustness. Additionally, introducing synthetic data generation methods (e.g., using GANs) could augment the training dataset and improve performance on previously unseen data.
- **Model Evaluation with Diverse VLMs:**
  While Qwen2 VL 7B was selected, further experimentation with a broader range of Vision-Language Models (VLMs) could uncover models better suited for this task. Models like CLIP or VisualBERT may offer better generalization

capabilities, especially for low-resolution images, and should be evaluated for potential improvements.

- **Improved Handling of Artifact Detection:**
  Given the importance of artifact detection, integrating additional techniques like anomaly detection or contrastive learning could enhance the model's ability to distinguish between artifacts and genuine features. This would improve the model's robustness to real-world variations and perturbations in the data.
- **Real-Time Inference Improvements:**
  For applications requiring low-latency responses (e.g., edge devices or mobile applications), improving the inference pipeline for real-time applications could be beneficial. Techniques such as model distillation or hardware-specific optimizations could be explored to reduce inference time further.
- **Cross-Domain Generalization:**
  To improve robustness across various domains (e.g., different types of images, artifacts, or tasks), the model could be fine-tuned on a more diverse set of data from different sources. This would help ensure that the model generalizes well to new or unseen contexts, expanding its applicability.

These improvements would help increase the model's accuracy, efficiency, explainability, and robustness, making it more versatile and applicable to a broader range of real-world use cases.

Broader Applications

- **Medical Imaging:** Enhance diagnosis with low-resolution images from X-rays or MRIs, detecting anomalies in resource-limited environments.
- **Industrial Inspection:** Detect defects and wear in machinery or infrastructure using low-resolution inspection imagery.
- **Art Preservation:** Classify and analyze low-resolution images of historical artifacts for conservation and restoration.
- **Social Media Moderation:** Enhance content moderation by classifying low-resolution images for inappropriate content.
- **Disaster Recovery:** Aid recovery efforts by analyzing low-resolution imagery to identify damaged areas.
- **Forensic Analysis:** Support criminal investigations with the analysis of low-resolution security footage and public records.

[1–13]

REFERENCES

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609* (2023).

[2] Bin Cao, Jianhao Yuan, Yexin Liu, Jian Li, Shuyang Sun, Jing Liu, and Bo Zhao. 2024. SynArtifact: Classifying and Alleviating Artifacts in Synthetic Images via Vision-Language Model. *arXiv preprint arXiv:2402.18068* (2024).

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929* (2021). https://arxiv.org/abs/2010.11929

[4] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. 2018. Countering Adversarial Images using Input Transformations. *arXiv preprint* abs/1711.00117 (2018). https://arxiv.org/abs/1711.00117

[5] Romeo Lanzino, Federico Fontana, Anxhelo Diko, Marco Raoul Marini, and Luigi Cinque. 2024. Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 3771–3780.

[6] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved Baselines with Visual Instruction Tuning.

[7] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[8] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning.

[9] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. 2024. AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error. *Ruhr University Bochum* (2024).

[10] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR. https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

[11] 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).

[12] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2024. LLaVA-CoT: Let Vision Language Models Reason Step-by-Step. arXiv:2411.10440 [cs.CV] https://arxiv.org/abs/2411.10440

[13] Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. 2024. FakeShield: Explainable Image Forgery Detection and Localization via Multi-modal Large Language Models. (2024).