AI-Generated Image Detection Using Simplified Hybrid Feature Fusion

Nashrah Haque
nhaque14@fordham.edu

Lavanya Pushparaj
lp3@fordham.edu

Thien An Pham
tpham42@fordham.edu

Abstract

AI-generated images (AIGI) from models like Stable Diffusion present challenges for distinguishing synthetic content from real images, crucial for media verification and misinformation control. This study proposes a novel hybrid detection approach that fuses low-level frequency features from the Fourier Transform with high-level semantic embeddings from the CLIP model. Using ImageNet-100 for real images and synthetic images generated by Stable Diffusion, three classifiers—XGBoost, Neural Networks, and SVM—were trained. XGBoost achieved the highest accuracy of 99.3%, demonstrating the effectiveness of the hybrid feature approach. The results highlight the potential of combining frequency and semantic features for robust AIGI detection.

1 Introduction

Artificial Intelligence-Generated Images (AIGI) have become increasingly prevalent due to advancements in generative models such as Stable Diffusion [11], DALL-E [10], and GANs [8]. These models can create images that are virtually indistinguishable from real photographs, presenting significant challenges in verifying the authenticity of media. The potential misuse of such synthetic content amplifies concerns in combating misinformation, detecting deepfakes [13], and maintaining the integrity of visual content in journalism, law, and public discourse.

Existing methods for detecting AIGI primarily rely on either low-level pixel-based analysis [15] or high-level semantic evaluations [9]. However, the rapid improvement of generative models has rendered many traditional approaches less effective, as these models increasingly replicate the statistical properties of real-world images. Addressing these limitations, hybrid detection methods have emerged as a promising alternative by integrating features across multiple domains [5, 6].

This study introduces a novel hybrid detection framework that combines low-level frequency features derived from the Fourier Transform [5] with high-level semantic embeddings extracted using the Contrastive Language-Image Pretraining (CLIP) model [7]. Frequency-based features capture statistical anomalies that are characteristic of synthetic image generation processes, while semantic embeddings provide contextual insights that enhance detection robustness [16]. The proposed method leverages the strengths of both feature spaces to improve classification accuracy.

To evaluate this approach, a dataset comprising real images from ImageNet-100 [4] and synthetic images generated by Stable Diffusion [11] was used to train and test three classifiers: XGBoost, Neural Networks, and Support Vector Machines (SVM). Among these, XGBoost achieved the highest accuracy of 99.3%, demonstrating the effectiveness of the hybrid detection method in distinguishing synthetic images from authentic ones.

This hybrid methodology aligns with prior research highlighting the benefits of frequency domain analysis [5, 6] and semantic embedding models [7] in detecting synthetic media. By achieving state-of-the-art detection performance, this study contributes to the growing body of knowledge on safeguarding digital media authenticity. Future research will focus on optimizing computational

efficiency and adapting the framework for real-time applications to extend its applicability to emerging generative models and other forms of synthetic content, including text and audio.

By addressing both the statistical and semantic aspects of image synthesis, this study provides a robust foundation for combating the challenges posed by AIGI. It sets the stage for further advancements in hybrid detection systems, paving the way for practical implementations in real-world scenarios.

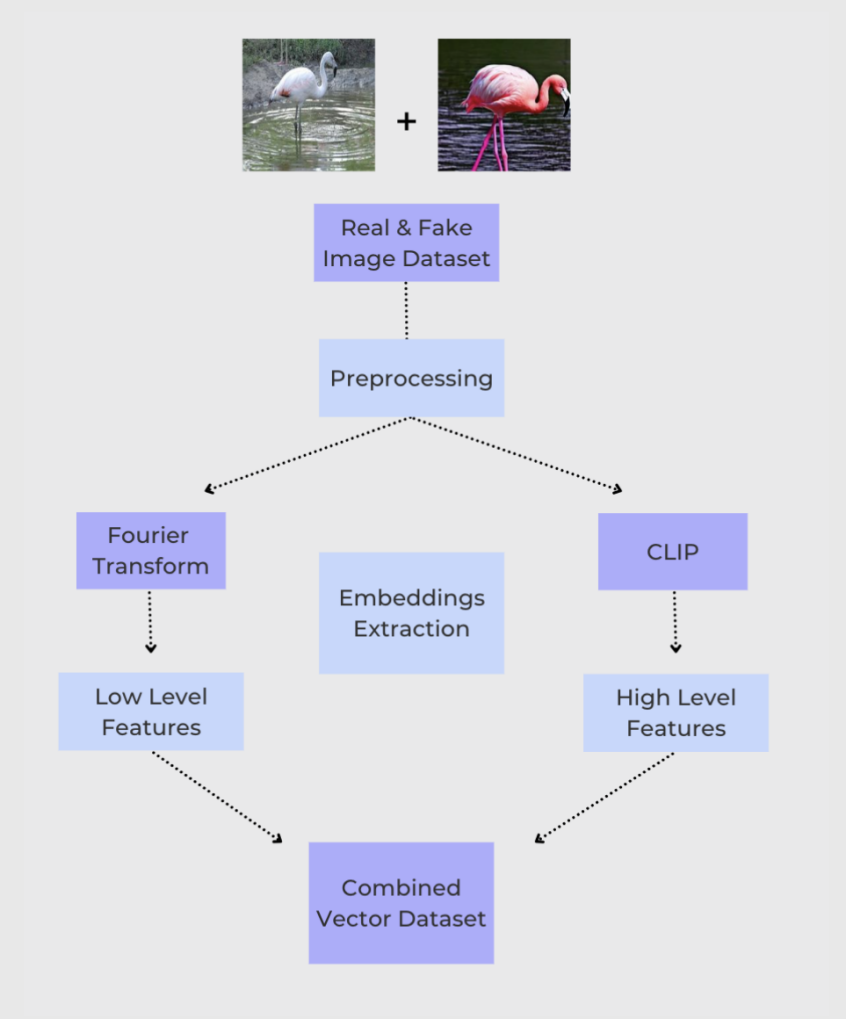

Figure 1: Flowchart illustrating the embedding extraction process. The diagram outlines the steps involved in low-level feature extraction using the Fourier Transform, high-level semantic feature extraction using CLIP, and the subsequent integration and labeling for training and testing.

2 Literature Review

Existing techniques for detecting AI-generated images can be broadly categorized into frequency domain analysis, semantic embedding models, and reverse diffusion-based approaches. Each method has shown promise but also suffers from limitations that restrict its adaptability and effectiveness.

Frequency domain analysis employs mathematical transformations, such as the Fourier Transform, to examine images in the frequency space rather than the pixel space. This technique is adept at revealing pixel-level artifacts like repetitive textures or unnatural edge patterns often found in lower-quality synthetic images. Bayar and Stamm [3] demonstrated the effectiveness of frequency analysis in detecting tampered images by highlighting inconsistencies in spectral data. Their study specifically used constrained convolutional networks to identify subtle manipulations, emphasizing the potential

of spectral patterns in exposing forgery. However, this method struggles to identify subtleties in higher-quality images generated by diffusion models, where pixel-level artifacts are minimized [5].

Semantic embedding models, such as the CLIP model developed by OpenAI, leverage a contrastive learning framework to link visual and textual representations. By analyzing high-level semantic relationships in images, CLIP embeddings can uncover conceptual inconsistencies that suggest synthetic origins. Radford et al. [7] used CLIP embeddings to evaluate semantic coherence in visual-text pairs, showing that these embeddings can distinguish between real and synthetic images by identifying mismatches in high-level semantics. Their work laid a foundation for detecting nuanced generative inconsistencies. While effective in capturing overarching image content, these embeddings are less capable of detecting fine-grained pixel-level discrepancies, particularly in sophisticated AI-generated content.

Reverse diffusion-based methods, such as Diffusion Reconstruction Error (DIRE), exploit the reverse generative process to detect synthetic images. This approach involves reconstructing input images using a diffusion model and analyzing the resulting reconstruction errors. Saharia et al. [12] applied reverse diffusion to reconstruct generative artifacts, highlighting artifacts left during the diffusion process. Their study revealed high detection accuracy but also underscored the resource demands of this approach. Although effective, reverse diffusion techniques are computationally expensive, requiring high-performance hardware, which limits their scalability for real-world applications [11].

Hybrid approaches combine the strengths of both frequency domain features and semantic embeddings to create robust detection systems. Yan et al. [14] introduced AIDE, a hybrid model that utilizes CLIP embeddings for high-level semantic analysis alongside frequency-based features to detect artifacts such as noise patterns and anti-aliasing. This framework demonstrated substantial improvements over state-of-the-art methods across multiple benchmarks.

Similarly, Rahman and Emon [1] proposed a hybrid detection framework that integrates Kolmogorov-Arnold Networks (KAN) with Multilayer Perceptrons (MLP). Their model excels in capturing complex patterns within AI-generated images and outperformed traditional approaches in robustness and generalization during out-of-distribution testing.

The primary limitation of existing approaches lies in their singular focus on either pixel-level artifacts or high-level semantics, which restricts their adaptability and effectiveness. Hybrid approaches address this gap by integrating diverse feature spaces to enhance detection accuracy and robustness.

3 Methodologies

Our approach to detecting AI-generated images involved a systematic pipeline, integrating real images from ImageNet-100 with synthetic images generated using Stable Diffusion, followed by feature extraction, model training, and testing. Below, we outline the detailed process.

3.1 Dataset Preparation

We utilized ImageNet-100 [4] as the source for real images, encompassing a diverse range of natural image categories. For synthetic images, we employed Stable Diffusion [11], a state-of-the-art latent diffusion model.Our main goal was that the images remain true to the class label. For both classes, real and fake we generated 1000 images each.

The generated images were labeled as "synthetic," while real images were labeled as "authentic." This labeled dataset served as the foundation for feature extraction and subsequent training.

3.2 Stable Diffusion Image Generation

To generate synthetic images for our dataset, we utilized Stable Diffusion, a state-of-the-art latent diffusion model [11]. The image generation process was implemented in Python, leveraging the Diffusers library and associated tools for efficient image synthesis. In the following, we describe the methodology.

3.3 Loading Stable Diffusion

The Stable Diffusion pipeline was loaded using the Diffusers library. This included the pretrained model weights from the "bguisard/stable-diffusion-nano-2-1" repository. The pipeline was initialized with a DDIMScheduler to control the diffusion process and optimized for memory efficiency through xFormers memory-efficient attention and VAE slicing.

3.4 Prompt Preparation and Embedding Generation

Each synthetic image was generated using a text prompt corresponding to the class labels in the dataset. Prompts were tokenized and passed through Stable Diffusion's text encoder to produce text embeddings. Unconditional embeddings were also generated for guiding the model during inference. A custom embedding class was employed to manage token-specific optimization when required.

3.5 Diffusion Process

The forward diffusion process applied latents initialized from a random Gaussian distribution, scaled according to the noise sigma defined by the scheduler. At each timestep, the latents were processed by the model's UNet to predict noise components, which were adjusted using guidance scaling. The adjusted latents were decoded by the variational autoencoder (VAE) into images. Postprocessing ensured the output images were normalized to the correct pixel range.

3.6 Class-Specific Image Sampling

To simulate class-specific variations, prompts were derived from ImageNet-100 class labels [4]. For each class, a fixed number of samples were generated, ensuring a balanced representation in the dataset. Images were saved in a structured directory for downstream tasks, facilitating clear labeling of synthetic content.

3.7 Framework Efficiency

The generation framework employed optimizations to streamline image synthesis. These included:

- **Memory Management:** xFormers attention and VAE slicing reduced GPU memory usage, enabling efficient large-scale generation.
- **Logging and Metrics:** Intermediate metrics, such as loss and confidence scores, were logged during generation, providing transparency and traceability.

This methodology ensured the creation of high-quality synthetic images that were critical for training and evaluating our hybrid detection models.

3.8 Low-Level Feature Extraction Using Fourier Transform

Low-level frequency features were extracted using the Fourier Transform, implemented via the OpenCV and SciPy libraries [5]. Each image was transformed into its frequency representation, revealing subtle patterns such as repetitive textures or unnatural edge artifacts. These frequency-domain features were saved as numerical vectors, capturing the spectral signatures indicative of synthetic content [2].

3.9 High-Level Feature Extraction Using CLIP

High-level semantic features were extracted using the CLIP model [7], which leverages a contrastive learning framework to encode visual and textual representations. Each image was passed through the pre-trained CLIP model to generate semantic embeddings, which capture the contextual and conceptual relationships within the image. These embeddings provide an additional layer of information that complements the frequency-domain features.

The embedding extraction process, depicted in Figure 1, outlines the pipeline for generating and combining features from both low-level frequency domains and high-level semantic embeddings.

The features are concatenated to form unified representations, which are subsequently used to train and evaluate the classifiers. The flowchart visually emphasizes the integration of complementary feature spaces, demonstrating the systematic approach to achieving robust classification performance.

3.10 Feature Integration and Labeling

The low-level frequency features and high-level semantic embeddings were concatenated to form a unified feature vector for each image. This integration aimed to leverage the complementary strengths of both feature domains, enabling a robust representation for classification. The concatenated vectors were paired with their respective labels ("synthetic" or "authentic") for training.

3.11 Model Training

We trained three different classifiers on the integrated feature set:

Support Vector Machines (SVM): A linear SVM was used to establish a baseline for the classification task.

Neural Networks: A custom feedforward neural network with multiple hidden layers was trained to explore the nonlinear relationships within the integrated feature space.

XGBoost: A gradient boosting model was employed for its robustness and ability to handle feature interactions effectively.

The models were trained using an 80-20 train-test split, with hyperparameter tuning performed via Optuna for optimal performance.

3.12 Testing and Evaluation

The trained models were evaluated on the test set to measure their accuracy, precision, recall, and F1 scores. Among the classifiers, XGBoost achieved the highest accuracy of 99.3%, demonstrating the effectiveness of combining frequency and semantic features. The results validated the robustness of the hybrid feature approach in distinguishing synthetic content from real images.

The next subsections will elaborate on each of the methodologies, and the models we used.

3.13 Fourier Transform

The Fourier Transform is a mathematical tool that analyzes the frequency domain of images, providing insights that are often hidden in the pixel domain. By decomposing an image into sine and cosine components, this technique reveals repetitive textures, edge artifacts, and other spectral patterns that are characteristic of synthetic imagery [5, 2]. These frequency-based features are particularly effective in detecting pixel-level anomalies introduced during the generative process of AI models, such as up-sampling or convolutional artifacts [5].

In this study, the Fourier Transform was implemented using the OpenCV and SciPy libraries, which facilitated efficient computation of frequency representations. These tools were employed to extract low-level pixel patterns, focusing on irregularities that are subtle but indicative of synthetic origins. The resulting transformation output was stored as frequency coefficients, enabling a seamless integration with high-level semantic embeddings in the classification stage [14].

The integration of frequency domain features into hybrid models, as demonstrated in this project, leverages the unique strengths of the Fourier Transform to identify artifacts that might be overlooked by spatial-domain or semantic-based approaches. This capability makes the Fourier Transform a cornerstone of this hybrid detection framework, effectively enhancing the robustness and accuracy of AI-generated image classification.

Figure 2: Spectral magnitude analysis of two images. The first column shows the original images, the second column displays their grayscale conversions, and the third column illustrates the corresponding magnitude spectra in the frequency domain. The spectral magnitude highlights the amplitude of frequency components, revealing patterns and textures not easily visible in the spatial domain.

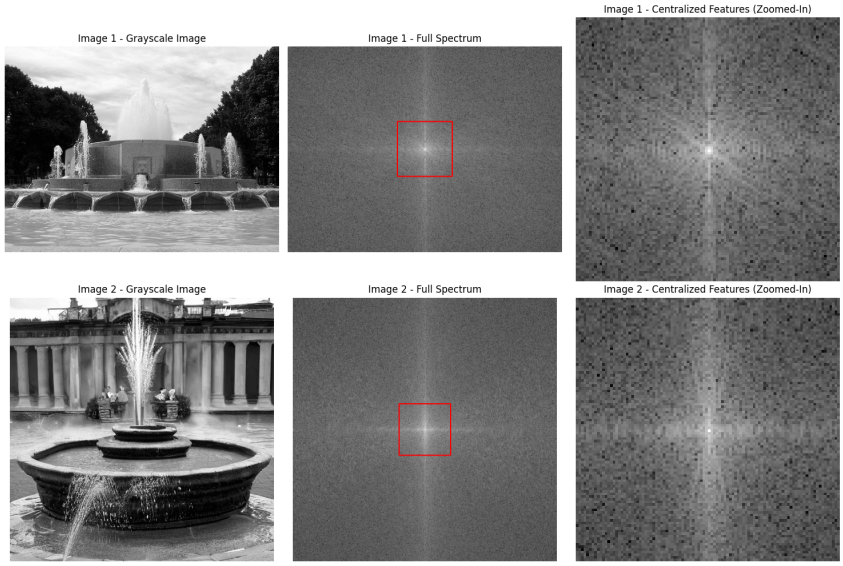

Figure 3: Frequency domain analysis of the centralized features. The image highlights the concentration of low-frequency components at the center of the spectrum, revealing dominant patterns and structures present in the data.

3.14 Clip Embeddings Extraction

To capture high-level semantic features, we employed the Contrastive Language-Image Pretraining (CLIP) model developed by OpenAI. The CLIP model maps images into a high-dimensional semantic space, allowing for the detection of conceptual inconsistencies that may be indicative of synthetic origins. Using Hugging Face's Transformers library, we extracted semantic embeddings for each image. These embeddings are powerful tools for identifying discrepancies in the semantic coherence of images, especially when dealing with AI-generated content. This process involved passing each image through the pre-trained CLIP model, generating embeddings that encapsulate the image's

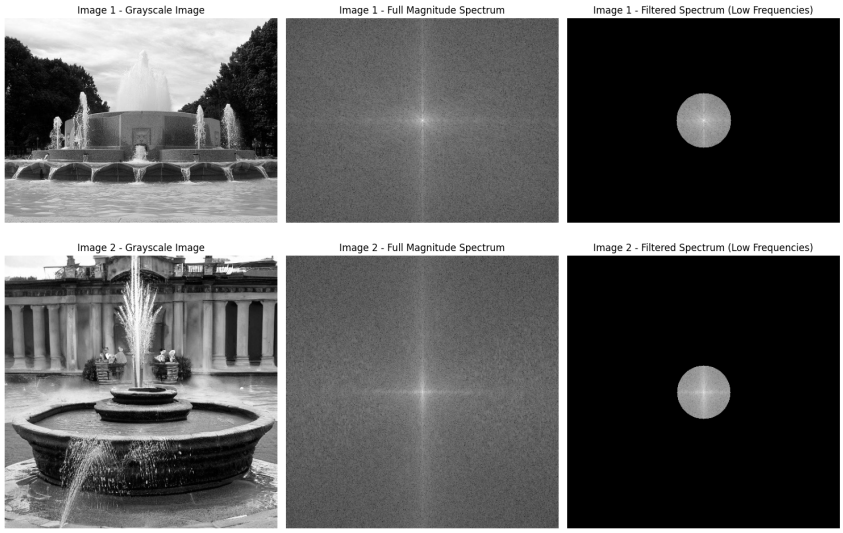

Figure 4: Frequency domain analysis after filtering. The image shows the effect of isolating low-frequency components, where high-frequency details are suppressed, emphasizing smoother and more prominent patterns in the spectrum.

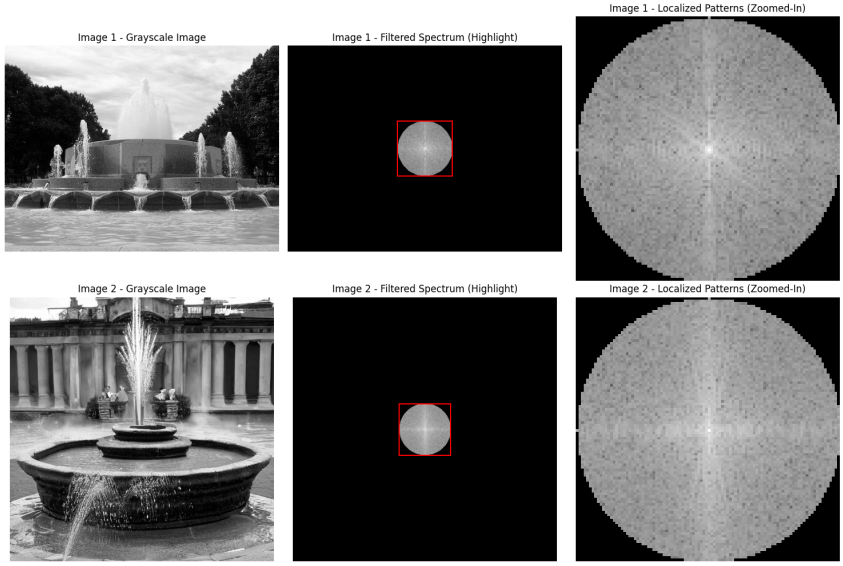

Figure 5: Zoomed-in analysis of the filtered magnitude spectrum. A subset of low-frequency components, extracted from the central region of the spectrum, emphasizes localized patterns and structures. This focused region highlights the most dominant low-frequency features, providing insights into smoother and broader variations in the data.

high-level features. The extracted embeddings were stored alongside the frequency features, forming a comprehensive feature set for classification.

3.15 Classification Methods

The classification step of the project involved combining frequency features and semantic embeddings into a unified feature vector, which was then used as input for three distinct classification models: Support Vector Machines (SVM), Neural Networks (NN), and XGBoost. Each classifier brought unique strengths to the detection process, ensuring a robust evaluation of the hybrid feature set.

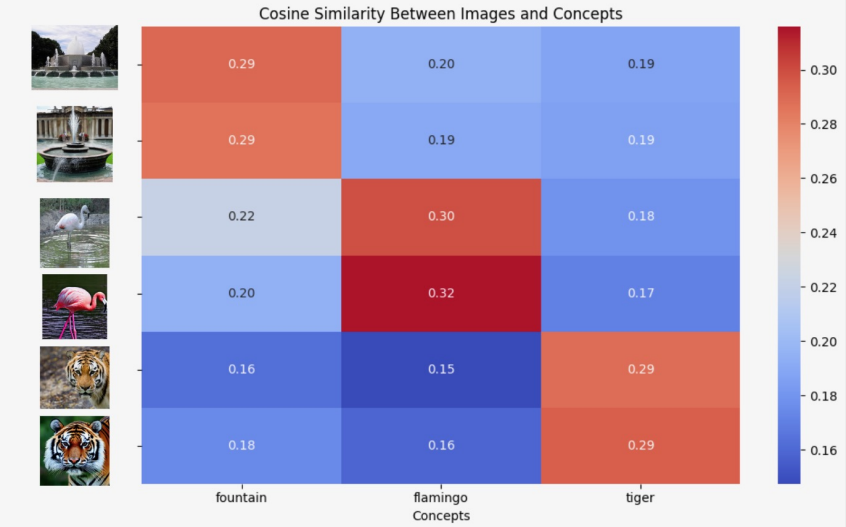

Figure 6: This heatmap visualizes the cosine similarity between CLIP embeddings of real and AI-generated images and their corresponding high-level concepts, such as "fountain," "flamingo," and "tiger." Each row represents an image, while each column corresponds to a concept, with color intensity indicating the strength of alignment—red areas signify high similarity, while blue areas indicate low similarity. This representation highlights CLIP's ability to extract and compare abstract features, providing insights into how well images align with their semantic meanings.

3.15.1 SVM

Support Vector Machine (SVM) was chosen for its ability to find optimal decision boundaries between classes. The theory behind SVM revolves around maximizing the margin between classes, which is defined as the distance between the hyperplane and the nearest data points from each class, known as support vectors. The margin γ is given by:

$$\gamma = \frac{1}{\|\mathbf{w}\|}$$

where $\mathbf{w}$ is the weight vector that defines the hyperplane.

For this project, hyperparameter optimization was conducted using Optuna, focusing on the regularization parameter C and the kernel type. The regularization parameter C controls the trade-off between maximizing the margin and minimizing the classification error. A log-uniform distribution was used for C, with values ranging from 10^{-4} to 10^{1}. The model was trained on the fused feature set and evaluated for its ability to classify images as real or synthetic.

To handle overlapping classes, SVM employs a soft margin, introducing slack variables ξ_i that allow some misclassification while balancing the trade-off between margin width and classification accuracy through the regularization parameter C. The soft margin optimization problem is formulated as:

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to the constraints:

$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where $\mathbf{x}_i$ are the feature vectors, y_i are the class labels, and ξ_i are the slack variables.

A key feature of SVM is the kernel trick, which enables the algorithm to handle non-linear data by projecting it into higher-dimensional spaces. The kernel function $K(\mathbf{x}, \mathbf{x}')$ computes the dot product in a higher-dimensional feature space without explicitly transforming the data. Popular kernels include:

$$K_{\text{linear}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}\cdot\mathbf{x}'$$

$$K_{\text{polynomial}}(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}\cdot\mathbf{x}' + c\right)^d$$

$$K_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$
$$K_{\text{sigmoid}}(\mathbf{x}, \mathbf{x}') = \tanh\left(\alpha \mathbf{x} \cdot \mathbf{x}' + c\right)$$

where c, d, σ, and α are kernel-specific parameters.

For this project, hyperparameter optimization using Optuna identified the Linear kernel as the most effective choice. The regularization parameter C was optimized to approximately 0.00138, and the kernel's parameter γ was set to "auto" for dynamic scaling. The input features were standardized using the StandardScaler:

$$\mathbf{x}_i^{\text{scaled}} = \frac{\mathbf{x}_i - \mu}{\sigma}$$

where μ and σ are the mean and standard deviation of the features, respectively.

To enable probabilistic outputs, the parameter `probability` was set to True. This allowed the model to provide confidence scores for predictions. Optuna conducted 50 trials to fine-tune these hyperparameters, resulting in a highly optimized and effective SVM model.

3.15.2 NN

Neural networks (NN) are highly flexible models that excel at capturing complex, non-linear relationships in data. For this project, a three-layer architecture was employed, optimized using Optuna for maximum performance. The first layer consisted of 512 to 2048 neurons, with 624 being the selected size. Subsequent layers were configured with progressively smaller sizes, ranging from 256 to 1024 neurons for the second layer (optimized to 551) and 128 to 512 neurons for the third layer (optimized to 138). This layered design allowed for increasingly refined feature extraction.

The network's architecture can be represented as follows:

$$\text{Layer}_1 : \text{Neurons in range of } [512, 2048], \text{ Selected: } 624$$
$$\text{Layer}_2 : \text{Neurons in range of } [256, 1024], \text{ Optimized: } 551$$
$$\text{Layer}_3 : \text{Neurons in range of } [128, 512], \text{ Optimized: } 138$$

Batch normalization was applied throughout the network to stabilize training by normalizing intermediate outputs, reducing sensitivity to input data distributions, and accelerating convergence. The batch normalization operation is defined as:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

where x is the input, μ is the mean, σ^2 is the variance, and ϵ is a small constant to avoid division by zero.

Dropout regularization was implemented with a rate p tuned between 0.3 and 0.7, with an optimal value of 0.67. Dropout randomly deactivates neurons during training to prevent overfitting, and is represented by:

$$\hat{y} = y \cdot \text{Bernoulli}(p)$$

where y represents the output of the neuron, and p is the probability that a neuron is kept active.

The network was trained over 10 epochs with a batch size of 128, balancing computational efficiency and validation performance. The learning rate η, optimized within a range of 10^{-5} to 0.1, was finalized at 0.00228 to ensure effective learning while avoiding divergence. The learning rate schedule follows:

$$\eta_t = \frac{\eta_0}{1 + \lambda t}$$

where η_0 is the initial learning rate, λ is the decay factor, and t is the epoch.

ReLU (Rectified Linear Unit) activation functions were employed throughout to introduce non-linearity, enabling the network to model intricate patterns in the fused feature set:

$$\text{ReLU}(x) = \max(0, x)$$

This configuration resulted in a robust neural network, capable of adapting to the complexities of AI-generated image detection.

3.15.3 XGBoost

XGBoost is a gradient-boosted decision tree algorithm renowned for its efficiency and predictive accuracy. The algorithm operates within a gradient boosting framework, combining weak learners—in this case, shallow decision trees—into a strong learner by iteratively minimizing errors. For classification tasks like real versus fake detection, XGBoost optimizes using a binary logistic objective, which directly minimizes classification errors by focusing on probabilistic predictions.

Regularization parameters, such as `lambda` (L2 regularization) and `alpha` (L1 regularization), are incorporated to control overfitting and improve generalization. The `n_estimators` parameter defines the number of trees to be built in the ensemble, with higher values potentially increasing the model's capacity to fit complex patterns but also risking overfitting. The `max_depth` parameter specifies the maximum depth of each tree, which determines how deeply the model can split the data; deeper trees can capture more intricate relationships but may also lead to overfitting if not properly constrained. Lastly, `colsample_bytree` controls the fraction of features to be randomly selected for each tree, introducing variability that can help prevent overfitting and improve generalization by ensuring that the model does not rely too heavily on specific features. These parameters are critical for fine-tuning the trade-off between model accuracy and generalization.Additionally, the evaluation metric `logloss` ensures that the model's probabilistic outputs are fine-tuned to minimize errors.

For this project, hyperparameter optimization was conducted using Optuna to maximize accuracy over 50 trials. Key hyperparameters included:

- `lambda`: Tuned within a range of 10^{-4} to 10.0, with an optimal value of 0.2535.
- `alpha`: Tuned within a range of 10^{-4} to 10.0, with an optimal value of 0.000268.
- `learning_rate`: Ranged from 10^{-4} to 0.1, with an optimal value of 0.094055.
- `n_estimators`: Adjusted between 100 to 500 trees, settling at 375.
- `max_depth`: Configured between 3 to 10, with an optimal depth of 5.
- `colsample_bytree`: Tuned between 0.3 and 1.0, with the optimal value being 1.

These hyperparameters were carefully tuned to balance model complexity and generalization. XGBoost demonstrated exceptional performance on the fused frequency and semantic feature set, effectively capturing intricate feature interactions and delivering robust predictions. Its capability to handle missing values and feature irregularities further cemented its role as a top-performing classifier in this study.

4 Novelty of Approach

This project introduces a hybrid detection model that leverages both frequency domain features and CLIP-based semantic embeddings, providing a novel way to detect AI-generated images. Existing methods typically focus on a single feature type—either pixel-level or semantic features—leading to limitations in generalization and robustness. By integrating these two complementary feature sets, our approach addresses a broader spectrum of image characteristics, making it more resilient to diverse generative models and datasets.

The fusion of Fourier Transform-based features with CLIP embeddings ensures a multi-level analysis of image content. The Fourier Transform captures pixel-level anomalies, such as edge artifacts and repetitive textures, which are often indicative of synthetic origins. Simultaneously, CLIP embeddings provide high-level semantic insights, enabling the detection of conceptual inconsistencies that might be missed by frequency analysis alone. The direct concatenation of these features into a simplified fusion layer reduces computational overhead, ensuring scalability and adaptability across hardware with limited resources.

Moreover, this hybrid model introduces enhanced generalization capabilities by bridging the gap between low-level and high-level feature analysis. While traditional methods struggle with adapting to emerging generative models, this approach dynamically learns from both pixel-level irregularities and semantic mismatches. This makes the model especially robust against novel image synthesis techniques, including diffusion-based models, which often escape detection by relying on sophisticated algorithms to mask visual artifacts.

Additionally, the model's design prioritizes efficiency and scalability. By streamlining the fusion of features and employing optimized machine learning algorithms like XGBoost and SVM, the system is suitable for deployment in real-time applications, such as media verification systems and content moderation platforms. This adaptability ensures that the detection framework remains effective even when integrated into resource-constrained environments, such as mobile devices or cloud-based systems handling high volumes of data.

Finally, the hybrid model's capability to generalize across diverse datasets and image manipulation techniques sets a new benchmark in AI-generated image detection. By combining efficiency, robustness, and adaptability, this project not only advances current methodologies but also opens avenues for broader applications, including video and audio synthesis detection, ensuring its relevance in combating synthetic media across multiple domains.

5 Demonstration

This section presents the web-based application developed to demonstrate the practical implementation of the AI-generated image detection framework. The frontend interface, designed using Streamlit, provides a user-friendly platform for real-time analysis and classification of images.

5.1 Interactive Interface

The application interface allows users to interact with the detection framework seamlessly. Its primary functionalities include:

- **Image Upload:** Users can upload images in JPEG or PNG format for analysis. The system processes the uploaded image to extract both Fourier features and CLIP embeddings.
- **Model Selection:** Users can choose from three pre-trained classification models (XGBoost, SVM, or Neural Network) based on their preferences.
- **Prediction Output:** The system displays the predicted class label (Real or Fake) alongside confidence scores in real-time. This output is presented clearly to enhance usability.

5.2 System Highlights

The application emphasizes efficiency, accessibility, and transparency:

- **Efficiency:** Optimized caching mechanisms reduce computational overhead, ensuring that predictions are delivered promptly without extensive resource use.
- **Transparency:** Real-time logs and progress indicators provide feedback during the image analysis process, improving user experience and error handling.
- **Accessibility:** The web-based platform is compatible across various devices, eliminating the need for specialized software installations.

5.3 Demonstration Workflow

The application workflow is straightforward and intuitive:

1. Users upload an image file (JPEG or PNG) via the interface.
2. The system extracts frequency-based features using the Fourier Transform and semantic embeddings using the CLIP model.
3. The selected classifier processes the features and predicts whether the image is Real or Fake.
4. Results, including the predicted label and confidence score, are displayed in real-time on the interface.

This demonstration highlights the system's ability to integrate complex feature extraction and classification methodologies into a simple, interactive tool, enabling practical applications for AI-generated image detection.

6 Prediction Results

To illustrate the functionality of the application, this section presents two sample predictions made using the web-based frontend: one for a real image and another for an AI-generated image.

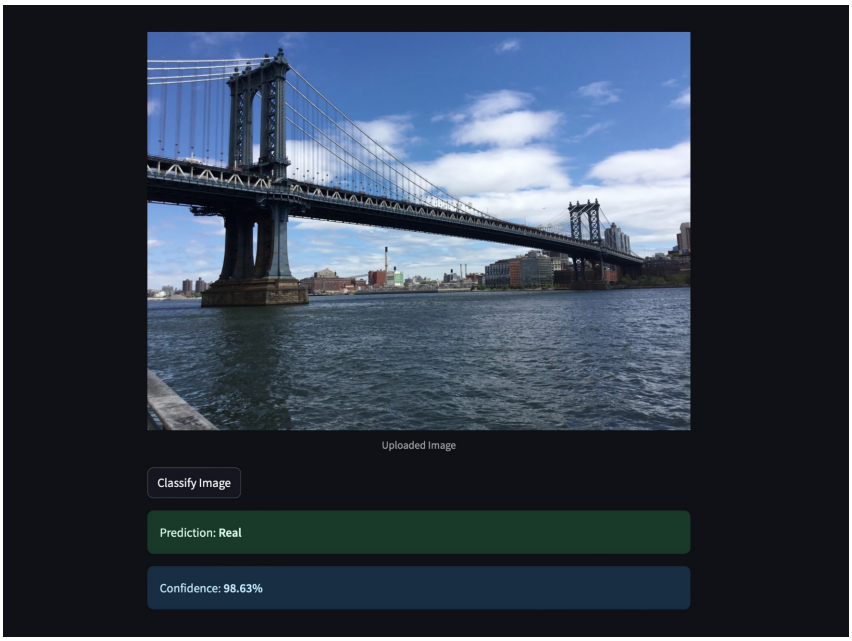

Figure 7: Example of a prediction for a real image. The system classifies the image as **Real** with a high confidence score, as shown in the interface.

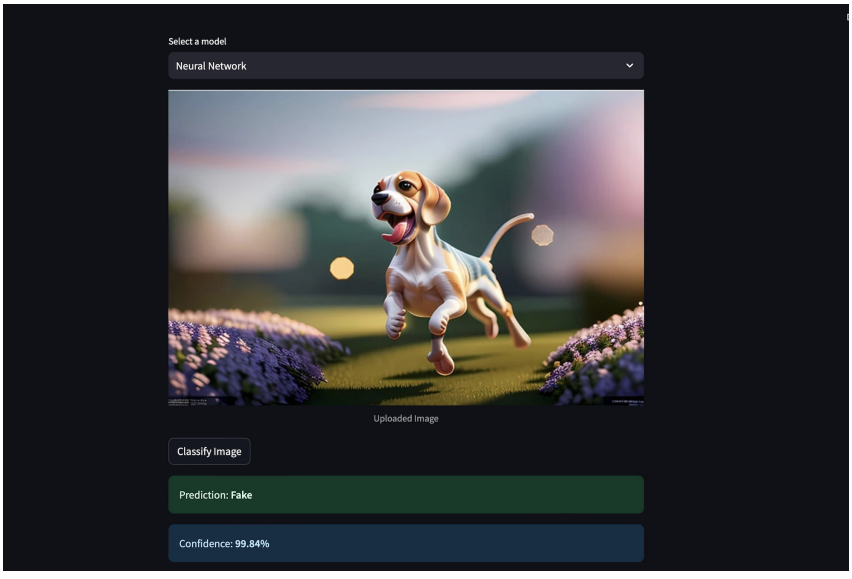

Figure 8: Example of a prediction for an AI-generated image. The system classifies the image as **Fake** with a confidence score, demonstrating its ability to detect synthetic content.

7 Results

7.1 Overall Performance

The hybrid detection model was evaluated using three classifiers - XGBoost, SVM, and NN - on a dataset containing real and AI-generated images. Among these, XGBoost demonstrated the best overall performance, achieving an accuracy of 99.3% and an AUROC of 0.9998. These metrics indicate that XGBoost was highly effective at distinguishing between real and synthetic images, offering both high precision and robust generalization. NN followed closely with an accuracy of 98.9% and an AUROC of 0.9996, while SVM achieved an accuracy of 98.% and an AUROC of 0.9990. Despite slight differences in accuracy, all classifiers displayed strong overall performance, validating the effectiveness of the hybrid feature set.

Metrics	XGBoost	SVM	Neural Network
Accuracy	**0.993**	0.98575	0.98925
Precision	**0.993**	0.98551	0.98417
Recall	0.993	0.986	**0.99450**
F1-score	**0.993**	0.98575	0.98931
AUROC	**0.999755**	0.999036	0.999555

Table 1: Overall Metrics Results

7.2 Class-level Performance

When examining performance at the class level, XGBoost excelled in detecting both real (Class 0) and fake images (Class 1). For each class, it achieved precision, recall, and F1-scores of 0.993, reflecting a balanced ability to identify both image types without significant trade-offs. NN exhibited a slight imbalance, with Class 0 showing lower precision (0.98400) but higher recall (0.98919), whereas Class 1 had higher precision (0.99450) but lower recall (0.98931). This indicates that the neural network prioritized identifying fake images over real ones. SVM, while consistent, displayed marginally lower metrics across both classes compared to the other two classifiers.

7.2.1 Class 0 (Real Images)

Metrics	XGBoost	SVM	Neural Network
Accuracy	**0.993**	0.98599	0.99444
Precision	**0.993**	0.98550	0.98400
Recall	**0.993**	0.98575	0.98919
Support	2000	2000	2000

Table 2: Per-class Metrics Results (Class 0)

7.2.2 Class 1 (Fake Images)

Metrics	XGBoost	SVM	Neural Network
Accuracy	**0.993**	0.98551	0.98417
Precision	**0.993**	0.986	0.99450
Recall	**0.993**	0.98575	0.98931
Support	2000	2000	2000

Table 3: Per-class Metric Results (Class 1)

7.3 Weighted Averages Performance

In terms of weighted averages, XGBoost continued to outperform, maintaining precision, recall, and F1-scores of 0.993 across the dataset. This consistency across metrics underscores its robustness and reliability. NN achieved weighted precision and recall of 0.9893 and an F1-score of 0.9892, slightly

trailing XGBoost but still demonstrating strong performance. Meanwhile, SVM achieved weighted metrics around 0.9858, indicating solid but slightly lower performance in aggregating predictions across the dataset.

Metrics	XGBoost	SVM	Neural Network
Accuracy	**0.993**	0.98575	0.98930
Precision	**0.993**	0.986	0.98925
Recall	**0.993**	0.98575	0.98925
Support	4000	4000	4000

Table 4: Weighted Averages Metrics Results

8 Learnings from Deep Learning Course

The project on AI-Generated Image Detection using Simplified Hybrid Feature Fusion integrates several key concepts and techniques from the Deep Learning course. This section outlines the relevant learnings and how they inform both the current project design and potential future extensions.

8.1 Concepts Used in Current Project Design

- **Supervised Learning**: The project employs supervised learning principles, where data consists of pairs of input images (x) and their corresponding labels (y) indicating whether they are real or AI-generated. This aligns with the course's goal of learning a function to map $x \rightarrow y$, which is foundational in tasks such as classification, object detection, and image segmentation. The classification methods utilized—Support Vector Machines (SVM), Neural Networks (NN), and XGBoost—are direct applications of these supervised learning techniques.
- **Feature Extraction**: The project leverages advanced feature extraction methods, specifically the Fourier Transform for frequency domain features and the Contrastive Language-Image Pretraining (CLIP) model for semantic embeddings. This approach reflects the course's emphasis on learning meaningful representations from data, as highlighted in the discussion on autoencoders and their ability to capture underlying structures within datasets. By combining low-level pixel patterns with high-level semantic representations, the model enhances its capability to detect synthetic artifacts across various generative models.
- **Generative Modeling**: The project directly addresses the challenges posed by generative models, which are designed to create new samples that mimic a given distribution. The course materials emphasize that generative models can be used for tasks such as artwork generation, super-resolution, and colorization, all of which are relevant to understanding the context in which AI-generated images are produced. The hybrid detection model aims to discern these generated images from real ones by analyzing their inherent characteristics.

8.2 Concepts for Potential Extension

- **Variational Autoencoders (VAEs)**: The course covered VAEs as a probabilistic extension of traditional autoencoders, which allow for sampling from learned distributions. Incorporating VAEs into the project could enhance its detection capabilities by enabling the model to generate synthetic images for training or to learn more robust latent representations of both real and generated images. This could improve the model's ability to identify subtle differences between classes.
- **Latent Space Exploration**: The concept of latent space analysis discussed in class could be applied to better understand how different dimensions of latent variables encode interpretable factors of variation in images. By analyzing these latent representations, insights could be gained into the distinguishing features of AI-generated images versus real ones, potentially leading to more effective detection strategies.
- **Self-Supervised Learning**: The course introduced self-supervised learning as a method to generate useful features without manual labels. This approach could be integrated into the current project by utilizing large amounts of unlabeled image data, allowing for improved feature extraction and robustness against various generative techniques.

- **Generative Adversarial Networks (GANs)**: Although not explicitly covered in the provided materials, GANs represent another avenue for extension. These networks consist of two competing models—a generator and a discriminator—that could be employed to refine the detection model further by training against increasingly sophisticated generated images. This competitive training could enhance the model's resilience against novel generative techniques.
- **Unsupervised Learning Techniques**: The discussion on unsupervised learning emphasizes discovering hidden structures within data without relying on labels. Techniques such as clustering or dimensionality reduction could be explored to identify patterns in unlabeled data, potentially revealing new insights into distinguishing characteristics between real and AI-generated images.

By applying these concepts from generative modeling, the project can not only enhance its current detection capabilities but also adapt to evolving challenges posed by advancements in generative technologies.

9 Discussion

Despite the promising results, the proposed hybrid detection model has certain limitations that warrant further exploration. First, the model heavily relies on the extracted frequency and semantic features, which may not fully capture novel artifacts introduced by emerging generative models. This dependency could limit its adaptability to entirely new image generation techniques. Second, while the Fourier Transform and CLIP embeddings are computationally efficient for feature extraction, the combined process can be resource-intensive, particularly for large-scale datasets or deployment in real-time applications on resource-constrained hardware. Another limitation is the reliance on a dataset that may not comprehensively represent the diversity of real-world AI-generated images, potentially impacting the model's ability to generalize across all types of synthetic content.

For instance, the model was initially trained on the CIFAKE dataset, which is relatively controlled and lacks the diversity of real-world AI-generated images. When tested on real-world images, the model's performance was significantly poorer, indicating a strong dependency on the dataset. Although the model's performance improved with additional training on more diverse data, there is still considerable room for improvement in terms of generalization. Additionally, while the classifiers showed robust performance, minor imbalances, particularly in Neural Networks, highlight the need for further fine-tuning of hyperparameters and architecture.

Future work could explore integrating SLIP (Self-supervised Learning with Image and Text Pre-training) embeddings into the model. SLIP has shown promise in improving the robustness and generalization of vision-language models by combining contrastive image-text learning with self-supervised pretraining on images. Leveraging SLIP could potentially enhance the model's ability to capture diverse and nuanced features in both real and AI-generated images, making it more adaptable to emerging generative techniques. Moreover, extending the dataset to include a broader range of synthetic and real-world images, along with continued optimization of the hybrid architecture, would be critical steps to improve both accuracy and scalability.

10 Conclusion

The project on AI-Generated Image Detection using Simplified Hybrid Feature Fusion integrates several key concepts from the generative modeling course, particularly in the areas of supervised learning and feature extraction. The classification methods employed—Support Vector Machines (SVM), Neural Networks (NN), and XGBoost—directly apply supervised learning techniques, where the goal is to learn a function that maps input data (x) to labels (y). This aligns with the course's focus on classification tasks, where models are trained on labeled data pairs to predict outcomes for new, unseen inputs.

The hybrid model's feature extraction approach, combining Fourier Transform for frequency domain features and CLIP for semantic embeddings, reflects the course's emphasis on learning meaningful representations from data. This fusion of low-level pixel patterns and high-level semantic information mirrors the concept of autoencoders discussed in the course, which aim to capture underlying

structures within datasets. By leveraging both frequency domain and semantic features, the model enhances its capability to detect synthetic artifacts across various generative models, addressing the challenges posed by advanced AI image generation techniques like Stable Diffusion and DALL-E.

The project's focus on detecting AI-generated images directly relates to the core concepts of generative modeling covered in the course. While the course materials primarily discuss generative models from the perspective of creation (e.g., VAEs and GANs), this project approaches the topic from the detection angle. This application demonstrates a practical use case for understanding the characteristics of generated data, aligning with the course's exploration of how generative models learn to approximate data distributions. The success of XGBoost in this context, achieving 99.3% accuracy, highlights the effectiveness of ensemble methods in leveraging the rich, complementary information provided by the fused feature set, showcasing the potential of combining multiple machine learning techniques for complex tasks like AI-generated image detection.

References

[1] T. R. Anon and J. I. Emon. Detecting the undetectable: Combining kolmogorov-arnold networks and mlp for ai-generated image detection. *ArXiv preprint*, abs/2408.09371, 2024.

[2] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9, 2024.

[3] B. Bayar and M. C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] J. Durall, M. Keuper, and F. Pfreundt. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1738–1746, 2020.

[6] A. Frank et al. Leveraging frequency domain information for deepfake image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):824–830, 2021.

[7] A. Radford et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[8] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[9] Z. Qi, Y. Chen, and X. Dong. Fusion of frequency and spatial features for aigi detection. In *Proceedings of the 2022 ACM International Conference on Multimedia*, 2022.

[10] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2022.

[12] C. Saharia, W. Chan, S. Saxena, and M. Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2202.12802*, 2022.

[13] R. Tolosana, S. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.

[14] S. Yan, O. Li, J. Cai, Y. Hao, X. Jiang, Y. Hu, and W. Xie. A sanity check for ai-generated image detection. *ArXiv preprint*, abs/2406.19435, 2024.

[15] Y. Zhang, S. Wang, X. Tan, and J. Yuan. Image forensics on gan-generated images. *IEEE Transactions on Information Forensics and Security*, 15:234–242, 2020.

[16] Y. Zhou and X. Zhang. Contrastive learning for image synthesis detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.