Detecting AI-Generated Images Created by Diffusion models

Arsh Banerjee
Adviser: Dr. Xiaoyan Li [IW 02]
Independent Work Report - Spring 2023

May 2, 2023

Abstract

The proliferation of artificial AI-Generated images has led to the prevalence of high-quality images being published online, with some being used to spread disinformation. This project involves the development of a tool for detecting AI-generated images, specifically from diffusion models, to help counter misinformation and qualitatively understand identifying key image attributes for this class of classification task. This paper will discuss the quantitative success of using convolutional neural networks trained on differing image features, the qualitative findings regarding the architecture of diffusion models, and ultimately work towards understanding how to counter the coming age of AI-powered deepfakes.

Contents

1. Introduction

The use of misinformation to control the masses dates back to the sixth century when historian Procopius of Caesarea rewrote past events to discredit Emperor Justinian to benefit himself [1]. In that era, misinformation traveled by word of mouth and papyrus, taking years, if not decades, to circulate throughout the population. After the invention of the printing press, American writer Edgar Allan Poe wrote at least six fake news stories, at least one of which involved a false story of a hot-air balloon crossing the Atlantic [1]. Still, the stories in the paper took weeks to circulate and could be easily retracted. Today, in an age of unprecedented digital connectedness, it has never been easier to propagate disinformation that fosters discursive conflict. With the availability of computing power online, anyone with internet access can generate realistic images depicting any prompt in a matter of seconds.

Practical image generation was first introduced in 2014, with the seminal paper related to generative adversarial networks (GAN)s that discussed the possibility of a generative and discriminative model being used to learn and generate samples of a dataset [5]. The generator model learns from the data distribution to try and create new data, while the discriminator attempts to distinguish between that generated data and the real data. Alternating rounds, the generator receives feedback from the discriminator to optimize its data creation, and the discriminator learns from the generator to better distinguish between the classes. Over time, both incrementally become better at their respective tasks and the generator model ultimately is able to create new data indistinguishable from the originals. In 2016, GANs were being used to generate new sample data for simple datasets like MNIST [10] with high degrees of success, and shortly after, GANs were utilized to generate realistic images of faces. Figure 1 shows the results of Yu & Porikli, which represented some of the highest quality results during the time of publication.

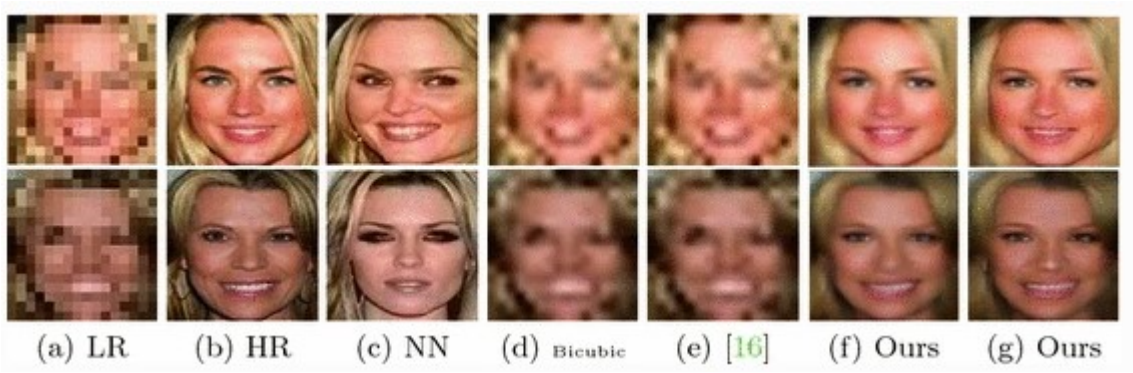

Figure 1: Results from Yu & Porikli's generation of faces using GANs

While utilizing GANs, researchers found that its architecture was prone to deconvolution and checkerboard artifacts which affected the overall quality of generations and limited the maximum resolutions for the images [6]. With research focused on high-resolution image generation, a new class of models emerged known as denoising diffusion probabilistic models, or simply diffusion models. One of the more popular diffusion models was created by the OpenAI organization and named Dalle 2. In a paper describing the architecture of the model and its quantitative and qualitative results, its authors noted it was able to "produce higher-quality samples" and "image diversity with minimal loss in photorealism" [7]. Figure 2 shows the results of the OpenAI, which illustrate the high resolution, image quality, and diversity of the possible generations.

Figure 2: Results from Dalle-2

Diffusion models have created the possibility for anyone with computing power to generate realistic images depicting any range of unique and diverse prompts. Everyone, from lone individuals to

state actors, gain the ability to distort reality, creating images depicting people doing things they have never done. Chesney and Citron discuss one notable example in "Deep fakes: A looming challenge for privacy, democracy, and national security" [2]. After the tragic shooting at Marjory Stoneman Douglas High School in Parkland, Florida, a student Emma Gonzalez published an image depicting her ripping up a piece of paper with a bullseye. Actors created an altered version of the image of her instead ripping up the Constitution which led to a barrage of attacks against her [2]. AI-generated images and videos will undoubtedly have negative consequences, from being used to humiliate or attack individuals to sowing discourse among the masses.

With so much potential harm, this project sought to create a tool, that when given an image, can generate a classification to determine whether or not it is AI-generated. Specifically, the goal of this project was to create a high-accuracy model for this binary classification task (AI-Generated or Real) and qualitatively understand what features of images can be exploited in order to better detect them.

2. Related Work

This section will discuss existing work regarding the detection of AI-generated images. Prior work in this area mostly revolves around the previous "generation" of generative adversarial network (GAN) models, while there are only a few papers discussing diffusion model detection due to their recency.

With the speed of mass distribution of images and videos on the internet, researchers quickly realized that systems needed to be developed in order to detect these generated images and videos as they became more realistic. In 2022, a handbook was published as part of a series of books regarding Advances in Computer Vision and Pattern Recognition discussing "Digital Face Manipulation and Detection" [8]. The handbook served to summarize the detection of generated images; however, the handbook and related papers almost exclusively discussed GAN models. It high-

lighted several notable methods, which included the concept of GAN fingerprints, detecting case asymmetries, color features, and data-driven features. The system for detecting GAN fingerprints was particularly successful. The method is described by Zhang et. al, who identified that the upsampling component of the GAN pipeline left behind a spectrum artifact that essentially acted like a fingerprint in every image, allowing them to not only detect fake images but also to identify the model it came from [12]. Specifically, while GAN architectures are diverse, the upsampling components have common elements, and their research identified that the upsampling component (transposed convolution) created a checkerboard artifact that could be detected via Fourier transformation. Figure 3 shows an example of detecting this spectral artifact from an AI-Generated image.

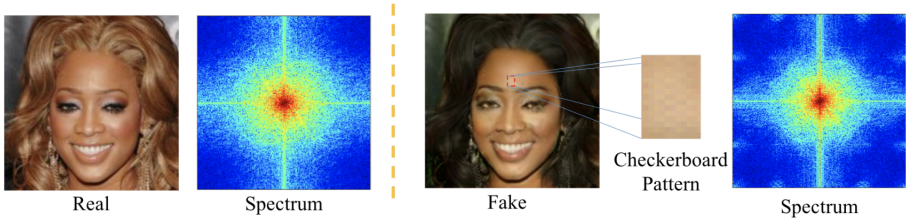

Figure 3: Example from Zhang et al. depicting artifact detection

Instead of training a neural network on raw pixels, these spectral inputs were used as input to a GAN image classifier with a classification rate of 0.95. While the research focused specifically on two popular models, CycleGAN and AutoGAN, since the upsampling component is common, the classifier generalized well to other GAN generators. Apart from exploiting the architecture of the models, researchers also attempted to use data-driven learning (convolutional neural networks trained on the pixels themselves). Roy et al. explored different CNN architectures and models to detect deep fakes and obtained a 0.9550 classification rate with a ResNet using 93,647,040 trainable parameters [8]. In addition to these approaches, researchers also trained classifiers on visual artifacts (blurriness, face warping) to detect AI-Generated images. Y. Li et al. trained classifiers on boundary artifacts, facial artifacts, and color mismatches and achieved an average accuracy of 0.92. However, on certain GAN models, this reached as high as 0.99 [8]. Past research into GAN

models clearly shows that the classification task is surmountable and can be completed with extremely high levels of accuracy, leveraging attributes about model architecture and image artifacts.

Although GAN image detection is a well-researched area, Diffusion models like Dalle-2 utilize a different architecture that is not prone to the same weaknesses exploited in GAN models. In the paper describing the creation and architecture of Dalle-2, its authors note that "For the first upsampling stage, we use gaussian blur, and for the second, we use a more diverse BSR degradation" [7]. This upsampling layer was created by Zhang et al. and incorporates the following: "noise is synthesized by adding Gaussian noise with different noise levels, adopting JPEG compression with different quality factors, and generating processed camera sensor noise" [11]. Figure 4, shows examples of the upsampling architecture on a degraded image.

Figure 4: Example from Zhang et al. depicting image upsampling

Because of the new architecture of diffusion models, "checkerboard artifacts" are not found in images, nor are visual artifacts as common. Accordingly, prior methods used on GAN networks to achieve high classification rates do not transfer to this new class of models. Moreover, due to its recency, research into diffusion model detection is limited. In January 2023, Sha et al. conducted research on the "Detection and Attribution of Fake Images Generated by Text-to-Image Generation Models" [9]. These Text-to-Image Generation Models include diffusion models such as Dalle-2 and Stable Diffusion. Sha et al.'s work focuses on experiments using two datasets: MSCOCO and Flickr30k. Both datasets provide real images as well as textual prompts describing the image. The experiments utilize diffusion models to generate images based on these prompts in order to train classifiers to conduct binary classification between the "real" and fake" images. Figure 5 shows the accuracies of the models trained by Sha et al.

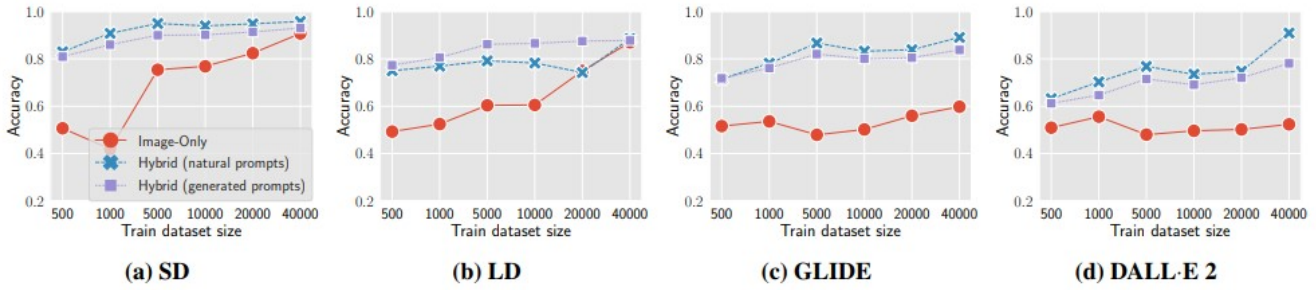

Figure 5: Accuracies of classifiers on different diffusion models [9]

The graph above shows that the models achieved accuracies of around 0.9 for the latent diffusion and glide models and around 0.95 for Dalle 2 and stable diffusion models. It should be noted that in this classification task, the models utilized information regarding the prompt as well as the image. As such, the model would likely not generalize well to just a dataset of images with prompts. Still, this work shows that these models leave informative features that can be leveraged to perform classification. A key distinction to recognize in this research is that model was trained on real-fake image pairs describing the same prompt, and the model utilized the prompt during classification. Moreover, the MSCOCO and Flickr30k datasets are almost entirely comprised of images of real objects in real-world settings.

Prior work heavily centers around the specific class of GAN models. The findings, such as the GAN fingerprints, visual artifacts, and color artifacts, highlight areas of interest for AI-Generated image detection. The work related to diffusion models is extremely limited and focuses on specific use cases; however, they serve as evidence that the classification of diffusion-generated images can be accomplished. Accordingly, not only was there a clear space to research the detection of generated images from diffusion models like Stable Diffusion and Dalle-2, but by analyzing real images as well as digital ones (digital drawings/sketches) this project could substantially build upon existing work. Without utilizing prompts and building a detector to only use raw images as input, this project would allow us to learn about how to approach diffusion-image classification as well as provide qualitative insights about the models themselves.

3. Approach

To build this system to detect images generated via diffusion models and to achieve the described goals, two convolutional neural networks were trained to create an ensemble model to perform classification. Ensemble models have been proven to have better generalization performance than individual models, and this effect is only strengthened with an ensemble of deep learning models [4]. Ideally, the ensemble model will be more robust which is needed due to the more diverse image class. The most consequential aspect of the approach however was identifying good predictive features to train the models on. To identify possible features, exploratory data analysis was conducted on the training portion of a dataset comprised of 10,000 images: 5,000 from a set of three diffusion models and 5,000 real images. This analysis, further described in the implementation section, showed color histograms of the images and the raw pixel data were viable features for training. This approach differs from prior work mainly in that it focuses on different predictive features compared to methods related to GAN image detection. The differing architecture of diffusion models calls for a new approach that is more robust. Furthermore, for the work that exists related to diffusion models, this approach performs classification on a broader class of images (not only real objects) using only the images themselves without prompts. Ideally, it would allow the model to correctly classify any image found online without any associated prompt creating a practical use case.

4. Implementation

4.1. Dataset

One critical aspect of this project was the dataset itself and the diversity of the image classes. While prior works related to diffusion models utilized datasets of real objects and their associated prompts to generate images from the diffusion models, this project utilized publicly available compilations of user creations from diffusion models (see Appendix). The compiled dataset contained 5,000 images labeled "AI-Generated" from the Stable Diffusion, Dalle, and Dalle-2 models with

each model having an equal sample distribution. It also contained 5,000 "real" images compiled from photographs, movies, human-drawn digital art, and portraits. Importantly, the dataset is representative of the image distribution that may be encountered online or in practical use as an "AI Image Detector." Mainly, the "Real" and "AI-Generated" image classes are not limited to images of common objects (person, umbrella, dog) and their associated prompts as in the MSCOCO dataset. Instead, images labeled as "AI-generated" are sampled from user creations from these particular models, making the sample space more diverse. Similarly, the class of real images is also made more diverse by incorporating human-drawn digital art. For instance, take the following example depicted in Figure 6.

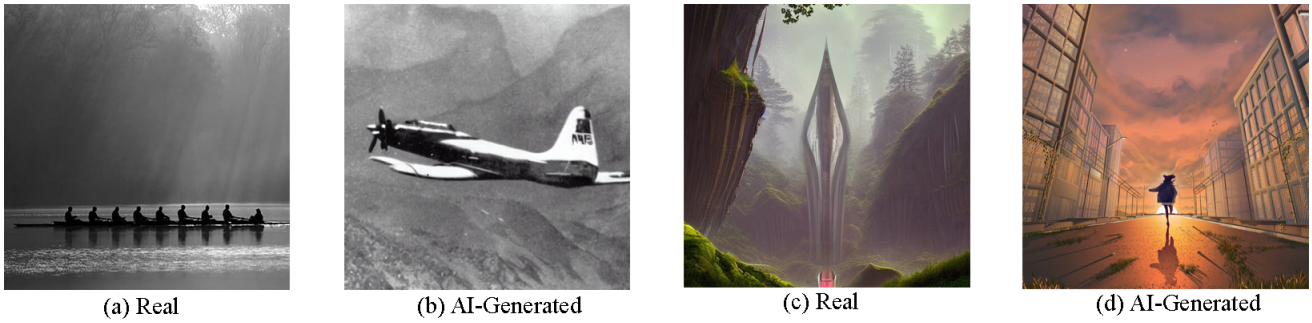

Figure 6: Example Images from Compiled Training Dataset

Figure 6 shows samples of images from the compiled dataset used to train the classification models. Examples (a) and (b) depict what "realistic" images may look like, while examples (c) and (d) depict digital art. Intrinsically, images of the real world have a ground truth of what objects look like, making classification between (a) and (b) a simpler task for machine learning models to learn. On the other hand, digital art has no ground truth making classification for this subclass a harder task. The goal of compiling together this diverse dataset was to train a model that can generalize well to all image generations, not just a specific class.

4.2. Data Analysis and Feature Selection

A critical aspect of any machine learning task with images is selecting the right features for training. The initial set of features evaluated were those highlighted in prior work related to GANs, specifically the classifier trained on GAN fingerprints. The following figure shows the

implementation of Zhang et al.'s methodology for spectral analysis [12].

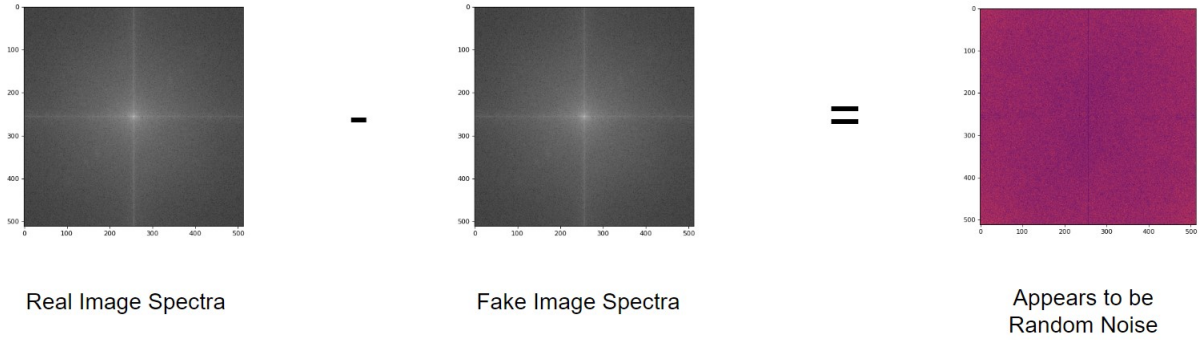

Figure 7: Spectral Analysis of Training Dataset by Class

Applying the python library SciPy's implementation of Fast Fourier transform on a set of 1,000 AI-Generated images and 1,000 real images, visually it appears that there is no discernable difference, certainly not to the extent of the "checkerboard" pattern seen in Zhang et al.'s work related to GANs. Qualitatively, this is explained by the upgraded upsampling components present in diffusion models, such as the BSR degradation layer in Dalle-2 described previously.

Another feature used in image tasks utilizes color information, which is often represented as a global histogram. A global color histogram essentially visualizes the joint probability of intensities of the different color channels. It is also translation, scale, and rotation invariant, thus, global histograms are robust for machine learning tasks. We evaluate this feature visually by computing the global histogram for 1,000 AI-Generated images and 1,000 real images and then plotting the average of these distributions. Figure 8 depicts the average global color histogram for the two classes of images.

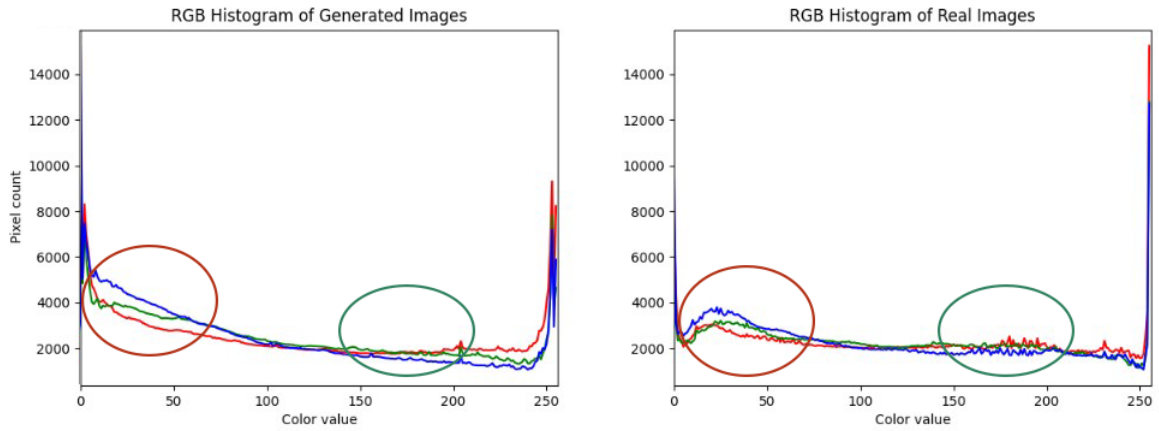

Figure 8: Histogram Analysis of Training Dataset by Class

Through this analysis, there are two main insights that are revealed. First, the color distribution in generated images appears to be more continuous, while it is more discretized for real images. This is particularly visible in the pair of ellipses drawn above, where distribution in generated images appears to be near linear while it oscillates heavily in real images. The second insight is that for the range [240-256] (the most extreme color values) real images tend to distribute heavily toward the high values, whereas generated images appear to normalize the values with this range. Since the histograms appeared visually to contain distinctive information that could be used for prediction, one of the features selected to train the models was global color histograms. Another feature often used for image classification is the raw pixels themselves in combination with a neural network that extracts and learns its own features. As such, the image information was selected as the second feature.

4.3. Structure and Pipeline

The classification system is contained within a series of python scripts (see Appendix for the link to the GitHub repository). The scripts utilize a variety of python libraries, namely numpy and scikit-learn for image preprocessing and feature extraction, while TensorFlow is used to initialize and train the two convolutional neural networks. The entire pipeline takes a single image as input, performs preprocessing and feature extraction, then, using those features, generates a binary prediction as to whether the image was AI-generated or real.

1. Preprocessing: The dataset was first cleaned by removing any images with heights or widths smaller than 200 pixels as well as any images with $\neq 3$ color channels. The images are represented as a numpy matrix of size [height $\times$ width $\times$ 3].
2. Feature Extraction: Using numpy's histogram function, a histogram is created for each color channel in the image matrix. 257 bins are used in the histogram as pixels can take values between [0,256] for any given color. This ensures there is a bin for every possible pixel value. This feature is represented as a numpy matrix of size [257 $\times$ 3].

Regarding the feature utilizing pixel information, the tool was designed with the intent to accept any resolution/size input image. Accordingly, since neural networks have fixed input sizes the initial attempt was to simply resize the image to the size 200 $\times$ 200 and utilize the resulting matrix as the input. Empirically, however, this was found to have a low classification rate. Because the input images were not bound by a certain aspect ratio or size, arbitrarily resizing it to 200 $\times$ 200 pixels resulted in heavy distortion. Instead, a random 200 $\times$ 200 pixel crop is taken of the image at its original resolution and size. Figure 9 illustrates using the random crop versus resizing.

Figure 8: Histogram Analysis of Training Dataset by Class

Empirically, the random crop method preserved the resolution and information and led to better predictive performance. This feature was created by selecting a random point in [height $\times$

width] using numpy, then forming and extracting a [200 × 200] box around that point.

3. Preparing the Dataset: Once the features are extracted for all images in the dataset, we are left with two matrices. One containing the histogram vectors for every image and another containing a random crop from every image. In order to make the convergence of the neural networks faster during training, the matrices are normalized. Then. the python library scikit-learn is used to split each matrix into a train, validation, and test set with each having 80%, 10%, and 10% of the data, respectively.

4. Model Architecture: Since we are dealing with two-dimensional features (images and histogram), Convolutional Neural Networks (CNNs) were used due to their ability to process and learn from complex spatial data. Images inherently have spatial information as the non-linear arrangement of pixels constitute objects and patterns. The histogram vector also contains spatial information regarding interactions between the different color channels. As a result, CNNs were utilized to perform the classification task. In order to create the CNNs, TensorFlow, a library developed by the Google Brain team for training deep learning models, was used. The following depicts the architecture used for the CNN with random crops as the input feature.

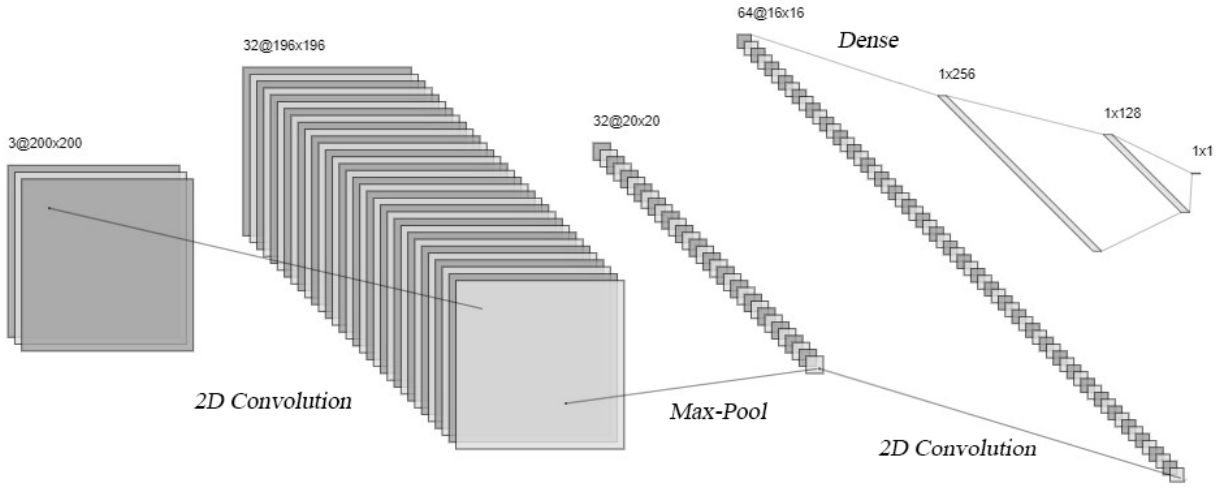

Figure 9: Architecture of CNN One

A Convolutional Neural Network has several key components: activation functions, layer types, and the loss function. The architecture depicted above utilized six layers, the first of which is a convolutional layer with 32 filters with kernel size 5x5. The large number of filters and small kernels are intended to increase the expressivity and discriminative power of the model. Smaller kernels are able to identify smaller patterns within an image, while the number of filters increases the number of patterns that can be captured. This design choice was informed by prior work related to GANs with identified small blurry patches or small patches of noise in images. Ideally, this small filter size will place the model in the best position to learn these patterns. This is followed by a Max Pooling layer which serves to reduce the dimensionality of the feature maps in order to minimize training time and model size. This is again followed by another convolutional layer with a larger filter size. Both convolutional layers are immediately followed by a dropout layer, which randomly discards 20% of the outputs from the given layer. Dropout acts as a regularization technique that helps the model to generalize better instead of relying on any particular set of weights in the network. Another regularization technique used is TensorFlow's "kernel_regularizer." This is applied to every layer of the network, and it adds a penalty term of the weights of the layer proportional to the norm of the weights. Accordingly, it encourages the model to use smaller weights which helps prevent overfitting. After the convolution and pooling layers, there are a series of fully connected layers that ultimately output a single value. The activation functions in all layers apart from the last are ReLu, which is defined as the following:

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

ReLu was chosen as it is computationally efficient and increases nonlinearity between layers. The final layer utilizes a sigmoid activation function which outputs an easily interpretable probability value between 0 and 1 regarding the predicted class probability of the image.

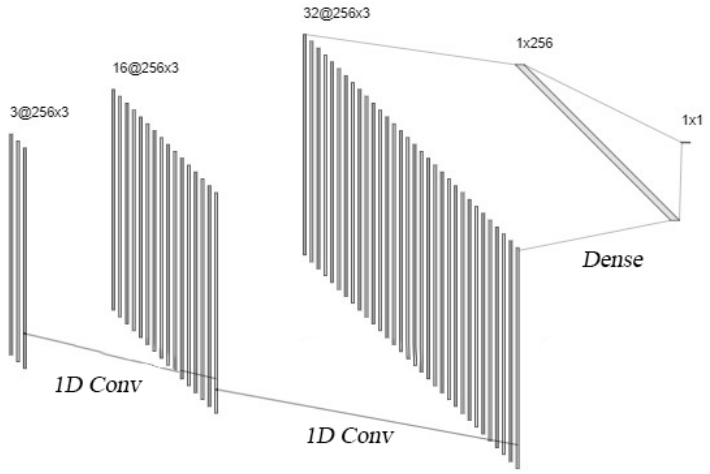

Figure 10: Architecture of CNN Two

The architecture of the second CNN follows a similar theme; however, instead of 2D convolution, this network utilizes 1D convolution due to the lower number of dimensions for the input vector. To summarize, one CNN takes a matrix input of size $[200 \times 200 \times 3]$, which is a random crop of part of the input image, while another CNN takes a matrix input of size $[256 \times 3]$ representing the color histogram. Both models output a sigmoid probability which is used to generate the final classification.

5. Model Training and Evaluation: Both models were trained utilizing TensorFlow's "fit" function. Both models also utilized binary cross-entropy loss, which is a common loss function to use when the output is binary (0 or 1 - AI-Generated or Real). It penalizes the model relative to how confident it is for making an incorrect prediction. For instance, if the model guesses the wrong label with $p = 0.55$, it is penalized less than if it guesses incorrectly with $p = 0.9$ where p is the probability of the label. The Image Crop CNN was trained for 25 epochs while the Histogram CNN was trained for 150 epochs, with both models using binary cross entropy loss defined as:

$$-(y \log(p) + (1 - y) \log(1 - p))$$

The following figure depicts this loss function for the Histogram CNN.

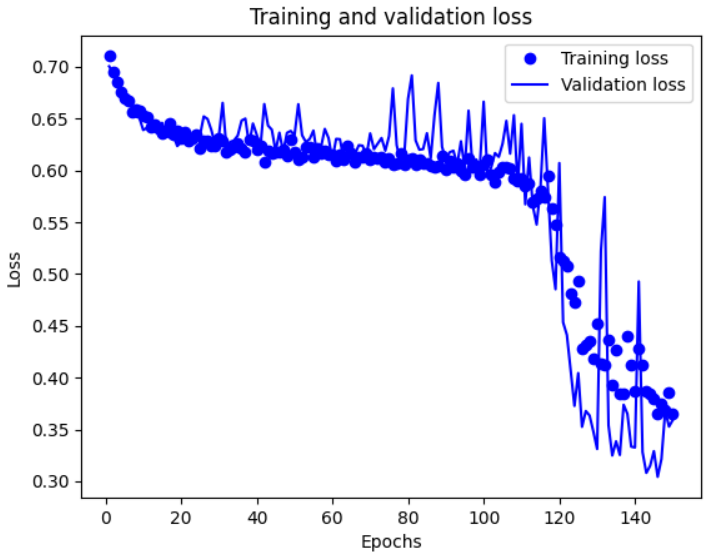

Figure 11: Training and Validation Loss of the Histogram CNN

The loss curves for the mode validate the model architecture as the training loss and validation loss trend downward and remain closely linked. Diverging loss curves imply that the model is overfitting on the training data and failing to generalize on the validation data, which may be a result of too many parameters in the architecture. Conversely, had the trend not been downward, it would have been indicative of underfitting, where the model does not have enough capacity to learn the underlying features in the data. Here, the loss curves show neither phenomenon is occurring, and the architecture of the model is relatively sound, and the trained model training would be successful in class prediction. The occasional spikes in the validation loss curve are likely created by encountering particularly difficult samples which indicates it is not generalizing well to a certain subclass of images.

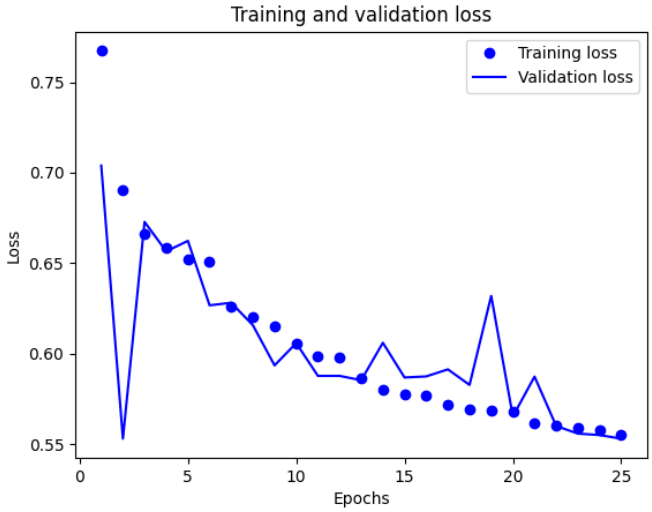

Figure 12: Training and Validation Loss of the Image Crop CNN

The loss curve of the CNN trained on random image crops paints a similar picture of neither overfitting nor underfitting, thus validating the chosen architecture.

5. Evaluation

5.1. Quantitative

In order to evaluate the success of the system quantitatively, classification accuracy is used (number of correct predictions / total number of predictions). Accuracy is chosen over alternatives like precision, F1 score, or AUC as the classes are equally distributed between the labels. In practice, the cost of a false positive is equal to the cost of a false negative. To create a functional classifier, it should recognize real images just as well as it recognizes AI-Generated ones, thus accuracy was sufficient to evaluate the models. The test set consisted of 1,000 images with equal distribution between AI-Generated images and real images. First, we analyze the accuracy of the histogram CNN. Below the figure plots the training accuracy and the validation accuracy over the 150 training epochs.

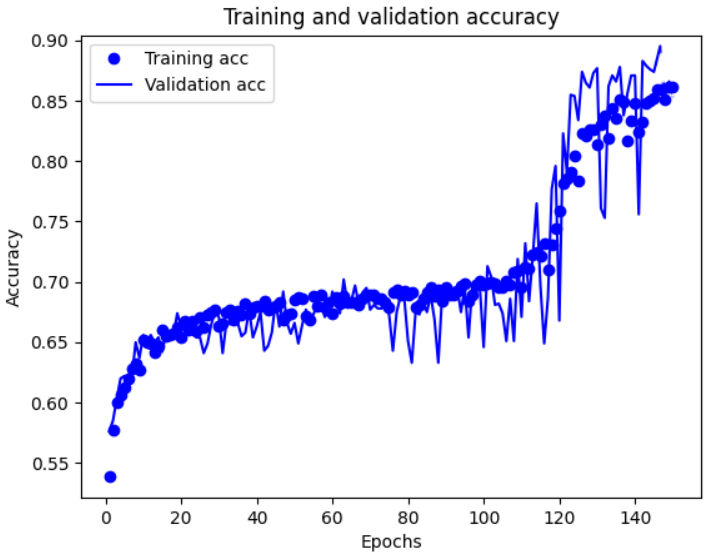

Figure 13: Training and Validation Accuracy of the Histogram CNN

The final accuracy of the Histogram trained convolutional neural network was 0.898 on the test set. While the model did not perform as well as the GAN counterparts, which often had accuracies of $> 97\%$, it does reach a similar threshold to Sha et al.'s classifier for diffusion models. The histogram CNN, however, reaches this accuracy without utilizing any information about the prompt. The high accuracy indicates this model was successful at learning the probability distributions of color between AI-generated images and real images.

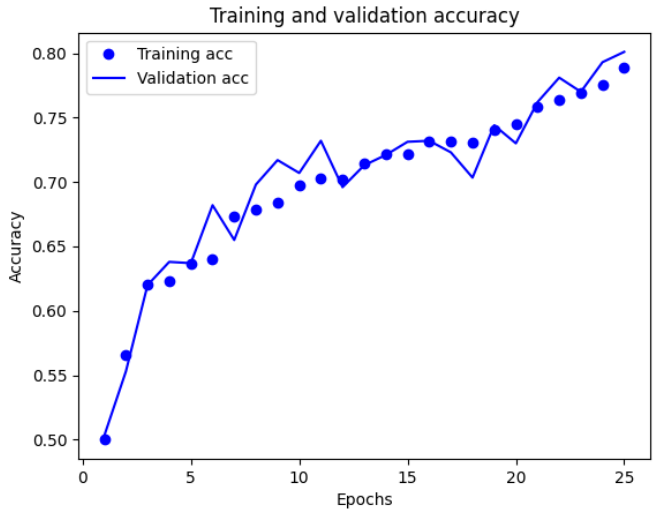

Figure 14: Training and Validation Accuracy of the Image Crop CNN

The final accuracy of the image-crop trained convolutional neural network was 0.801. This accuracy seems reasonable, however, it does not perform nearly as well as deep learning approaches on raw pixel information discussed in the prior works. While the CNN does have some predictive power, given that random guesses would have accuracy = 0.5, it cannot be practically used to detect AI-generated images as a standalone model. This model may not have performed as well as expected due to a variety of reasons, however, most likely due to it having exponentially more parameters ($\tilde{6}$,000,000) than the prior CNN, its convergence was much slower and would likely require more data to generalize properly. Interestingly, the Image Crop CNN had a higher in-class accuracy for real images which allowed the ensemble to have an increased accuracy compared to either standalone model.

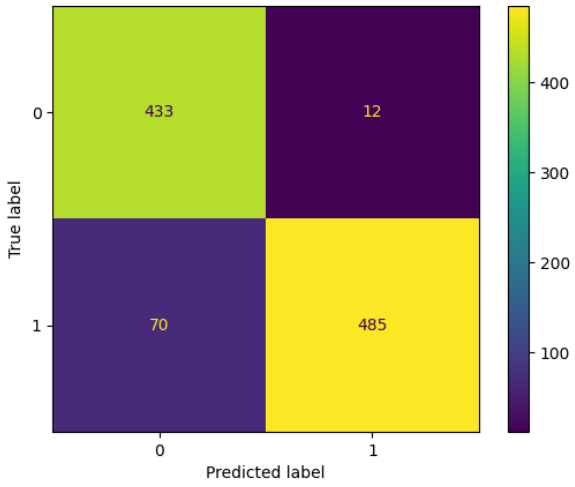

Figure 15: Confusion Matrix of Ensemble

Using a combination of the two Convolutional Neural Networks, the ensemble was able to achieve an accuracy of 91% on a test set of 1,000 images.

5.2. Qualitative

Qualitatively, one goal of the project was to understand diffusion models similar to how the exploration of GAN image detection improved the understanding of the GAN generation pipeline. The color histogram appears to be an informative feature from the quantitative analysis. Diffusion

models generate images iteratively by adding Gaussian noise to some input vector, then a model attempts to recover the input vector by learning the original probability distribution of the sample, thus creating a new image [3]. A potential explanation for the discernible histograms of these images based on their architecture is that the diffusion models are forced to learn a continuous density function (CDF) to model the data. For optimization purposes (gradient optimization), the CDF is differentiable and thus smooth and continuous. On the other hand, a camera's sensor cannot capture color intensities continuously, allowing for the jagged edges and discontinuous distribution in the color histogram.

Another important aspect of the evaluation was analyzing the cases where the model failed to predict the labels correctly. The weaknesses and limitations of the system may reveal areas of improvement and future challenges for this type of image classification.

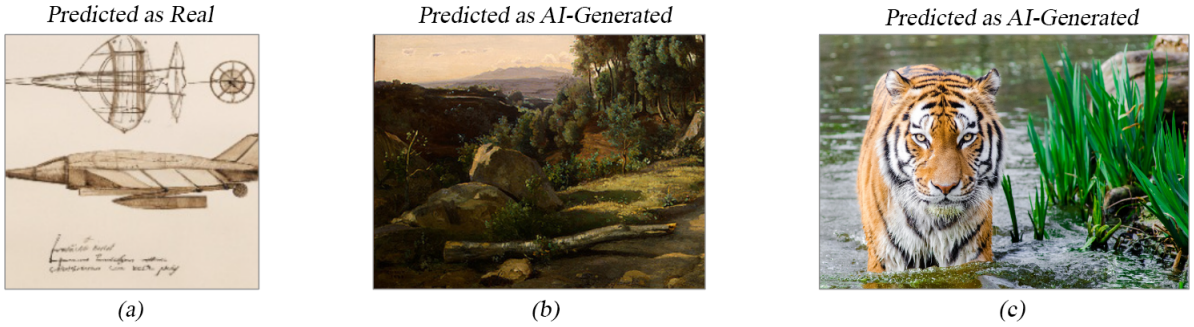

Figure 16: Misclassified Samples from Test Set

These were three misclassified images in the test set, the first being AI-Generated and predicted as real and the remaining two being real images predicted as AI-generated. The first two images represented a larger class of errors that occurred when attempting to classify digital art. When humans create digital art, they have access to a near-continuous color spectrum which may result in the histogram feature performing poorly. The right image of the tiger revealed a potential flaw in the implementation as cropping high-resolution images could result in features that capture very small portions of the original image, thus failing to capture informative information.

6. Conclusion

6.1. Effectiveness and Insights

This research project stemmed from the idea of creating a system to classify diffusion-generated images online. As these images continue to become more lifelike, social media companies, news networks, and consumers will have to evaluate every image and video in order to negate the potential harm. In regards to this original goal, the system's accuracy proves it can be used practically to detect AI-generated images. A combined classification rate of 0.91 using only images shows the potential for a widescale system to exist. While it does fall short of the baseline set by GAN image classification, this system illustrates the viability of diffusion detection and remains on par with other diffusion detection models while using less information (predicts without prompts). A secondary goal of the project was to better understand weaknesses in the diffusion model architecture via exploratory data analysis and a qualitative evaluation of the results. The spectral analysis of the dataset showed that the upgraded upsampling layer rectified many of the issues that GANs were prone to and also showed the need for new image features for detection. The predictive power of the histogram feature led to an analysis of diffusion models that revealed a potential architectural issue that may lead to distinctive differences between real images and diffusion-created images.

While the models may not be able to be deployed broadly due to a limited dataset compared to the infinite possible creations by diffusion models, the project made concrete steps towards illustrating approaches to such a system and building upon the functioning of diffusion models.

6.2. Limitations and Future Work

A key limitation highlighted in the error analysis is the ability to properly classify digitally created images. This is likely to be a broader challenge for the task of AI detection, however, the ensemble model created was particularly vulnerable due to the features it used. A large portion of the predictive power was a result of the histogram feature, which performs poorly in this sub-class.

One potential solution to implement in future work is to better the performance of the image crop CNN. A possible path toward this goal is to vastly increase the size of the training data. Convolutional neural networks which utilize raw pixel data, often have exponentially more training data. For instance, the shallow network VGG16 was trained on ImageNet which has 14 million images. Increasing the amount of training data would better the image CNN's performance and have the added benefit of making the entire ensemble more robust. Because of the limited dataset (10,000 images), this model is likely not robust to the infinite possible sample for diffusion-generated images.

For future work directly related to the implementation, one particular aspect that can be improved is the random crop. Using the random crop as an input was chosen empirically, comparing it only to a center crop and resizing the image. While these two methods performed worse to the random crop, a systematic evaluation would almost certainly reveal a better method. For instance, utilizing blob detection or sift keypoint detection to crop onto the most relevant parts of the image may yield better performance.

Another important limitation to consider, which applies to many machine learning models, is adversarial machine learning. Adversarial machine learning is a field of research surrounding understanding models and creating engineered data to trick them. Just as researchers identified issues with the upsampling pipelines with GANs, an updated version was created for diffusion models to avoid easily detectable traces of AI Generation. Similarly, it would be easy to process generated images afterward to evade the qualities that the histogram feature attempts to capture. In order to prevent against such attacks, the main solution is to make the model more robust. Accordingly, this limitation ties into the future work of creating a data augmentation pipeline which would help make the model more robust to new data samples and engineered data.

7. Acknowledgements

I would like to sincerely thank Dr. Xiaoyan Li. It was through her seminar that I was able to learn more about how to approach machine learning problems and how to overcome different pitfalls throughout the course of the project. I am also thankful to all my classmates in COS IW02 for their insights and feedback every week which helped me iteratively improve my project.

8. Honor Code

I pledge my honor that this paper represents my own work in accordance with University regulations

/s/ Arsh Banerjee

9. References

[1] Joanna M Burkhardt. "History of fake news". In: *Library Technology Reports* 53.8 (2017), pp. 5–9.

[2] Bobby Chesney and Danielle Citron. "Deep fakes: A looming challenge for privacy, democracy, and national security". In: *Calif. L. Rev.* 107 (2019), p. 1753.

[3] Florinel-Alin Croitoru et al. "Diffusion models in vision: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[4] Mudasir A Ganaie et al. "Ensemble deep learning: A review". In: *Engineering Applications of Artificial Intelligence* 115 (2022), p. 105151.

[5] Ian Goodfellow et al. "Generative adversarial nets ian". In: *Mining of Massive Datasets; Cambridge University Press: Cambridge, UK* (2014).

[6] Augustus Odena, Vincent Dumoulin, and Chris Olah. "Deconvolution and checkerboard artifacts". In: *Distill* 1.10 (2016), e3.

[7] Aditya Ramesh et al. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* (2022).

[8] Christian Rathgeb et al. *Handbook of digital face manipulation and detection: From deepfakes to morphing attacks*. Springer Nature, 2022.

[9] Zeyang Sha et al. "DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Diffusion Models". In: *arXiv preprint arXiv:2210.06998* (2022).

[10] Yaniv Taigman, Adam Polyak, and Lior Wolf. "Unsupervised cross-domain image generation". In: *arXiv preprint arXiv:1611.02200* (2016).

[11] Kai Zhang et al. "Designing a practical degradation model for deep blind image super-resolution". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4791–4800.

[12] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. "Detecting and simulating artifacts in gan fake images". In: *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2019, pp. 1–6.

10. Appendix

All of the project code is hosted on Github at

`https://github.com/arsh-banerjee/Detecting-AI-Generated-Images`.

The folder contains code to load the saved datasets as well as the saved models. The original datasets can be found at:

1. `https://www.kaggle.com/datasets/superpotato9/dalle-recognition-dataset`
2. `https://www.kaggle.com/datasets/dibyarupdutta/dmimagesubset`