CSCI 1430 Final Project Report
Evaluating Fourier Transform Features for Detection of AI-Generated Images

Sami Nourji, Everest Yang, Sujith Pakala, Tanay Subramanian
TA: Winston Li
Brown University

Abstract

This paper evaluates the addition of Fourier Transforms into CNN models designed to distinguish real images from AI-generated ones, addressing challenges posed by the rise of hyperrealistic content produced by AI. Using the CIFAKE dataset, we implement a CNN architecture with Fourier Transform features to evaluate the model's success in classifying synthetic images. Our hypothesis is that incorporating frequency information via Fourier Transforms, in addition to spatial domain information, into a CNN can enhance the detection of AI-generated images by leveraging subtle frequency inconsistencies. This was ultimately disproved by our research, as our best-performing baseline CNN achieved a testing accuracy of 98.58%, while our Fourier-based model reached an accuracy of 98.50%. Our findings highlight that incorporating Fourier features into the detection pipeline provides valuable insights, although the overall accuracy depends mostly on the CNN architecture. This research aims to encourage future research in the growing field concerning digital authenticity.

1. Introduction

Misinformation and privacy are pressing concerns in today's modern world. As chatbots and generative AI become more sophisticated, such technologies can create hyperrealistic fake images that are nearly impossible for an individual to discern. Misusing these innovative technologies has significant implications for politics, social trust, and even individual security. For example, AI-generated images have already been involved in election interference, celebrity impersonations, and malicious pranks, underscoring the importance of a model that reliably detects fake images.

Consequently, our project becomes essential for verifying digital content's authenticity as realistic synthetic images can now be generated in seconds. However, the central challenge to solving this problem is that AI-generated content can replicate minute details such as lighting, shadows, and texture with high precision, making conventional detection methods less effective. Furthermore, it is not feasible to manually label AI-generated content at scale, highlighting the importance of automated tools in detecting artificial images.

We became familiar with these types of images through both social media and exploratory generation with tools such as Stable Diffusion. After lengthy discussions on the subject, we noticed that AI-generated images display textures that appear smoother than 'real images.' To test this theory, we decided to focus on AI-Generated image detectors, and introduce frequency domain information into the models through the Fourier Transform. This paper proposes a novel AI-Generated Image Detector to determine whether adding frequency domain information would enhance the model's ability to discern real images from AI-generated ones. Studies have shown that AI-generated images have unique characteristics - such as specific frequency patterns, smooth texture, and artifacts - which distinguish them from real images. By applying Fourier Transforms, we can quantify these differences in the frequency domain where smoothness and periodic artifacts are more evident. This approach aims to enhance the model's robustness in detecting artificial images.

2. Related Work

The rapid growth of generative adversarial networks (GANs) has created opportunities and ethical concerns regarding the misuse of synthetic images. In recent academia, the CIFAKE dataset has become standard in distinguishing AI-generated images from real photographs.

This dataset was created by generating synthetic images using latent diffusion to mirror the ten classes of the CIFAR-10 dataset [2]. The synthetic dataset was paired with real images. Using a CNN, the study achieved 92.98% accuracy across 36 network topologies. Explainable AI techniques, using Gradient Class Activation Mapping, shows that the model focuses on small imperfections in the background instead of the main object [1]. This dataset has been used for many AI-detection studies, notably the 'Harnessing Machine

Learning for Discerning AI-Generated Synthetic Images' paper by Wang et al. (2024) achieving a 97.74% accuracy [4], as well as research by Lađević et al. (2024) using lightweight CNNs [3]. The latter approach yielded similarly high results as the first paper, while only using four convolutional and two hidden layers to identify AI-generated images. This architecture was also tested on benchmark datasets and Sentinel-2 images, and outperforms four state-of-the-art methods with its lightweight architecture.

Our main contribution is the addition of Fourier Transform information to CNNs to enhance efficiency in image classification tasks. Based on their work, we integrated a Fourier Transform into our model to improve spatial-frequency feature extraction [5]. By leveraging the Fourier Transform, Zak et al. (2023) accelerate training times by up to 71% with greater accuracy and reduced computational complexity.

We used these studies as a foundation for our research, focusing on the integration of Fourier Transforms into CNNs to enhance the detection of AI-generated images.

3. Method

We leveraged the CIFAKE dataset which contained 120,000 total images - 60,000 images are AI-generated and the other half are real [1]. We decided to use this dataset because it was free, publicly available, and easy to access via Kaggle. Furthermore, it seems to be used by many researchers in several research papers we read, supporting its credibility in the field. We used 100,000 images to train the CNN and the remaining 20,000 images were used for testing. Due to our laptops' hardware limitations, we leveraged Brown's remote GPU cluster, OSCAR, to train our model, especially given the sheer amount of data we were feeding it.

Since we were working on CNN architectures and keeping track of a lightweight image dataset, we decided to repurpose the Homework 5 code for this project, allowing us to work in a familiar environment and focus on training rather than setup.

To test our hypothesis, Fourier Transforms contain frequency information about AI-generated images which could be leveraged to improve classification tasks, we started by reimplementing the architecture outlined in the paper with the highest CIFAKE accuracy, as depicted above (Figure 1). The paper had details about the convolutional blocks and the type of layers used, but their details on the fully connected head were lacking. As a result, we had to perform many tests to identify the best hyperparameters such as learning rate (1e-3), number of epochs (50), and the dropout rate (0.3).

This paper served as a starting point and baseline for our testing, so we decided not to change the hyperparameter values after our initial conclusions. We also did not perform any data augmentation (such as rotation, scaling, etc.) and

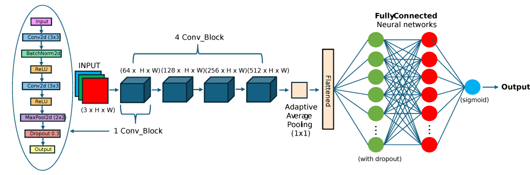

Figure 1: Original paper model architecture

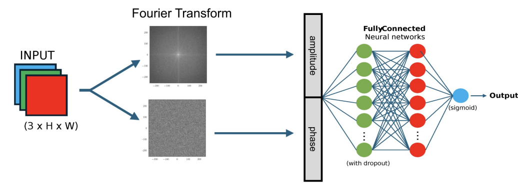

Figure 2: Fourier Transform only model architecture

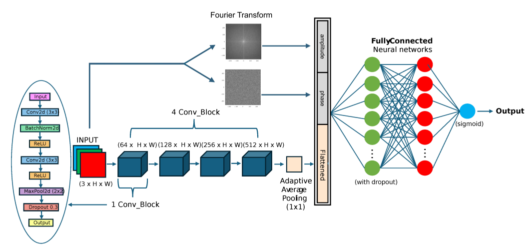

Figure 3: Original CNN with concatenated frequency data model architecture

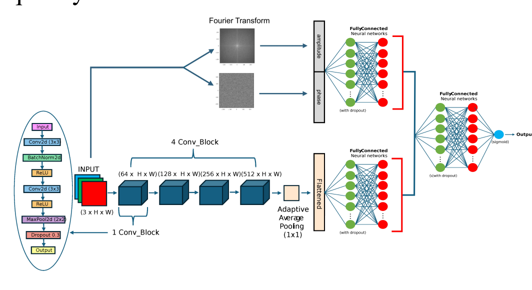

Figure 4: Combined parallel architectures (Original CNN and fully connected Fourier Transform) model architecture

we preserved the paper's lightweight architecture style by keeping all subsequent models under 10 million parameters. Furthermore, we didn't apply any pre-processing techniques to the original image dataset, since the model architecture already seemed to handle the raw data effectively.

We then made our first major architecture modification which aimed to establish whether the Fourier Transform of input images, i.e. their representation in the frequency domain, contained any information that could help differentiate between real and AI-generated images. To test this, we used a simple neural network that took an image, performed a Fourier Transform to extract its amplitude and phase, and ran it through many dense layers to output a classification (c.f. Figure 2).

After obtaining positive results from this exploration, we conducted our first experiment to evaluate the effect of adding this frequency information to the original CNN architecture we recreated from the paper. As seen in Figure 3, we determine the amplitude and phase of the input image, and concatenated it to the flattened output of the convolutional layer. This new vector is then passed through a dense neural network, as was case with the convolutional layer output only in the original paper architecture. We performed an additional baseline test where random noise, instead of the Fourier Transform information, was concatenated with the output of the convolutional layer. This experiment was intentionally designed to add the Fourier information aggressively into the original CNN architecture, as we wanted the dense layer to process both spatial and frequency information at the same time.

Following this experiment, we had a solid idea of the predictive power of the frequency domain of an image. However, we were unsure whether this information was complementary to the spatial information, or if it was a more noisy representation of the data with no added value. To investigate this, we introduced our second experiment, which can be thought of as a combination of the first experiment and the original architecture. As can be seen in Figure 4, we ran the Fourier Transform neural network and the CNN architecture in parallel, and then combined the outputs of both neural networks (respectively size 256 and 128) into a final fully connected neural network. Conceptually, this would enable the "combined" neural network to determine the value of both outputs and decide which one to use, without having to process both the spatial and frequency data at the same time (which was the case in the first experiment). We also conducted a baseline of this experiment using random noise instead of the Fourier, and this enabled us to deduce whether this "combining" network completely excluded the Fourier output, or if it used it to make better predictions.

These experiments were carefully designed to test our architecture, and we detail our findings in the subsequent section.

4. Results

The results of the experiments outlined in the previous sections are summarized above (Table 1); the following is a more detailed explanation and interpretation of these accuracies.

Our reimplementation of the CNN model (c.f. Figure 1) yielded a test accuracy of 98.58%. This performance provided a baseline for comparison with the results of our experiments that incorporated Fourier Transform information.

The second baseline we established was with a model architecture using only the Fourier Transform information (c.f. Figure 2) which resulted in a test accuracy of 82.53%.

Model	**Test Accuracy**
Original CNN model (Figure 1)	98.58%
Fourier Transform only (Figure 2)	82.53%
Concatenated Fourier (Figure 3)	
Baseline (Random Noise)	50.35%
Fourier Transform	95.36%
Combined Parallel Architectures (Figure 4)	
Baseline (Random Noise)	98.50%
Fourier Transform	98.50%

Table 1: Model Performance Comparison

This moderate accuracy tells us that Fourier Transform information independently holds value in classifying between real and AI-generated images as it was significantly above a random or 50% accuracy. However, independently, it is less accurate than using convolutional layers on the original images which had an accuracy of 98.58%. This was expected given the superiority of CNN architectures when dealing with image data.

As discussed in the aforementioned "Method" section, our first experiment combined the Fourier Transform information with the output from the convolutional layers (c.f. Figure 3). Our baseline conditions, which appended random noise to the convolutional output vector, performed with a near random classification accuracy of 50.35%. However, our experimental conditions, which appended the Fourier Transform vector, resulted in an accuracy of 95.36%. While the addition of both types of information, noise and Fourier Transform, reduced the accuracy of the model from the original CNN model, the noise essentially rendered the model useless while the Fourier Transform information only reduced accuracy slightly. These results suggest that the Fourier Transform information, although not able to improve the accuracy, did contain predictive information.

These results suggest that frequency information has predictive potential for AI-generated image detection, but we wanted to determine whether it provided added value to the CNN architecture. Our second experiment combined both Fourier and CNN architectures into one model (c.f. Figure 4). The results of both and random noise yielded the same accuracy as the very first reimplemented CNN architecture - 98.5%. This lack of a difference in accuracy between the baseline and experimental group is different from the significant difference observed in the concatenation experiment (c.f. Figure 3). This suggests that when the added information is directly appended into the same fully connected layers as the convolutional layer, the model uses all of the information provided, causing there to be a large accuracy distinction in experiment 1 between the random noise and Fourier Tranform inputs. However in experiment 2, when the model was given these outputs separately, it was able to choose the information to use to predict classification. Given that the accuracies for the baseline and the experimental group were both 98.50%, it's clear the model ignored the noise and Fourier Transform information, and only used the convolutional layer information regardless.

4.1. Technical Discussion

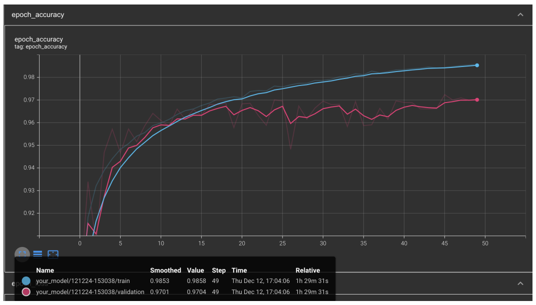

Figure 5: Training and validation accuracy of the original CNN model

The focus of our work for this project was around designing simple and insightful experiments. To do so, we performed many tests to determine architectural features, while also keeping the number of parameters low.

A notable example of this is our varying fully connected neural network sizes. The initial CNN head was considered as a baseline, with perceptron numbers going from 256 to 1 with steps of ½. The Fourier-only fully connected network saw an additional 512 dense layer being added to ensure meaningful patterns were extracted from the magnitude and phase features of input images. This decision was also motivated by the decrease in number of parameters, which allowed us to increase the size of the head.

The most interesting experiments we undertook were for our "combining" architecture (c.f. Figure 4). A big challenge we had was determining the size of the parallel dense layer outputs. Given that we wanted to append them together and pass them through a third fully connected network, it was important to consider their respective sizes and proportions. We initially noticed a poor accuracy with a model design that only took in a single prediction from each parallel architecture. Given that this "decision forest" style architecture provided poor results, we turned to larger output sizes to allow the combining network to extract more information from each architecture. This provided better results, and after experimenting with many splits, the 1:2 ratio between Fourier output and CNN output seemed to work best. However, given that this architectural design was empirically determined and was inspired by conceptual intuition, we are unsure if a "combining" neural network was the best option. Perhaps, other types of layers or blocks would've been more effective (e.g. introducing an attention layer).

Another interesting aspect of the original CNN architecture was that we didn't succomb to overfitting, despite strong performances often exceeding 95%. For example, the gap between training and validation accuracy was very small, as can be seen in Figure 5 for the original CNN accuracy.

Using the CIFAKE dataset, we also faced constraints in hyperparameter optimization due to limited GPU resources, despite leveraging Brown's OSCAR GPUs.

5. Conclusion

In this paper, we created novel model architectures that evaluated the addition of frequency information into CNNs designed to distinguish real images from AI-generated ones. Our experiments demonstrated that Fourier Transforms contain information about whether an image is AI-generated. However, we found that incorporating frequency features into CNN models did not improve results nor provide any added value.

A significant limitation of our project is that while the CIFAKE dataset is relatively new, it does not account for the significant advances in generative AI made in the last year. Hence, while this dataset allowed for efficient training and testing, it does not represent the variety or complexity of real-world images, limiting the generalizability of our findings. Future research should curate datasets containing images with higher resolutions and more diverse content to ensure better performance in real world applications.

References

[1] J. J. Bird and A. Lotfi. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. *IEEE Access*, 2024. 1, 2

[2] A. Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. 2009. 1

[3] Lokner Lađević, Tomaž Kramberger, Rok Kramberger, and Danijela Vlahek. Detection of AI-Generated Synthetic Images with a Lightweight CNN. *AI*, 5(3):1575–1593, 2024. 2

[4] Yong Wang, Yifan Hao, and Ai Xia Cong. Harnessing Machine Learning for Discerning AI-Generated Synthetic Images. *arXiv preprint*, arXiv:2401.07358, 2024. 2

[5] Jakub Zak, Anna Korzynska, Antonina Pater, and Lukasz Roszkowiak. Fourier Transform Layer: A Proof of Work in Different Training Scenarios. *Applied Soft Computing*, 145:110607, September 2023. 2

Appendix

Presentation

For a detailed presentation of our research, please refer to the following link: Research Presentation

Team contributions

Everest Yang: I developed the code for integrating the Fourier Transform into our model, including the implementation for concatenating the flattened vector dimensions. Additionally, I conducted an extensive review of related works and research papers to identify optimal CNN architectures. I also contributed significantly to the writing of the paper, specifically on the Introduction, Related Works, Technical Discussion, Conclusion, and key parts of the Method section. Furthermore, I formatted the entire document into a comprehensible research paper format using LaTeX and a .bib file for our references.

Tanay Subramanian: I was responsible for finding literature concerning current research concerning neural networks and the integration of Fourier transforms into classification tasks. Additionally, I helped develop the code for the baseline model architecture, in addition to working on the final paper, contributing to the Abstract, Introduction, Methods, and Conclusion sections. I also worked extensively on the final presentation slides, helping summarize our research in a concise and visually appealing manner.

Sujith Pakala: Once we realized that our project would require more computing power than our computers or Google Colab would allow, I took on the role of understanding and debugging our set-up with Oscar to be able to test our models in a time efficient manner. I also spearheaded the development of the code for our baseline without the fourier transform and the code for just using the fourier transformation after investigating architectures used historically in the literature. I also worked with Sami to design our "experiments" and created the graphs used in the report.

Sami Nourji: I took on the role of Project Manager for this final project, helping develop the project idea, and managing the allocation of tasks among teammates. My work within the project involved developing the Fourier experiments, model architectures, and training the model. I also worked on interpreting the model results and writing the method, technical discussion, and conclusion sections of the report.