Using Log Scaled Fourier Transformations for Deepfake Detection.

Sidharth Rangarajan

Monta Vista High School

Abstract

Deepfakes, synthetic pieces of media, have taken over our online ecosystem in recent years, posing a significant threat to our digital security. This paper discusses the idea of implementing a log-scaled Fourier transformation for deepfake detection. By converting images to the frequency domain, Fourier transformations help identify deepfakes through specific frequency patterns. Experimental results show noticeable differences between transformed deepfakes and real images. Experiments demonstrated high model accuracy for models trained and tested on frequency images. In conclusion, these results show potential for frequency analysis at the forefront of deepfake detection.

Introduction

In the past year, North America had a 1074% increase in deepfake-related fraud (Abdullah et al., 2024). This is no simple tech trend, but a rising concern. Unlike other forms of fraud, deepfake-related cases only improve over time. Traditional verification methods fail to catch sophisticated deepfakes. For decades, we have analyzed images spatially. Modern-day convolutional neural networks (CNNs) are built on the principle of spatial domain analysis, established decades prior. With improvements in our analysis techniques, it is only natural to improve pre-processing accordingly. In the 19th century, French mathematician Joseph Fourier introduced what is now known as the Fourier Transformation in his memoir on heat conduction, submitted to the Institute of France. Fourier's work demonstrated that complex functions could be represented as sums of sinusoidal components. As time passed, mathematical functions originally

introduced by Fourier were extended to signal processing. This involved audio analysis, where signals are broken down into frequency components characterized by amplitude and phase. 2 centuries later, the well-accepted audio technique was no longer uniquely functional. James Cooley and John Tukey, respected mathematicians, designed the FFT or Fast Fourier Transform (Diagram of FFT, Figure 1). They took the revolutionary idea of the 19th century and integrated it for image analysis. With the help of their countless predecessors, they discovered a way to analyze images with a Fourier transformation that was computationally possible (Fisher et al., 2003). Their accomplishments were well respected. Fourier transformations were no longer a simple point of hypothesis; they were practically experimented with. Yet, despite the achievements of scientists like Fourier and Cooley, we continue to overlook this powerful, underexplored analysis technique. This study aims to leverage the research and accomplishments of Fourier and Cooley and apply them to real-world problems. This paper hypothesizes a novel way to detect deepfakes using the Fast Fourier transform.

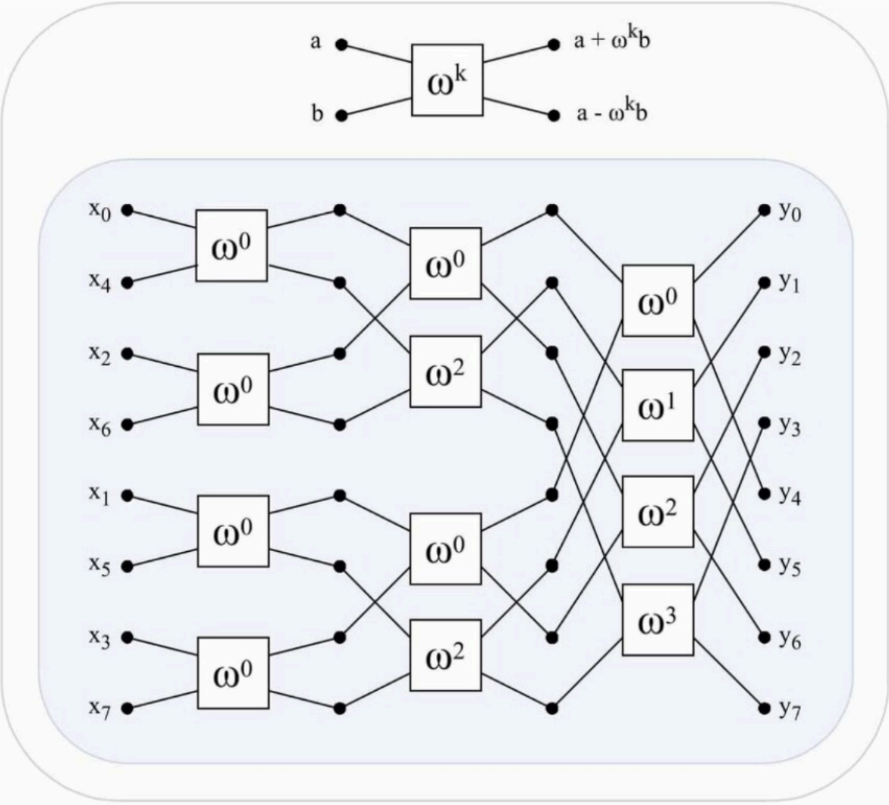

FIGURE 1. Diagram of the Cooley-Tukey FFT algorithm (Ding, 2024). The diagram labels the input and output with the transformations that take place. This specific transformation has 3 layers.

Background

Deepfake detection is no new problem. In a paper published in the MDPI journal, author Ali Raza proposed a "DFP" approach to detecting deepfakes. Raza's idea was to add a VGG16 layer along with standard CNN layers to create a more adept model (Raza's model architecture by layer, Figure 2). Raza kept the standard input and developed model layers, adding complexity to capture subtle details in deepfake images. Raza, along with several of his predecessors, continues to optimize the layers of modern models, without considering the input itself. Continuously developing only the model is not only short-sighted but also leads to homogeneous results. In a separate paper, published by IEEE, author Yogesh Patel proposed a D-CNN. The idea was to add layers to a standard CNN to create a more attuned deep CNN. Patel's model architecture (Patel's model architecture by layer, Figure 3) implemented additional layers with a binary classification. Beyond just spatial-domain methods, there exist bodies of work that have examined deepfakes using frequency domain analysis. Durall et al. (2020) showed how deepfake images generated by common CNN models often fail to produce the natural spectral distributions of a real image. Their findings revealed how generated images carry detectable frequency domain artifacts. Moreover, several other researchers have proposed detection pipelines that involve frequency domain analysis, such as the Discrete Cosine Transform (DCT). But despite these developments, this field has largely focused on what happens inside the model rather than what is fed to the model in the first place. Both spatial and frequency approaches tend to accept the input for what it is and focus more on optimizing downstream processes. This paper explores the input layer as an independent variable in the detection pipeline. By incorporating frequency domain analysis into a high-capacity classification model, this paper focuses on the input images specifically. By doing so, this approach builds upon both spatial and frequency studies by analyzing the effect of an FFT on the inputs of a model.

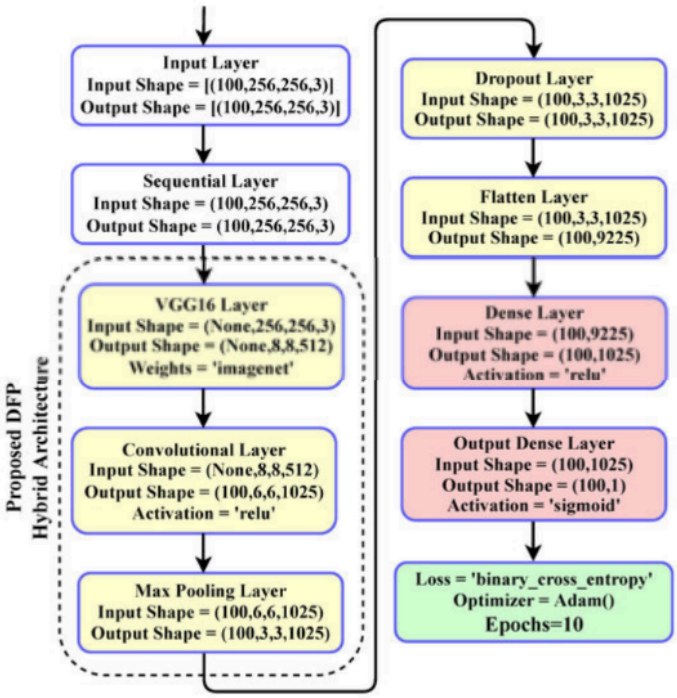

FIGURE 2. Ali Raza's model architecture. Raza implements an additional VGG16 Layer to a standard CNN (Circled). VGG16 uses ImageNet weights and takes a 256x256x3 input.

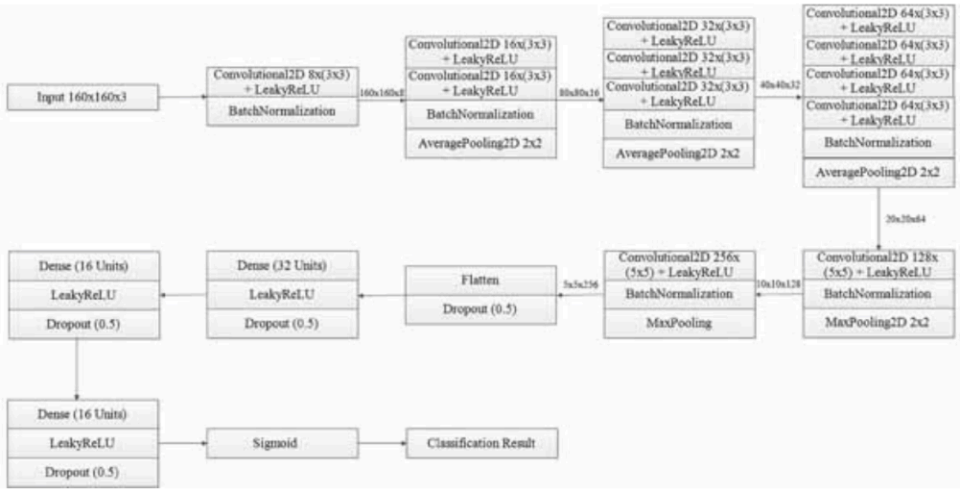

FIGURE 3. Yogesh Patel's model architecture. Patel implements several additional layers to a standard CNN. These layers include several convolutions and occasional dropouts.

Materials and Methods

In the process of creating a model, I conducted a preliminary experiment on FFTs themselves. To evaluate FFT performance, I applied the transformation to two distinct images. The first was artificially generated on the dark website Only Fakes. Only Fakes was at the center of deepfake fraud in 2024, reportedly creating 20,000 deepfake IDs daily, bypassing verification attempts across thousands of platforms (Chang 2024). As a control, I picked a standard California-issued ID. Applying the FFT to both IDs saw similar results. Although deterred, I further experimented with the transformation and decided to apply a log scale technique on top of the transformations. The result was two completely distinct images that no longer seemed similar. Log transformations function by compressing the range of frequencies in an image. In raw Fourier-transformed images, high frequencies dominate the images and drown out weaker, more subtle frequencies. When a log transformation is applied, the contrast between high and low frequencies is reduced, making subtle frequencies more visible. To analyze the effects of log transformations mathematically, it is important to consider how power spectrums work in standard FFTs. Power spectrums are the magnitude or strength of each component of a frequency image (Fisher et al., 2003). The formula for this magnitude is defined as follows, where F(u,v) is the frequency representation of the frequency component at coordinates (u,v).

$$P(u, v) = |F(u,v)|2 = Re[F(u,v)]2 + Im[F(u, v)]2$$

(Equation 1)

The standard power spectrum has a wide spectrum range and can span several orders of magnitude. To solve this problem, a log scaling technique was implemented. The idea is to compress the range of values by applying a log function to the power spectrum (Fisher et al., 2003). The new formula can be seen below.

$$S(u, v) = log(1 + P(u, v))$$

(Equation 2)

After implementing this formula and reapplying the transformation on both images, clear differences started to appear. A frequency-transformed version of the deepfake ID was first analyzed (128px by 128px Figure 4). It appears to be mostly black with circles of white. The bright white dot in the center is the DC component, a representation of the brightness or intensity in the original image. The gray circles/semicircles are tied to specific frequencies and periodic patterns. These circles represent the repeating grids or textures that are present in deepfake images. To the right of this image is a

frequency version of an authentic ID (128px by 128px Figure 5). Issued by the state of California, this ID was picked as a control to compare with its deepfake counterpart. In the image, there is once again a bright white dot, or DC component. This similar detail in both images validates the control used in this experiment, which had similar external factors(brightness) as the deepfake image. In the control image, several white lines run across a gray pixelated gradient. Breaking down the image, the white lines represent strong low-frequency components, which are likely from the structure and layout of the ID. These white lines signify that the original image had a grid-like, organized layout. The cross-like figure in the middle suggests that the image had naturally balanced features, like those in authentic documents. The grayish hue of the image indicates that the original picture had more low-frequency components(details) than high-frequency ones, which fits a natural document. The spread of the gray dots throughout the image suggests these details are natural, not artificially added. After formulating the FFT, I applied it to the data from a Kaggle dataset called CIFAKE. CIFAKE contains real images from the CIFAR-10 dataset along with their AI equivalent, made with Stable Diffusion, a high-tech generative AI. This dataset was already split into a train and test (85% to training and 15% to testing), and did not require any further refinement. I passed on the frequency data to a ResNet-50 model from Tensorflow. ResNet-50 is a high-class image classification model, which was repurposed and retrained for deepfake detection in this experiment. ResNet-50 differs from standard models through its use of residual layers. Unlike typical models, residual layers pass the original input along with the transformed output. This can help preserve details in the original frequency image that could be lost along the convolutions applied to the image. To customize the ResNet-50 model for this specific deepfake analysis, I unfroze the final 25 layers of the model for fine-tuning and left all other layers frozen.

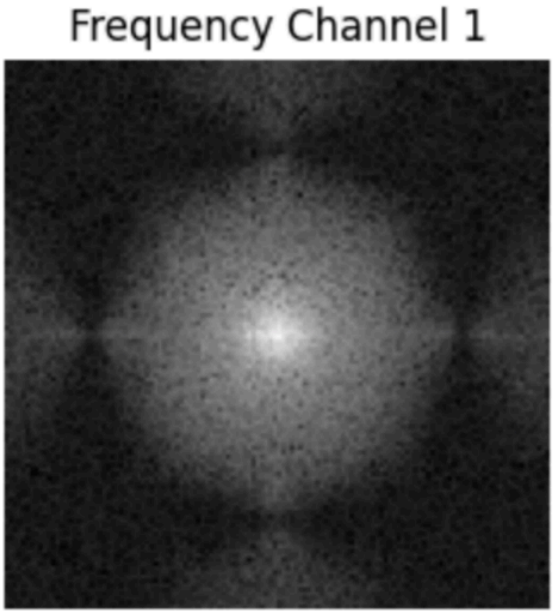

FIGURE 4. Fourier image of an Only Fake ID. The image shows a bright component in the center.

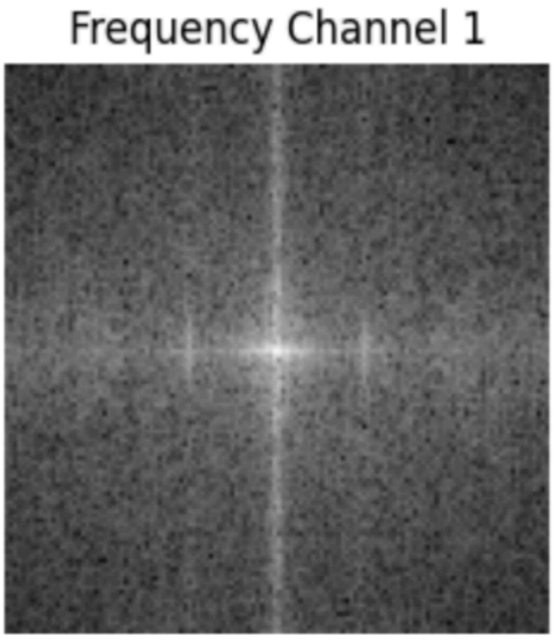

FIGURE 5. Fourier image of a real ID. The image has a similar DC component to Figure 4. There are white lines across the image, which are similar to the low-frequency components in real images.

Results and Discussion

I concluded this study on Fourier transformations by evaluating and fine-tuning model performance. The model reported a mean accuracy of 77.98% ± 4.5% after 10 trials on 40 epochs. For comparison, a ResNet-50 model trained on the same dataset using spatial-domain analysis reported an accuracy of 39.92% (Average of 10 trials, 40 epochs each). Because the input images had to be resized to relatively small dimensions, some spatial features may have been degraded. In addition, a lack of computer resources prevented extensive hyperparameter testing. Both limitations lead the standard spatial model to underperform random chance. Even so, the performance gap between the models does not accurately reflect the potential of FFTs. To properly train a model, several constraints had to be made. When the FFTs were applied to every image in full, the model required considerable computational power. The required processing power was too high, even for GPUs on Amazon's EC2. Due to limited computational power, images had to be preprocessed at smaller sizes (smaller than figures 4 and 5), blending important details. In addition, the excessive computational requirements detracted from the log-scaling technique applied. Table 1 (model accuracy) shows the accuracy of models trained on images of different sizes. The 128px-by-128px image was too large for the model to process; it showed promise, as present in figures 4 and 5. At this size, with the proper scaling technique applied, it has not yet been tested; however, the results shown here provide incentive for further research. Although the best possible model couldn't be tested, a worse version still achieved almost 80% accuracy.

Image Size	Could Train Model	Accuracy
128px by 128px	No	--
64px by 64px	Yes	77.98%
32px by 32px	Yes	40.59%

TABLE 1. Accuracy of the model trained on images of different sizes. The table above shows the accuracy (if available) of a model trained and tested on frequency images of different sizes (px stands for pixels).

To achieve this accuracy, several trial-and-error hyperparameters were tested. One of the main hyperparameters was epochs. Epochs refer to the

number of times your model is fully trained on a set of data. For example, 2 epochs would translate to 2 full runs on the training data. In the table below (accuracy per epoch Table 2), several different epoch amounts are listed with their corresponding accuracy. After 40 epochs, the accuracy curbs and begins to decline. Running many times on the same data, the model begins to draw patterns that aren't present in other data. To prevent this, it's important not to overtrain while also training enough to increase accuracy.

Epochs	**Accuracy**
5	60%
10	67%
20	75%
40	77.98%
50	77.97%

TABLE 2. Accuracy after epochs. The table above shows the average accuracy over 10 trials for the model after a certain number of epochs. Average accuracy rises to 40 epochs and then decreases.

To explain the accuracy in depth, out of 20,000 test images, the model predicted 8754 true positives, 1246 false negatives, 3158 false positives, and 6842 true negatives (confusion matrix, Figure 6). Moreover, the precision, recall, and F1-score for fake images are 0.8459, 0.6842, and 0.756, respectively. At the same time, the precision, recall, and F1-score for real images are 0.7347, 0.8754, and 0.799, respectively. Overall, these numbers signify the model's strength in classifying true positives, but also its weakness in classifying false positives.

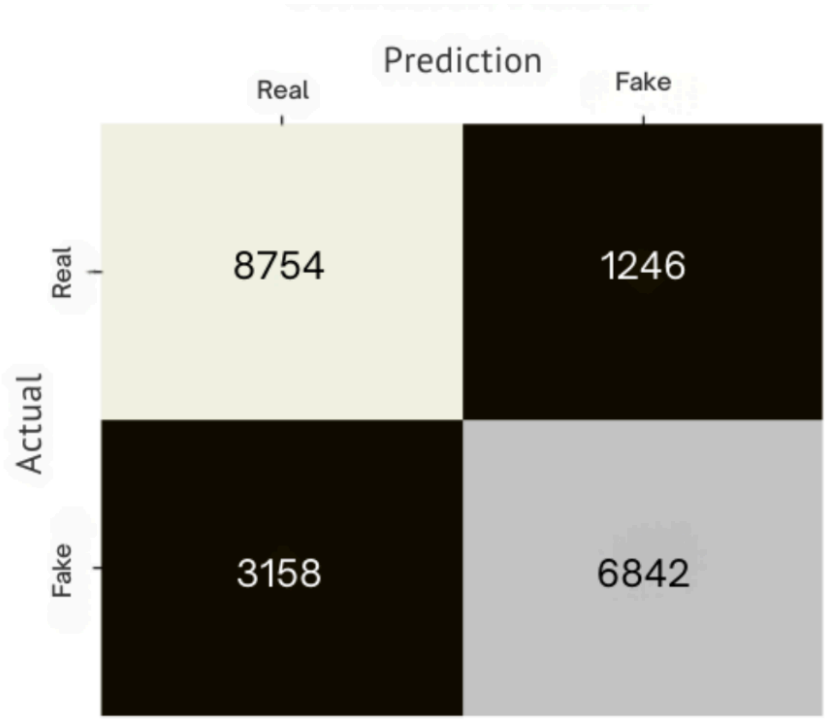

FIGURE 6. Confusion matrix of the model. This confusion matrix depicts the false and true positives, as well as the false and true negatives, that the model produced on the test dataset.

Putting the numbers aside, there were some weaknesses in the model training. For certain images, the characteristics of the real images were more akin to fake images. In the future, skewing response to real over fake could be beneficial (ex: changing class weights in the loss function). This is demonstrated in a frequency version of a real image from the train dataset (128 px by 128 px Figure 7). In the image, there are circle-like figures that are similar to those in the fake images. Although a cross pattern is shown (a characteristic of real images), there is a lack of gray hue, and the overall image seems fake. For each image, the FFT was applied to an already gray-scale image. The title "Frequency Channel 1", present in Figure 7, refers to the only frequency channel after the image was gray-scaled. If the image had not been gray-scaled, there would be 3 channels, for each RGB color.

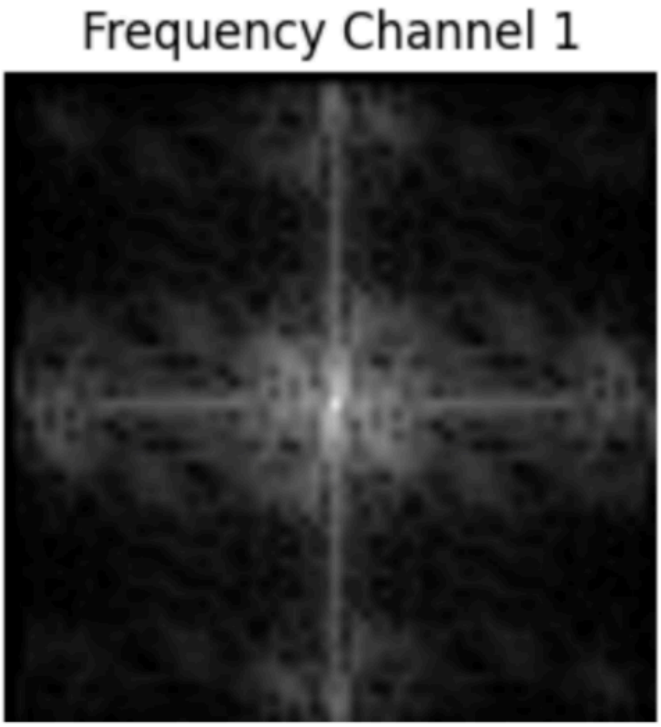

FIGURE 7. Fourier image of a real image for the train dataset. The image shows features of fake images in circular-like figures. The figure has overall elements of real images with a grid-like shape

To further catch errors, data augmentation was implemented with the model. Data augmentation is an approach where data is modified, through orientation or color spectrum, to create a variety of data. This variety can help train a model that can accurately classify all kinds of images. It further helps root out rudimentary problems within the model. For this experiment, data augmentation included random horizontal flips, random brightness adjustments(±0.2), and random contrast scaling (range of 0.8-1.2). Moreover, I implemented an AdamW configuration with a learning rate of 1x10^-4 to better position the model (Loshchilov and Hutter, 2019). AdamW is a model configuration that separates the decaying of weights from learning rate regulation. This can help decrease weights every cycle and reduce the reliance on certain weights(features). The table below (validation accuracy through epoch checkpoints of different models, Table 3) depicts the different results of the model, using validation accuracy as a determining factor. Validation accuracy is the accuracy of the model on a validation set. This validation set is randomly split from the train dataset at the start of each epoch and is used to monitor model performance during training. In my experiment, there was an 80 to 20 train/validation ratio. As the data demonstrates, leveraging both the data augmentation and the AdamW model greatly helped increase

validation accuracy as the model trained. Although both the validation accuracy and actual accuracy of the final model were not considerably high, the process and research of Fourier transformations show potential for deepfake detection and could be an area for extended research.

Models / Epochs	Base	Data Augmentation	Data Augmentation and AdamW
5	35%	40%	42%
10	50%	57%	65%
40	65%	75%	80.29%

TABLE 3. Validation accuracy of models at epoch checkpoints. The table above displays the validation accuracy for different models after 5, 10, and 40 epochs.

Conclusion

This paper explored the idea of an altered preprocessing layer for deepfake detection. FFTs were applied to the input images to highlight image inaccuracies or irregularities. After several preliminary experiments, the transformations proved useful in visually differentiating frequency patterns between real and AI-generated images. Although limitations were made, applying the transformation to a model saw notable results. Achieving an almost 80% accuracy and improving from epoch to epoch shows promise for Fourier transformations in the future. The results may not have been extraordinary, but they shine a light on a new form of AI development. The idea of frequency analysis for deepfake detection represents a promising direction in the field of machine learning.

References

Abdullah, S. M., Cheruvu, A., Kanchi, S., Chung, T., Gao, P., Jadliwala, M., & Viswanath, B. (2024). An analysis of recent advances in deepfake image detection in an evolving threat landscape. *arXiv preprint arXiv:2404.16212*. https://doi.org/10.48550/arXiv.2404.16212

Bird, J. J. (2023, March 28). CIFAKE: Real and AI-generated synthetic images [Data set]. Kaggle. https://www.kaggle.com/datasets/joshua-bird/cifake-real-and-ai-generated-syntheticimages

Chang, W. (2023, August 13). Council post: AI is the final blow for an ID system whose time has passed. *Forbes*.

https://www.forbes.com/sites/forbestechcouncil/2023/08/13/ai-is-the-final-blowfor-an-id-system-whose-time-has-passed/

Ding, L. (2024). Diagram of the Cooley-Tukey algorithm performing the classical FFT on an input x[Rn]. *A novel approach for efficient fast Fourier transform computation* [Figure 4]. ResearchGate. https://www.researchgate.net/figure/Diagram-of-the-Cooley-Tukey-algorithm-23-performing-the-classical-FFT-on-an-input-xRn_fig4_380076622

Durall et al. (2020): Cited in the Background section but has no corresponding reference entry. Full citation: Durall, R., Keuper, M., & Keuper, J. (2020). Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions. In Proceedings of CVPR 2020.

Fisher, R. B., Perkins, S., Walker, A., & Wolfart, E. (2003). Image transforms – Fourier transform. *HIPR2: Hypermedia Image Processing Reference*. University of Edinburgh. https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm

Loshchilov & Hutter (2019): Cited in the Results and Discussion section (for AdamW) but missing from the reference list. Full citation: Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In Proceedings of ICLR 2019.

Patel, Y., Munir, K., Raza, A., & Almutairi, M. (2023). An improved dense CNN architecture for deepfake image detection. *IEEE Access*, 11, 22081–22095. https://doi.org/10.1109/ACCESS.2023.3245678

Raza, A., Munir, K., & Almutairi, M. (2022). A novel deep learning approach for deepfake image detection. *Applied Sciences,* 12(19), 9820. https://doi.org/10.3390/app12199820