Detection Approach of GANs-based Morphing Attacks By using Benford's Law

Juan Marquez Diaz[1], Juan Martinez[2], Yubo Zhou[3]

Department of Computer Science, California State University-Fresno

[1]juanmark21@mail.fresnostate.edu; [2]junmartinez@mail.fresnostate.edu; [3]yubozhou@mail.fresnostate.edu

Abstract— ***Throughout the years, advancements in Generative Adversarial Networks has resulted in the creation of highly realistic fake images. This has also raised concerns about vulnerabilities and the usage of exploitations in face recognition systems with GAN-generated face images. Our small team proposed the idea of using Benford's law in order to distinguish between real and fake images. By gathering various datasets, our team attempted and concluded that Benford's Law was not a viable solution.***

Keywords—Fake Image Detection, GAN, Benford's Law, DCT, JPEG

I. Introduction

With the technological advances in the following years, the usage of face recognition software and hardware has increased. However, so has the development of Generative Adversarial Networks and the generation of fake realistic face images. One prominent Generative Adversarial Network that has been in the spotlight for the past few years is StyleGAN. With the usage of automatic, unsupervised separation of attributes, and variation operations, StyleGAN can produce some realistic face images [1]. These innovative techniques can raise critical concerns about being used as an exploitation method against biometric systems. It is worthwhile to investigate more to find out effective countermeasures to overcome these detection problems. Each GAN architecture may introduce different traces, thus making the generalization of the detection a complex task, which means that a detector that has been trained to detect images generated by a specific GAN architecture could not be suitable for a different GAN scheme [2]. For this reason, this has raised concerns about how well algorithms can detect GAN-generated images. One approach taken for detecting GAN-based images is to use Benford's Law [2]. Given that Benford's Law is used as a measurement for real-life numerical datasets, we deem to investigate if Benford's Law can be applied to measuring the numerical data from the GAN-based images with the provided baseline from Benford's Law [2]. The works of Bonettini, et al. proposed the Discrete Cosine Transform (DCT) coefficient of a JPEG image as an ideal parameter for the detection of the anomaly, by comparing the extracted features from JPEG images with the Benford's Law model. In our project, we aim to specifically focus on verifying the feasibility of using the Benford's Law method proposed by Bonettini, et al. on detecting GAN-based fake human face JPEG images.

II. Related Work

When exploring our topic, we referenced Venkatesh, et al.'s work for the GAS-generated morphs threats and Bonettini, et al.'s for the Benford's Law implementation. These sources provide insights into formulating our project.

A. GAN Generated Morphs Threats

The recent improvements made in GAN architectures such as StyleGAN can successfully generate realistic facial images with high quality [3]. This is achieved by embedding the images into latent space which is further optimized to synthesize the high-quality and high-resolution image [4]. Figure 1 demonstrates the image generation process using StyleGan architecture [1][3]. The StyleGAN module reconstructs the images corresponding to the updated latent codes ($L_1^{'}$, $L_2^{'}$) and latent spaces (w_1, w_2) by utilizing two neural networks. The average latent code L_M is computed as follows:

$$L_M = \frac{w_1 * L_1^{'} + w_2 * L_2^{'}}{2} \qquad (1)$$

Finally, the average latent code obtained will be passed into the synthesis network to generate the morphed image[3].

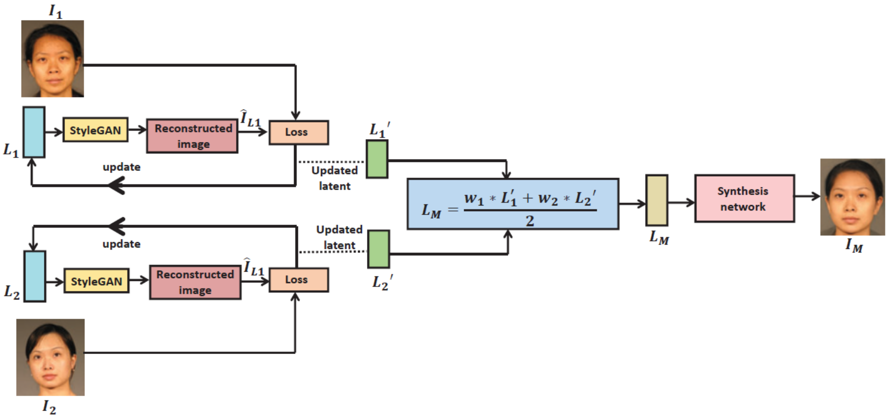


Fig. 1 Block diagram of the morphed face image using SytleGAN

The result of Venkatesh, et al.'s work indicates that the StyleGAN obtained a notable MMPMR (Morphing Match Presentation Match Rate) and FMMPMR (Fully Mated Morph Presentation Match Rate) compared to other GAN techniques [3]. In other words, the GAN-based morph technique indicates the real threat potential to the face recognition system.

B. On the Use of Benford's Law to Detect GAN-generated Images

Inspired by the limitations of current CGI detection problems on GAN-generated images, Bonettini, et al. proposed a new approach to detecting fake images by applying Benford's Law on the DCT coefficients of JPEG images. They deem that the quantized DCT coefficients of the natural images are valid for Benford's Law, but the synthetic one is not [2]. Benford's Law has been used as an effective countermeasure to detect the anomaly in naturally occurring collections of numerical data. In most cases, the GAN-generated images will alter the image parameter which leads to an invalidation of Benford's Law [2]. Their work proposed an approach to constructing a Benford-related feature vector with a selected set of bases, DCT frequencies, and analysis of the JPEG Quality Factor and passing it through a Random Forest classifier. Their result indicates that their classifier can obtain a relatively high accuracy in detecting non-facial fake images generated by several popular GAN architectures [2]. In this case, it gives us confidence that their approach is possibly feasible to detect fake facial images.

III.Methodology

In this project, our primary method is to manipulate the DCT coefficients of JPEG images and detect the anomaly of the extracted feature by utilizing Benford's Law.

A. DCT

DCT essentially stands for Discrete Cosine Transform. DCT coefficient values explore pattern recognition within an image with the usage of a quicker Fourier transform [10]. Our approach was to use DCT and extract the important coefficients from prominent face features. The image would first be passed into the program and it's normalized using the luminance values. We outline the image into pixel blocks, and the image is then grouped into several 8-by-8-pixel blocks. We apply the DCT transform on each pixel block.

To apply DCT transform to a pixel block, we are required to understand how it'll be computing the coefficients. For starters, DCT is usually applied and worked with one-dimensional data [11]. However, since we'll be working with two-dimensional data, we need to modify our DCT call in our program. To use it on two-dimensional data, we need to apply it to the transpose of the matrix twice. This will result in the DCT coefficients of both dimensions together, instead of just having the value of one dimension. The numerical results are expressed in the form of cosine wave frequencies when graphed and the most important aspects of the image, such as the most prominent face areas, are compressed and stored in the larger coefficients.

B. Benford's Law

Benford's Law is also known as the First Digit Law or Significant Law. Its focus is on exploring the statistical frequencies of real-life numerical datasets [2]. It's been known to be used in finances, world populations, and other aspects that contain real-life numerical sets. Benford's Law follows a numerical equation where it obtains the probability sum of a digit, with that digit being a number in the range of 1 to 9.

$$p(d) = log_{10}\left(1 + \frac{1}{d}\right)$$

d	1	2	3	4	5	6	7	8	9
$p(d)$	30.1%	17.6%	12.5%	9.7%	7.9%	6.7%	5.8%	5.1%	4.6%

Table 1. Benford's Law baseline probabilities

We decided to use Benford's Law because we believe that it serves as a great baseline for the DCT coefficient values. Plus, since it is commonly used in real-life applications, we believe that it could also be applied to DCT-processed face image data. Furthermore, we can determine if the face image is real or not, depending on how well it lines up to the predetermined baseline.

C. Divergence for Classification

To determine whether the extracted features of the images, we implemented the classifier with a combination of three divergences- Reyi, Tsallis, and Kullback-Leibler as follows:

$$Reyi\text{: } D^{R}(\widehat{p}|p) = \frac{1}{1-\alpha}(log \sum_{d=1}^{9} \widehat{p}(d)^{\alpha} * p(d)^{(1-\alpha)}) \ where\, \alpha = 0.3 \qquad (2)$$

$$Tsallis\text{: } D^{T}(\widehat{p}|p) = \frac{1}{\alpha-1}(\sum_{d=1}^{9} (\widehat{p}(d)^{\alpha} + p(d)^{\alpha}) + \sum_{d=1}^{9} (\widehat{p}(d) * p(d))^{\alpha} \ where\, \alpha = 0.3$$

(3)

$$Kullback - Leibler\text{:: } D^{KL}(\widehat{p}|p) = \sum_{d=1}^{9} \widehat{p}(d) * log\frac{\widehat{p}(d)}{p(d)}) \qquad (4)$$

The primary concept to use divergence is to measure the difference between the empirical $\widehat{p}(d)$ with its ideal Benford's Law $p(d)$. If the empirical data is below the threshold, it can be classified as following Benford's Law. Otherwise, it should be a violation.

IV. Data

For our dataset, we initially attempted to obtain the facial dataset from the National Institute of Standards and Technology (NIST) database. Due to the limited time, we could not afford to wait for the clearance of the license submission to be accepted so we opted to find a dataset on Kaggle. We found the one called "Real and Fake Face Detection" [6]. We opted for obtaining over 2000 fake and real facial images in JPEG format from the dataset. Our original plan was to train our StyleGAN3 model by using the dataset on Kaggle. However, the training process of StyleGAN3 requires high-performance GPUs. Even with the server (no GPU option) provided by Dr. Belman [10], we could not successfully implement the training of our model. Therefore, we ended up choosing 400 images from the Kaggle dataset, 200 real faces, and 200 fake faces. Among the fake image dataset, 100 were labeled easily identifiable as fake while the rest were more difficult to identify. These images involved were all colored and had a 600 x 600 dimension.

For a reason of further analysis in our later step, we also used the StyleGAN3 pre-trained model (FFHQ)to generate 50 super realistic facial images[5]. These images were converted into JPEG format and resized in the same dimension as the Kaggle dataset. The generated images were used as further verification of our result obtained by using the DCT features and Benford's Law.

V. Experiment

To implement our project, we formulated our experiment into four sections- data preprocessing, feature extraction, classification, and further verification. The implementation includes a DCT coefficient and first-digit probability extractor and two classifiers, one with the divergence method, and the other with the chi-square method.

A. Preprocessing

At the first step, we had to preprocess and determine the data that we were planning to pass through to our extractor. Two sets of data (a total of 400) from the Kaggle dataset were chosen, one containing 200 real faces, and the other containing 200 GANs-generated fake faces. Among the 200 fake images, 100 were classified as easy-level, and the other 100 were classified as hard-level [6]. For the purpose of further verification, 50 realistic fake face images generated by StyleGAN3 were chosen as well.

After setting up the necessary datasets, we passed them through the DCT extractor constructed by using the Scipy Python library. All the JPEG images were converted into grayscale by using the luminance values, reducing the features by ignoring the color scale [2]. After that, the extractor extracted the DCT values from the images which work as an intermediate variable passing through to the first digit probability feature extractor in the next step.

B. First Digit Probability Feature Extraction

After extracting the DCT values from the previous step, the counting first digit probabilities function in the extractor module would compute the first digit probability of each image according to the base of 10. In this process, the extractor counted the DCT value frequencies starting with the first digit from 1 to 10 respectively. The extraction features were stored in a spreadsheet. Each feature entry extracted (Fig.3) could be visualized as follows:

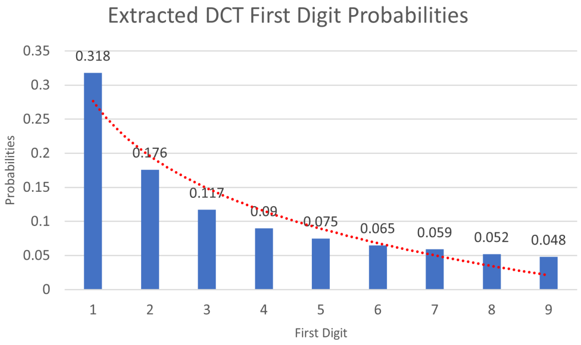


Fig. 2 Extraced DCT First Digit Probabilities

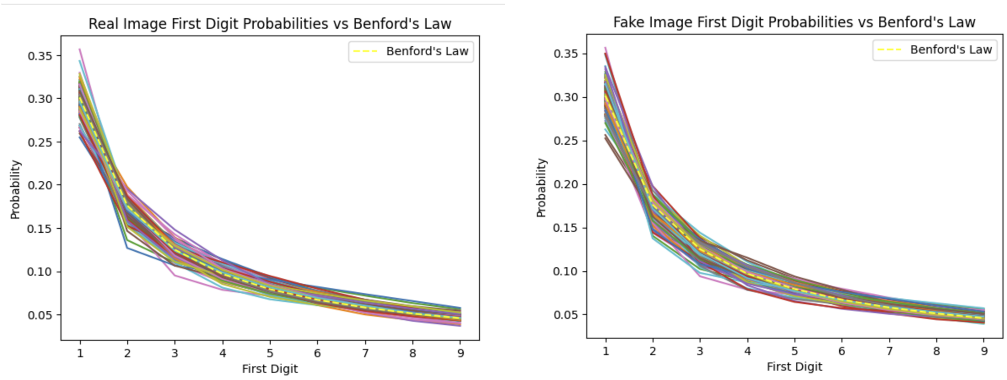
C. Classification

After we extracted the first digit probabilities of the DCT value from our data, we passed the extracted features into our classifier. We constructed a simplified version of the classifier by reference to Bonettini, et al.'s method [2]. The classifier combines three divergences- Reyni, Tsallis, and Kullback-Leibler that we mentioned in the methodology section, and the alpha was set as 0.3. Alpha smaller than 1.0 is more sensitive to differences in smaller probabilities between the empirical data and Benford's Law criteria. A heuristic threshold of 0.0002 was chosen for making a decision. After passing through the extracted features into the classifier, we were able to obtain the boolean results of the classification process.

VI. Result

A. The First Digit Probabilities Extraction

After passing through the two sets of testing data into the extractor, we successfully extracted the first digit probabilities of the DCT values from these JPEG images. In Fig. 3, we visualized the extracted features of real and fake images compared with the benchmark, Benford's Law (dashed yellow) expected values. The results of the real images performed as we expected with a minimal deviation from the benchmark values. However, the fake images performed similarly to the real images which we did not expect. To detect the anomaly of the fake images, a relatively observable difference from the benchmark should be detected.



a). Real Images vs. Benford's Law (dashed yellow)

b) Fake Images vs. Benford's Laws (dashed yellow)

Fig. 3 Extraced DCT First Digit Probabilities vs. Benford's Law

As a result of the unexpected performance, we computed the mean value of empirical FDP (First Digit Probabilities) of the real and fake images and compared them with Benford's Law. In Fig 4, the curves of the FDP mean of both real and fake images are superimposed with the expected values of Benford's Law. The behavior of the extracted data implies an inefficient result of the later classification.

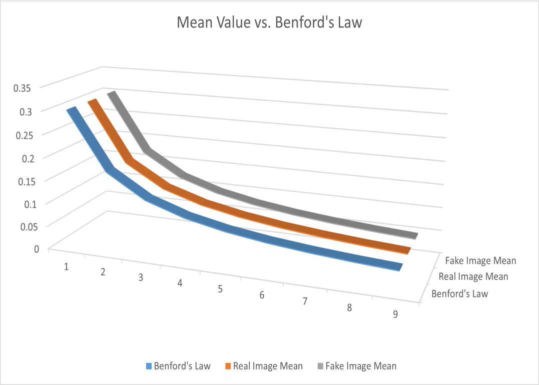


Fig. 4. FDP Mean of Real& Fake Images vs. Benford's Law

B. *Accuracy*

After feeding the extracted features into our classifier that combines three divergence functions, we obtained a result of the detection depicted in Fig 5. 48 of the real images and 50 of the fake images were classified as following Benford's Law. 152 of the real images and 150 of the fake images were classified as violating Benford's Law. Overall, we obtained a 49.5% accuracy in the classification which is much lower than we expected. A high FNR (False Negative Rate) which is 76% and a low TPR (True Positive Rate) which is 24% (Fig. 6) indicate the reason for the poor accuracy of our classifier. The high FNR and low TPR mean that our classifier is not very effective at identifying positive cases. Even if we tried to set a larger threshold, the performance became worse in FPR than the heuristic one (0.0002) that we chose. The deficiency of the classification module is probably due to the intrinsic of the DCT which is a frequency domain. We couldn't conclude our hypothesis at this point so we tried another method for the classification by using chi-square.

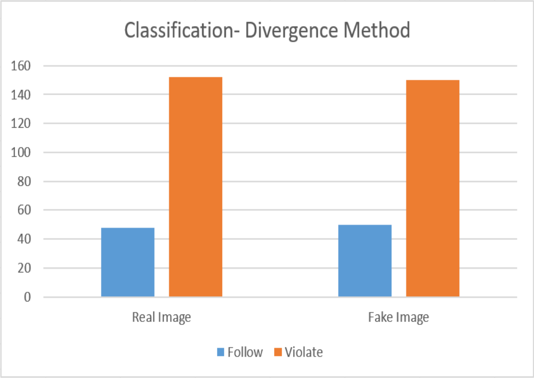


Fig. 5 Classification Result By Using Divergence Method

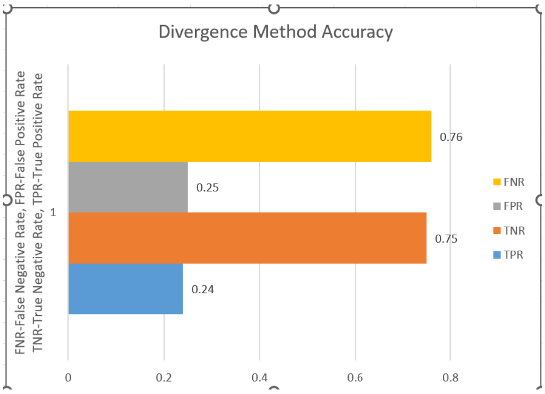


Fig. 6 Accuracy of Divergence Method

C. *Chi-Sqaure Method*

To conclude our findings, we decided to implement another classification method by using the Chi-Square Test on our data. Mostly due to the fact that we wanted to confirm our new hypothesis, which is that maybe the usage of DCT values with Benford's Law is not optimal.

$$x = \sum \sqrt{\frac{(observed - expected)^2}{expected}}$$

Using the Chi-Square equation, we define which variables will stand for which data. Our observed values will be the values obtained from the DCT. Meanwhile, we define our expected values as the baseline values from Benford's Law. We also defined the null hypothesis as "there exists an association with the values", and the alternative hypothesis as "there does not exist an association". To determine this, we compare the computed result with a predefined p-value of 0.05. If the resulting p-value from the Chi-Square Test is greater than the predefined value, then we accept the null hypothesis. Otherwise, we accept the alternative hypothesis. Using the provided equation, we sum the DCT coefficients with the baseline values and we obtain our results. We do this 200 times for the real and fake images.

Once we obtained our results, we were shocked with half of our results. When it came to the real images, the Chi-Square Test determined that the DCT coefficients of the real images follow the baseline values. In other words, the values follow Benford's Law, therefore deeming that the equation serves as a good measurement tool. However, when it came to the fake images, none of the image's DCT values violated the baseline, therefore concluding that the fake images also follow Benford's Law, as seen in Fig. 7. This is incorrect however, we expected the most fake images to fail and violate Benford's Law. Instead, we ended up with all 200 fake images following Benford's Law.

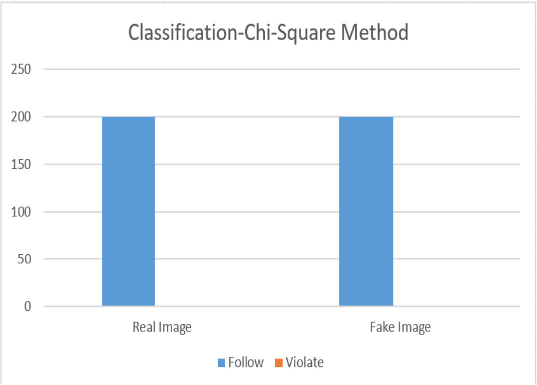


Fig. 7

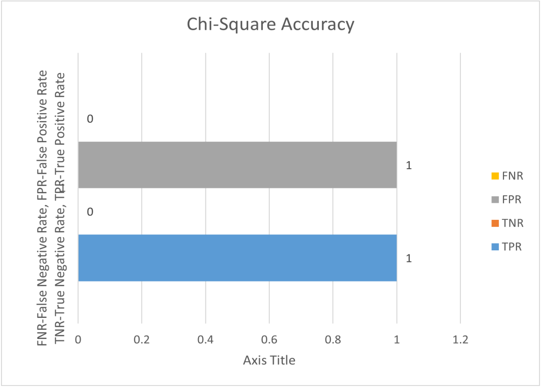
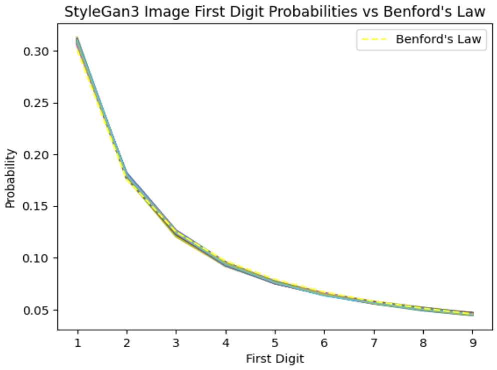


Fig. 8

D. Further Analysis

To end our research and conclusions, we decided to test Benford’s Law with some generated images using StyleGAN3[5]. We lowered our baseline of images from 200 to 50, mostly due to our limitations in generating due to our equipment. We also set the truncation trick to 0.8, which made the images similar to each other. We follow the same process and extract the DCT coefficients, obtain Benford’s Law baseline values, and classify them using Reyni, Tsallis, and Kullback-Leibler divergences. We then graph our DCT coefficients and Benford’s Law values and we end up with the following result:



What can we conclude from the graph? After analyzing the results, we came to the conclusion that StyleGAN produces some realistic fake images, therefore making the images follow Benford’s Law. Given that the StyleGAN value line differed or contained high and low spikes throughout, then we can conclude that the images do not follow Benford’s Law. The values were superimposed to Benford’s Law baseline and as a team, we concluded that StyleGAN serves as a great fake face image generator because there are no high spikes or abnormalities in the plotted graph. This brought up the question that maybe our test could serve a different purpose. Benford’s Law and DCT coefficients can serve as a test of how accurately a GAN can produce a “realistic” image. There’s the possibility of incorporating new algorithms or image processing formulas that can improve our original statement and also benefit our newly generated question.

VII. Conclusion

Through our experimentation of using Benford’s with DCT values, we have come to the conclusion that Benford’s Law is not an optimal method for GANs-based image detection. This is because GAN attempts to minimize the difference between the real and fake samples, which resulted in a graph that was on par with Benford’s Law graphs. Effectively showing that it was perfectly capable of replicating a passing standard. As a result, this outcome can come to be taken into consideration as a test for GAN algorithms and see if they pass a Benford’s law test. Although our approach was incapable of classifying fake and real images, we fully believe it's still possible to use Benford’s law with other feature extraction algorithms.

Acknowledgement

We would like to thank Dr Belman (Fresno State University) for providing us with the opportunity and resources to develop this program. We would also like to thank Karras, T, and their team for the open-source StyleGAN generator. We would also like to thank Nam, S, and their team for the 1000 generated images over at Kaggle. Lastly, we would like to thank the University of Fresno State for once again giving us the opportunity to develop and show our work.

References

[1] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.00453

[2] Bonettini, N., Bestagini, P., Milani, S., & Tubaro, S. (2021, April 5). On the use of Benford's law to detect GAN-generated images. GitHub. https://github.com/polimi-ispl/icpr-benford-gan

[3] S. Venkatesh, H. Zhang, R. Ramachandra, K. Raja, N. Damer, and C. Busch, "Can GAN Generated Morphs Threaten Face Recognition Systems Equally as Landmark Based Morphs? -- Vulnerability and Detection," arXiv:2007.03621 [cs.CV], Jul. 2020. [Online]. Available: https://arxiv.org/abs/2007.03621

[4] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? CoRR, abs/1904.03189, 2019.

[5] Karras, T., Aittala , M., Laine , S., H\"ark\"onen, E., Hellsten , J., Lehtinen , J., & Aila, T. (2021). *Alias-Free Generative Adversarial Networks*. GitHub. https://github.com/NVlabs/stylegan3#readme

[6] Nam, S., Oh, S. W., Kang, J. Y., Shin, C. H., Jo, Y., Kim, Y. H., Kim, K., Shim, M., Lee, S., Kim, Y., Han, S., Nam, G., Lee, D., Jeon, S., Cho, I., Cho, W., Yang, S., Kim, D., Kang, H., … Kim, S. J. (2019, January 14). *Real and Fake Face Detection*. Kaggle. https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection

[7] Open AI GPT-3.5: https://chat.openai.com/

[8] Bing AI: https://www.bing.com/search?q=Bing+AI&showconv=1&FORM=hpcodx

[9] Amith Kamath Belman

[10] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," in IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, Jan. 1974, doi: 10.1109/T-C.1974.223784.

[11] Khayam, Syed Ali. "The Discrete Cosine Transform (DCT): Theory and Application." Michigan State University 114.1 (2003): 31