Comparison of Intelligent Deep Fake Identifiers Using Machine Learning

Daniel Felix[1], Aarush Kachhawa[2], Prahit Yaugand [3]

[1]Saint Francis High School, danysamf1@gmail.com
[2]Saint Francis High School, aarushkachhawa@gmail.cm
[3]Mission San Jose High School, prahit.yaugand@gmail.com

ABSTRACT

Computer-generated fake images – known as "deep fakes" – are often utilized to misconstrue public perceptions, leading to widespread confusion and controversy. Recently, the advancement of deep fake technology has led to the production of ultra-realistic fake images – decreasing the efficacy of previously used detection tools. Current classification models detect deep fakes through the identification of abnormal features, such as pupil irregularities and color discrepancies. Nonetheless, images can easily be created to deceive these models using Generative Adversarial Networks (GANs). GANs are created to identify and abuse the weaknesses of a classifier to generate undetectable images. Thus, by creating more accurate detection tools, deep fake images become even more indistinguishable. In order to aid the classifier, preprocessing methods can be created to highlight the discrepancies in images. Therefore, when the altered image is fed in the network, it can be easily categorized as genuine or false. Our evaluation metrics grade different preprocessing procedures and compare them against a normal convolutional network to test for improvements.

Keywords: DeepFakes, Convolutional Neural Network (CNN), Discrete Cosine Transform (DCT)

I. INTRODUCTION

Modern-day deep fake technology has the power to create ultra-realistic fake images, which enables the spread of misinformation. Anyone with access to a smartphone can create confusion and chaos with deep fake images of people. Initially, deep fake generators produced low-quality images which looked warped and visibly fake to the human eye. However, over the past few years, deep fake generators have improved tremendously, making detection much more complex. The rise of new powerful networks such as GANs brought about the production of high-quality fake images. GANs or Generative Adversarial Networks consist of two neural networks that compete with each other to improve. In this case, the generator, one of the networks, creates deep fake images while its counterpart, known as the discriminator, classifies them [1]. Due to the successful nature of these competitive networks, many studies work to detect images produced by them. As a result, numerous approaches have been formulated by researchers. Most approaches attempt to classify a particular feature to distinguish false images from genuine ones. Many papers focus on acquiring more data for accurate results, while others focus on identifying facial features such as irregularities with the shape of the pupil in the image. More recent approaches have even looked into the color discrepancies of the face. Nonetheless, these techniques are all designed to be utilized independently and do not have the functionality to be combined with other methods.

Our goal is to utilize various preprocessing techniques, which work to alter the original images before feeding them into the neural network, and compare these methods to find the most accurate and robust solution. Since preprocessing is applied prior to feeding data into a classifier, our techniques have the capability to be used to bolster other existing or future detectors. One technique involves the use of the sharpen kernel, which increases the contrast in the image. The sharpen kernel is a 3 by 3 grid with the center containing the value 5, corners containing the value 0 and remaining values -1 [2]. It is multiplied with each image section to create the new image. This method is quick and can be applied to any image easily.

We also plan to implement another novel technique that breaks down an image into frequencies to determine the validity of the image. These frequencies, which are invisible to the naked eye, are detectable with the use of a Discrete Cosine Transform or DCT. The DCT is a method to break down an image into its underlying pixel frequencies [3]. Furthermore, the DCT can take in numerical data

and then convert them into sums of cosine with alternative frequencies. This can be utilized in deep face detection as it can capture features of an image that are typically lost in the pixels of the image. Through this novel technique, we will have the capabilities to detect more advanced deep fakes.

II. LITERATURE REVIEW

The advent of deep fake technology has brought the power to create misinformation to the general public. Initially, deep fake generators produced low-quality images which look warped and visibly fake to the human eye. However, the rise of new powerful networks such as GANs has brought about the production of high-quality fake images. Thus, deep fake detection has been a hot topic for machine learning researchers. As a result, several novel detection techniques have emerged, ranging from feature to color discrepancy detection in the image.

Deep fake detection technology has evolved considerably. One of the first models we researched used an incremental process. In this method, models were continuously fed data, which allowed for later data collection. These models worked well, starting at 81.5% accuracy and increasing, but the model improved at a slower rate when more data was collected. Also, although the model did perform quite well on the training data, it was prone to overfitting [4]. Another novel detection model focused primarily on irregularities of facial features; it used eye pupil distortions to classify images as deep fakes with an accuracy of 91% [5]. However, we managed to find a model with even higher accuracy. This model with an accuracy of 99.7% used color discrepancies in order to identify deep fakes [6].

The aforementioned models focused on deep fakes created by the GAN models, and thus the goal of their work was to detect GAN-created fake images. However, these models only focused on a particular type of feature and did not have the capability to be used in conjunction with other techniques. Additionally, while the incremental model and the color disparity model could be extended to finding fake images in objects, the pupil differentiation mode was limited to human images. Another limitation of the pupil irregularity model was that it based its model on the assumption of pupil shape regularity which may not always be true. As deep fakes become more realistic, the distortion of human features becomes less evident, leading to a decline in detection accuracy. Finally, as mentioned above, one issue with the incremental model was that it was prone to overfitting due to a large amount of data.

In order to address the gaps in past research, we propose to create multiple preprocessing methods. One removes irrelevant features in the image, and the other highlights important features of a GAN-created image. By using the Discrete Cosine Transform (DCT), we are able to separate an image into its high-frequency components. After analyzing the High-Frequency Components (HCF) and extracting features of pixels in images with repetitive structures - such as a facial image with skin and hair - we are finally able to use a neural network to determine the pattern with repetitive structures. Additionally, our second technique utilized a sharpened kernel. The sharpen kernel is a method of image processing which gives an image more contrast, thus highlighting hidden details. It splits an image into multiple components, where each pixel is then contrasted with the pixels around it. Through the use of these two processing techniques, we are able to not only address the gaps in research, but provide a system that can be utilized by future research as well.

III. METHODS

Deep fake detection models are often difficult to train and time consuming. The use of DCTs allowed shorter training time for each epoch and thus permitted our model to utilize more epochs, thereby obtaining a superior accuracy level.

A. Dataset

A large and robust dataset was required to effectively detect DeepFake Images. We used a large dataset from Kaggle [7]. This repository contained 70,000 deep fake images - generated by StyleGan - and 70,000 real images - from Flickr. The images were colored and shared the same size, 256x256 pixels. Additionally, the images included a variety of different faces, ranging in age and facial features. The zip file of the data was uploaded to our drives. We

then unzipped the images using Google Colab and used 30,000 of those images for our project.

B. Preprocessing

Before the data was inputted, it was essential to transform and normalize it. Since the images from the dataset were colored, each pixel had 3 values from 1 to 255. Thus, each image was shaped into a 256x256 numpy array with 3 channels of RGB in each element of the array.

C. Sharpen Kernel

The sharpen method utilized kernel multiplication to transform the image. We used a kernel of size 3x3 with a 5 in the center, -1s to the sides and 0s on the corner. This helped show the contrast in the image, thereby allowing the neural network to analyze more features resulting in better accuracy.

D. Discrete Cosine Transform

The discrete cosine transform is an even mathematical function; it converts a finite number of signals into an array of frequencies, which can then reconstruct the original image [8]. It was used to analyze 8x8 pixel sections in an image. The 256x256 array was split into multiple 8x8 arrays, where the Discrete Cosine Transform was performed. The Discrete Cosine Transform mapped each RGB pixel value into a frequency value. These 8x8 kernels were then fed into a neural network which classified the images as real or fake.

E. Convolutional Neural Network

A Convolutional Neural Network was chosen due to its effectiveness in image classification. As shown in Figure 1, the network utilized a sequential model with 3 convolutional layers alternated with 3 pooling layers. The model used dropout and batch normalization in each layer to regularize and normalize the data in the model. ReLU was used as the activation function in every layer as it provided fast and simple computation for the data. The final layer employed the sigmoid activation function, to scale the output in the range from 0 and 1 for the binary classification system.

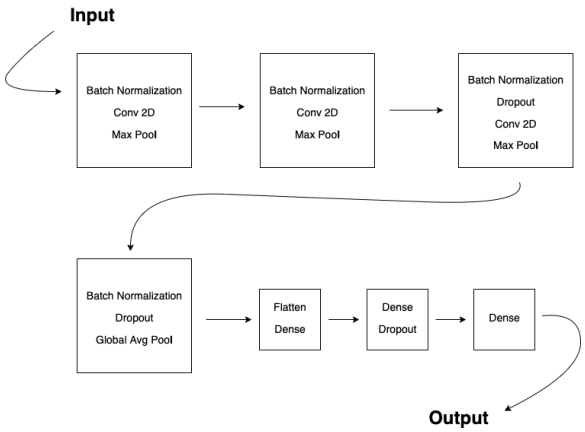


Figure 1. Diagram of the architecture of the Convolutional Neural Network

F. Evaluation

In this study, the performance of the model was evaluated with the following metrics: accuracy, precision, recall, and F1-score. Accuracy is the percentage of correct predictions to total predictions made by our model. In this case, precision would be the ratio between images classified as real and total number of real images classified. Recall is the ratio of correct predictions that an image is a deep fake in comparison to all deep fake images which were classified. F1-score is the harmonic mean of precision and recall. Additionally, a confusion matrix summarized the prediction results of the model. The values in the confusion matrix: true positive (TP), true negative (TN), false positive (FP) and false negative (FN), were used to calculate the evaluation metrics, as shown in Table 1. The model predictions of the test data – which employed 20% of the overall data – were used to construct the confusion matrix. Finally, we utilized various graphs – ROC curve, training graph and loss graph – to further visualize the effectiveness of the model's predictions and rate of the training process.

Accuracy = (TP + TN) / (TP + TN + FP + FN)	Precision = TP / (TP + FP)
Recall = TP / (TP + FP)	F1 score = 2 * (precision * recall) / (precision + recall)

Table 1. The calculations of evaluation metrics using values from a confusion matrix

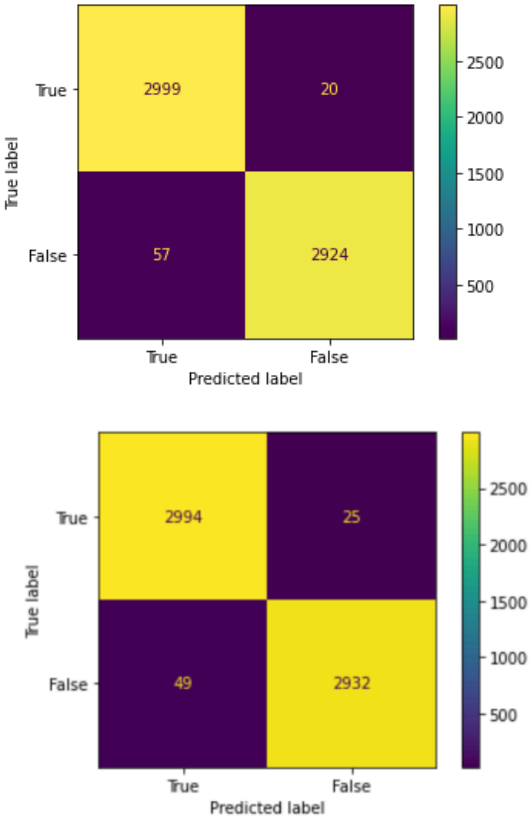
IV. RESULTS

Once our model was fully trained, we evaluated its performance using various evaluation metrics, as shown in Table 2. The testing accuracy was a marker of the model's success rate on predicting the input data. However, unlike the accuracy score, the precision, recall, and f1 score values were calculated independently for real and fake images. These scores varied minimally for the basic CNN, but varied dramatically for the rest of the preprocessing approaches.

Type	Accuracy	Precision (0, 1)	Recall (0, 1)	F1 Score (0, 1)
Grayscale DCT 256 * 256 pixels	98.8%	98%, 99%	99%, 98%	99%, 98%
Sharpen Kernel 256 * 256 pixels	98.7%	98%, 99%	99%, 98%	99%, 99%
Grayscale 256 * 256 pixels	98.0%	97%, 99%	99%, 97 %	98%, 98%
Basic CNN 256 *256 pixels	97.1%	95%, 99%	99%, 95%	97%, 97%

Table 2. Model scores for accuracy, precision, recall and f1 score.

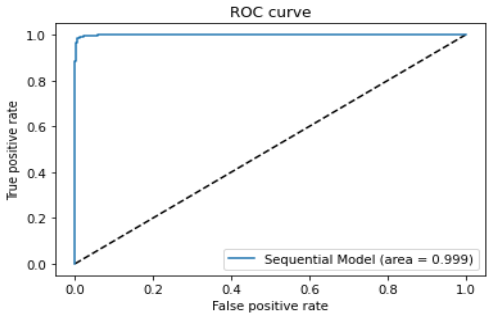
A confusion matrix was utilized in order to visualize the biases of the model when predicting different classes. Figure 1 shows the breakdown of the model's predictions for true positive, true negative, false positive and false negative cases, based on each preprocessing approach.

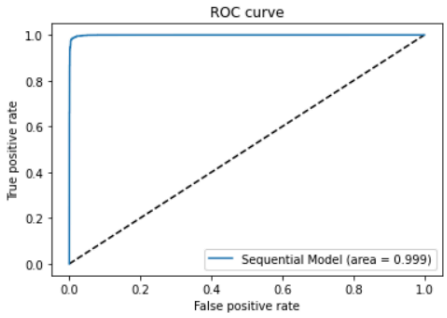


Sharpen Kernel 256 x 256 Pixels
Grayscale DCT 256 x 256 Pixels

Figure 2. Confusion matrix for each preprocessing method

Additionally, as shown in Figure 2, the ROC curve of the model demonstrates the tradeoff between specificity and sensitivity. The area under the ROC curve, a number between 0 and 1, demonstrates the model's ability to distinguish between real and fake images.



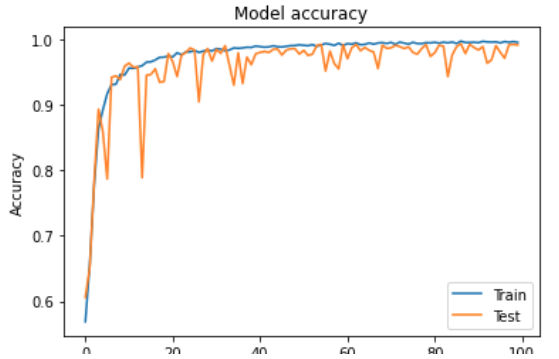


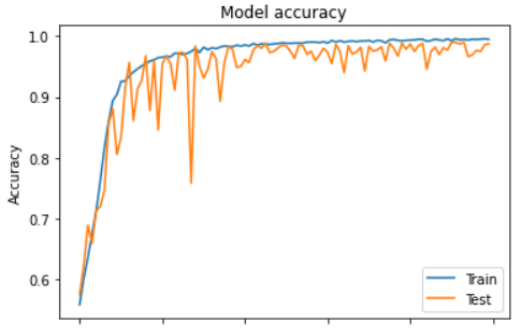
Sharpen Kernel 256 x 256 Pixels

Grayscale DCT 256 x 256 Pixels

Figure 3. ROC curves for each preprocessing method with AUC scores

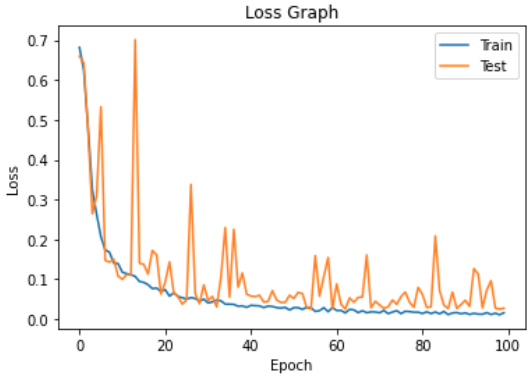
Finally, as shown in Figure 3, the model training and validation accuracy scores were graphed in order to visualize the results of model training. In Figure 4, the loss value graphs for training and testing were also plotted to help visualize the model's success of minimizing loss.

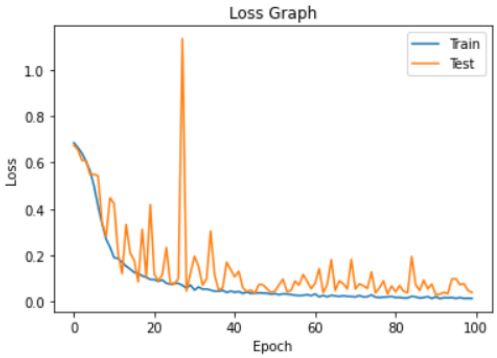




Gr

Figure 4. Model training and testing graphs for each preprocessing method





Sharpen Kernel 256 x 256 Pixels

Grayscale DCT 256 x 256 Pixels

Figure 5. Loss graphs for each preprocessing method

Out of all these methods the Grayscale DCT method performs the best in terms of accuracy. However the preprocessing method utilizing the sharpen kernel was not far behind, even outperforming the DCT model in terms of f1 score. The Sharpen kernel was better at classifying true values while the DCT method was better at classifying the false ones. Our final results help us conclude that the two preprocessing methods helped increase the accuracy of the simple convolutional model with RGB and Grayscale images.

V. DISCUSSION

Both the sharpened image and DCT techniques were extremely successful in aiding the CNN to classify Deep Fake images at 98.7% and 98.8% respectively.

Given that the Deep Fake images our model trained on were extremely difficult to distinguish with the naked human eye, both preprocessing techniques proved not only able to discern between real and fake images, but at a much higher rate than humans. Thus, both methods demonstrated their viability as detection tools in response to hyper-realistic deep fakes. Another benefit to using these methods is that preprocessing can be applied to any neural network.

Though we were able to amass a high validation accuracy for both preprocessing techniques, there were adjustments made in order to allow for cheaper computational costs. Due to computational capacity, RGB images had to be converted to Grayscale for our DCT function. If we had more RAM, we would be able to convert our RGB image to YCbCr format which would provide more features and therefore might result in higher accuracies.

While we were experimenting with different types of sharpen kernels, we delved into the field of kernels. Further research could be expanded to this sector to find better kernels to enhance the model. Researchers could also use the DCT model that we created as it also increases the accuracy.

VI. CONCLUSION

This study sought to increase the accuracy of a Deep Fake detection algorithm using various preprocessing methods.To solve this problem, we created a regular Convolutional Neural Network model as the control in our experiment. We added several preprocessing techniques to the model to yield higher accuracy. These methods included using the basic grayscale method, the DCT method, and the sharpen kernel. Our results presented the effectiveness of an underutilized method, the Discrete Cosine Transform. It also demonstrated several preprocessing methods that could be used to improve the accuracy of any neural network classifier.

In the future, the use of a grid search algorithm can be utilized on distinct kernels to find the best kernel preprocessor. Furthermore, the Discrete Cosine Transform can be combined with the sharpen kernel, leading to accurate results for general case detection of deep fake images. Finally, successful preprocessing techniques can be mapped together in order to allow neural networks to extract more meaningful information from an image.

REFERENCES

1. K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng and F. -Y. Wang, "Generative adversarial networks: introduction and outlook," in *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588-598, 2017, doi: 10.1109/JAS.2017.7510583.
2. *Vepuri, K. S., & Attar, N. (2021). Improving the performance of deep learning in facial emotion recognition with image sharpening. International Journal of Computer and Information Engineering, 15(4), 234-237.*
3. *Rao, K. R., & Yip, P. (2014). Discrete cosine transform: algorithms, advantages, applications. Academic press.*
4. Marra, F., Saltori, C., Boato, G., & Verdoliva, L. (2019, December). Incremental learning for the detection and classification of gan-generated images. In *2019 IEEE international workshop on information forensics and security (WIFS)* (pp. 1-6). IEEE.
5. Guo, H., Hu, S., Wang, X., Chang, M. C., & Lyu, S. (2022, May). Eyes tell all: Irregular pupil shapes reveal GAN-generated faces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2904-2908). IEEE.
6. Li, H., Li, B., Tan, S., & Huang, J. (2020). Identification of deep network generated images using disparities in color components. *Signal Processing*, *174*, 107616.
7. 140k Real and Fake Faces. (2020, February 10). Kaggle. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces
8. Giudice, O., Guarnera, L., & Battiato, S. (2021). Fighting deepfakes by detecting gan dct anomalies. *Journal of Imaging*, *7*(8), 128.