

AI Betrays AI: An Exploration of Applying Machine Learning in Facial Recognition

Mike Mao¹, Guillermo Goldsztein[#] and Joanna Gilberti[#]

¹The Bear Creek School, USA

[#]Advisor

ABSTRACT

The rise of AI deepfakes following the launch of ChatGPT and its AI counterpart DALL-E has sparked fear that the boundary between real and fake can no longer be identified. In this study, it was found that machine learning algorithms can be reliably used to distinguish between real and AI-generated images of human faces when provided with high resolution 300x300-pixel images with an accuracy score of 99.07%. This paper will cover the findings of this study by reviewing the main ideas behind machine learning, supervised learning, and neural networks and then examining the application of these techniques to a binary classification problem involving image classification.

Introduction

The influence of machine learning and AI on today's society became noticeable in 2021 when San Francisco-based AI research and development company OpenAI launched its deep-learning AI image generator named DALL-E which aimed to generate high-quality digital images from textual descriptions. The release of its latest version DALL-E 2, coupled with the rise of famed large-language model ChatGPT, has sparked countless ethical, privacy, and security concerns regarding AI-generated synthetic media.

In September 2023, researchers Zeyu Lu et al., conducted a study benchmarking human and machine-learning model's ability to detect AI-generated images. They found that while "humans struggle significantly to distinguish real photos from AI-generated ones, with a misclassification rate of 38.7%", machine-learning models fare much better, achieving "a 13% failure rate under the same setting used in the human evaluation" (Lu et al., 2023).

This study re-explores the use of machine learning models to distinguish between real and AI-generated images by evaluating whether the difference between real and AI-generated images of human faces can still be reliably identified using a machine learning model trained on a set of 966 high-resolution 300x300-pixel real and AI-generated human faces. It was discovered over the course of this study that machine learning models can be reliably used to detect between real and AI-generated images with a 99.07% success rate.

This paper will begin by first explaining the methodology behind the model used, describing the machine learning process, the target of my research, and the construction and training of the model. This will then be followed by a summary of the results of my research and concluded by a discussion of this study's significance.

Methodology

Data Set Collection and Generation

This study aims to implement the use of a machine learning model in distinguishing between real and AI-generated images using a data set consisting of a total of 1289 high-resolution 300x300-pixel images of human faces, some of which were real images while others were AI-generated, retrieved from the public data set source Kaggle. This data

set was presented to the machine learning model as a collection of pictures and information on whether it is real or fake. Each pair consisting of an image and its information is what is known as an example, with the image called the feature of the example and its status as real or fake being the label. A section of the data set is shown in Figure 1.

Image-Label Sample from Data Set

Image	Label
	real
	fake

Figure 1. A sample from the data set depicting the feature and label pair for two examples.

The concept of using a machine learning model to accurately identify between real and AI-generated images falls under the category known as supervised learning. In supervised learning problems, a model is asked to predict the labels when the features of an example are fed to it as input. The general strategy within machine learning is to train the model on a collection of examples where the features and labels are known. The model is given each image-label pair. Throughout its training, the model learns to identify characteristics of the features that point towards a specific label.

More specifically, the focus of this study is what is known as a binary-classification problem. In a binary-classification problem, each example can take only one of two possible values. In this study, this means that each image can either be real or AI-generated. Originally, the labels of each example were a single word (“real” or “fake”). However, this cannot be understood by a machine learning model. So, one-hot-encoding must first be performed to replace a label of “real” with 1 and “fake” with 0. This means that the data in Figure 1 is transformed into the data in Figure 2. Additionally, images cannot be understood by a model. So, each image is converted into an array that shows the values of individual pixels.

Transformed Image-Label Sample from Data Set

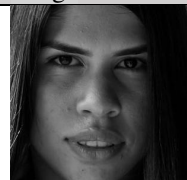
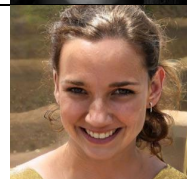
Image	Label
	1
	0

Figure 2. A sample from the transformed data set after one-hot-encoding.

The process of converting each image into a matrix can be understood by first understanding the composition of images. Each image is essentially a grid of cells called pixels, with each pixel taking a numerical value between 0 (white) and 255 (black) that determines its color. This is seen in Figure 3 which places an example image and its pixel representation side-by-side.

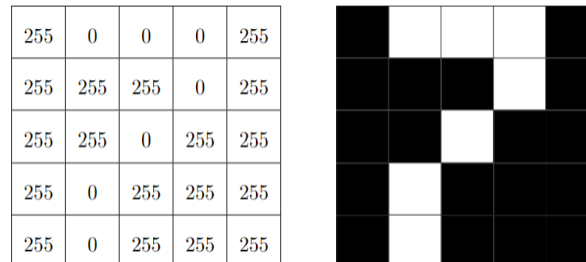


Figure 3. An example image and its pixel representation.

This grid of cells containing the color intensities of each pixel within an image is stored in matrix form for the model to read. It is crucial that the image is transformed into matrix format because throughout the machine learning process, the image must be manipulated in many ways through transformations or simple algebra. The above example image's matrix representation is shown in Figure 4.

$$\begin{bmatrix} 255 & 0 & 0 & 0 & 255 \\ 255 & 255 & 255 & 0 & 255 \\ 255 & 255 & 0 & 255 & 255 \\ 255 & 0 & 255 & 255 & 255 \\ 255 & 0 & 255 & 255 & 255 \end{bmatrix}$$

Figure 4. An example image's matrix representation.

Application of Machine Learning Model

In binary-classification problems, the model can be seen as a function that takes the features of an example as input and outputs the probability that the example corresponds with a specific label. Specifically, it can be seen as a function that takes in an array corresponding to an image and outputs the probability that the image is real. By convention the function is denoted by $\hat{y}(x_1)$ where x_1 can be seen as the array that is outputted. The image is classified as fake if $\hat{y} > 0.75$ and fake if $\hat{y} \leq 0.75$.

This model operates using what are called neural networks. In short, this means that the model consists of a series of functions (called layers) that are grouped together so that when the features of an example are passed into the input layer, it can eventually produce an output of the label. The output of each layer becomes the input of the following layer until the function \hat{y} is outputted in the output layer. An overall structure of a neural network with 5 layers is shown as follows.

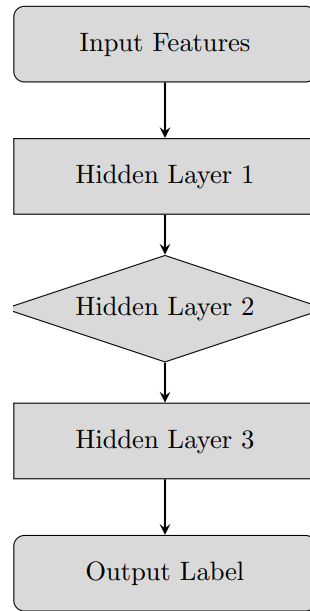


Figure 5. A diagram of the composition of a neural network.

There are several kinds of layers that can be utilized for machine learning purposes. Because layers are essentially the building blocks of a neural network, there are certain factors that determine a neural network. These factors are as follows:

1. The number of layers within the neural network.
2. The types of layers used.
3. The hyper-parameters that correspond to each layer.
4. The type of activation function that is used for each layer.

While there are countless types of layers that can be utilized for machine learning purposes, only convolutional, max-pooling, flatten, and dense layers were used.

For each convolutional layer, there is a 3 x 3 filter. This filter moves across each image and combines with the image to form a map of the image. This essentially allows the model to read each pixel present in the 300x300-pixel images presented to the model.

The max-pooling layer summarizes the features of the map generated by the convolutional layer. This simplifies the problem of determining if an image is real or AI-generated for the model because it only needs to focus on less data as the dimensions have been shrunk. Forcing the model to try to learn based off a huge 300x300-pixel image wouldn't make sense. After the max-pooling layer, the image data has now been shrunk, making the process of learning which characteristics determine a real image of a human face vs. an AI-generated one more feasible.

Even after the convolutional layer and max-pooling layer has been executed by the machine learning model, the data is still in a 2D format which is difficult to process. The flatten layer now converts the 2D data into a 1D array format, which is now much easier to process.

The two dense layers are the last layers present in the model. These are the layers that take the 1D data format presented by the flatten layer and use it to determine which characteristics determine if an image of a human face is real or AI-generated and classifies each image. Once the dense layers have been executed, the model has now learned to distinguish between real and AI deepfakes.

One pass of the data set isn't enough to ensure that the model has truly learned from the data set of thousands of images. The above process involving all 5 layers is then executed 9 more times. Each time the model passes through the entire data set is called an epoch. The capability of the model to distinguish between real and AI-generated images is now ensured with 10 epochs.

Just like there are several kinds of layers that can be utilized to determine a neural network, there are also several different activation functions that can be applied to each label. These activation functions are crucial for machine learning because they give the model the ability to identify non-linear relationships between the input and the label, crucial for image classification because whether an image is real or AI-generated may not be obvious at first. For this study, the activation functions utilized were the sigmoid and rectifier activation function (ReLU).

The sigmoid function is an activation function mathematically defined by the following formula:

Equation 1: The equation for the sigmoid activation function.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

It has a series of important properties which are as follows:

1. $0 < \sigma(x) < 1$ for all x .
2. The sigmoid function is both continuous and increasing for all x .
3. As x approaches $-\infty$, the value of $\sigma(x)$ becomes arbitrarily close to 0. In contrast, as x approaches ∞ , the value of $\sigma(x)$ becomes arbitrarily close to 1.
4. A $x = 0$, the value of $\sigma(x)$ is 0.5.

A graph of the sigmoid function is as follows.

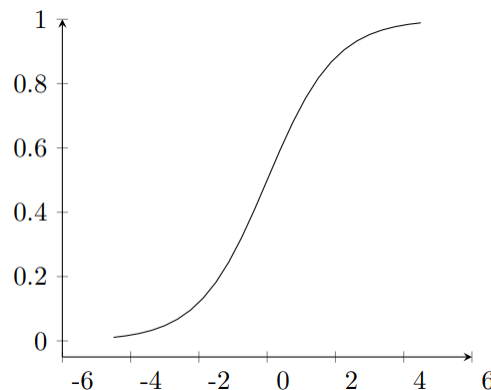


Figure 6. A graph of the sigmoid activation function.

The ReLU function is an activation function parametrically defined in mathematics by the following formula:

Equation 2: The equation for the ReLU activation function.

$$ReLU(x) = f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Like the sigmoid function, the ReLU function also has a series of important properties which are as follows:

1. The ReLU function is greater than or equal to 0 for all x .

2. If x is less than 0, the value of the ReLU function is 0.
3. If x is greater than 0, the value of the ReLU function is equal to x .
4. At $x = 0$, the value of the ReLU function is 0.

A graph of the ReLU function is as follows.

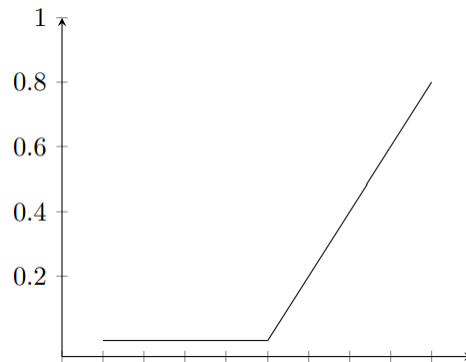


Figure 7. A graph of the ReLU activation function.

This model is then applied to the data set. In total the data set contains 1289 high resolution 300x300-pixel images of human faces of all races, some of which are real and others are fake. There is a total of 589 real human face images and 700 fake human face images. The data set is then randomly divided into a training and validation set. 75% of the images will fall under the training set and the remaining 25% under the validation set. This means that the training set consisted of 966 images while the validation set contained 323 images. The model is then trained on the training set using 10 epochs. Each epoch is essentially one "pass" of the data set through the model. So, the training set is passed 10 times into the model.

Results

Once the model is trained on the training set, it is validated using the validation set and its accuracy is measured. Over the course of the validation, the model performed exceedingly well with an accuracy of 99.07% success rate. Among the 323 images in the validation set, the model correctly predicted whether the image was real or fake 319 times, only failing four times.

Because the type of machine learning problem tackled in this study is a binary classification problem, the error measurement is what is known as a binary cross-entropy error. If the series $y_1, y_2, y_3, \dots, y_n$ is the series of labels for a set of examples and $\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots, \hat{y}_n$ is the series of predicted labels generated by the model for the same set of examples, the binary cross-entropy error is mathematically defined by the following equation:

Equation 2: The equation for the binary cross entropy error.

$$BCE(y, \hat{y}) = \frac{-1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

The binary cross-entropy error also has a series of properties that are important to understand. These properties are as follows:

1. $BCE(y, \hat{y})$ is always greater than or equal to 0.

2. The closer \hat{y} is to y , the smaller $BCE(y, \hat{y})$ will become.
3. If $y = \hat{y}$, then $BCE(y, \hat{y}) = 0$.

Note that because y represents the series of actual labels for a set of examples and \hat{y} represents the predicted labels for those examples, when y and \hat{y} are very close, the binary cross-entropy error $BCE(y, \hat{y})$ becomes really small, signaling that the model is very accurate.

When training its predictions, the model performs what is called logistic regression. This technique only works for binary classification problems. Logistic regression first assumes that the predicted labels are in the form $\hat{y} = \sigma(w_1x_1 + w_2x_2 + \dots + w_kx_k + b)$. The model then chooses the parameters w_1, w_2, \dots, w_k , and b such that the binary cross-entropy error of the model on the training set is as small as possible. These parameters then help define the model and generate the predictions on the validation set.

To further test the accuracy of the model and check for bias, a separate Kaggle data set was used containing images of human faces from different races. The races being: White, Black, Asian, Indian, and Other. 100 images of each race were taken, converted into arrays, and passed into the model. The probability that the model misclassified the photo as AI-generated was displayed.

It was found that the model had minimal bias for all 5 races of human face images, with the model having less than a 15% chance of mis-classifying the images as fake for all 5 races of human face images.

Discussion

Over the course of this study, it has been demonstrated that machine learning models are highly reliable for distinguishing between real human face images and AI-generated human face images when given high resolution 300x300-pixel images of both categories. Furthermore, it has been shown that when trained correctly, machine learning models are also capable of having a minimal bias of less than 5% when classifying these images.

While these findings are promising, it is anticipated that such a model may fail to perform up to its expectations in the real world when data sets may not be as balanced, and images are often not as high resolution. To see if machine learning models will be held up in a practical test, more research must be done.

Acknowledgments

I would like to thank everyone who has helped me along my research journey whether it be my mentor, advisor, or teacher, thank you for the valuable insight and guidance provided to me on this topic and the research process.

References

1. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill Science/Engineering/Math.
2. Burkov, A. (2019). The Hundred-page Machine Learning Book. Andriy Burkov, 1.
3. Lu, Z., Huang, D., Bai, L., Qu, J., Wu, C., Liu, X., & Ouyang, W. (2023). Seeing is not always believing: Benchmarking Human and Model Perception of AI-Generated Images. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.48550/arXiv.2304.13023>
4. Dettmers, T. (2022). "Deep Learning in a Nutshell: Core Concepts." NVIDIA Technical Blog. <https://developer.nvidia.com/blog/deep-learning-nutshell-core-concepts/>

5. Boulahia, H. (2021). Fake-Vs-Real-Faces (Hard).
<https://www.kaggle.com/datasets/hamzaboulahia/hardfakevsrealfaces>
6. Mistol, M. (2023). UTKFace (Aligned & Cropped). <https://www.kaggle.com/datasets/moritzm00/utkface-cropped>
7. TensorFlow Developers. (2024). TensorFlow (v2.17.0). Zenodo. <https://doi.org/10.5281/zenodo.12726004>
8. Keras: The high-level API for TensorFlow. TensorFlow. (n.d.) <https://www.tensorflow.org/guide/keras>
9. Mao, M. (2024). Machine Learning Model on Real vs. Deep-Fakes.
<https://www.kaggle.com/code/mikem27/machine-learning-model-on-real-vs-deep-fakes/notebook>