# Classifying AI Art:
# Differentiation of Text-to-Image Generative Models

Julie Chen

**Abstract**

Controversy over the ownership of digital art concepts and the art pieces themselves has arisen with the proliferation of artificial-intelligence tools. Efforts can be made in the distinction between "human-created" art and those of text-to-image generative models, as well as between popular AI art generators. Various image classifiers were tested for accuracy in distinguishing between the artwork created by the DALLE-2, Midjourney, and StableDiffusion generators, and common patterns among the generators were identified to improve classification performance. Among the tested supervised learning algorithms, a random forest approach was most effective at correctly assigning AI artist to its corresponding artwork, and dimensionality reduction can vastly decrease computation efforts while preserving predictive power of the models.

## 1. Problem Statement

A recent explosion in artificial intelligence software and tools has sparked controversy across many professions, one of which is the digital art market and the broader art community. Popular text-to-image generators powered by artificial intelligence such as DALL-E, MidJourney Bot, and StableDiffusion have led to a surge in what is now known as "AI art," which some people have creatively used to generate artwork quickly for profit. As phrased by Elizabeth Penava (2023) of The Regulatory Review, the rapid proliferation of artwork created by artificial intelligence as consumer products raises the question of "who regulates this digitally created art and whether the courts can prevent theft of creative ideas and techniques in the process of its generation."

Distinguishing between human-made art and AI-generated art is currently an important topic due to ownership claims of intellectual property, copyright, or other legal ramifications. The accepted definition of what can be considered "real artwork," both legally and colloquially, has been challenged by the development and refinement of modern AI generators. In addition, if AI-powered text-to-image generators become normalized and widely accepted as content creation tools, there may also be suitable applications for distinguishing between AI art generated by different text-to-image generators as well as between AI-produced versus human-produced work for ethical, legal, and business reasons.

In summary, given a large set of images generated by DALL-E 2, Midjourney, and StableDiffusion, this project aims to classify artwork produced by three popular AI generators. The following questions may be answered:

1) Can artwork by each AI generator be accurately attributed to each "artist" based on randomly selected images made with different users' text prompts?
2) What, if any, inherent patterns or features separate the outputs produced by each image generator?
3) In extension, can the works of AI generators that used the exact same text prompt be effectively distinguished from each other?

## 2. Data Source

Data was collected from three separate AI text-to-image generators databases:

– DALL-E 2 Image Gallery (https://dalle2.gallery/#search),
– Midjourney (Kaggle: Midjourney User Prompts & Generated Images), and
– DiffusionDB (https://poloclub.github.io/diffusiondb/)

OpenAI released DALL-E 2, successor to the original DALL-E 1, in 2022. At the time of this project, the DALL-E 2 image database contained 53,264 images generated by 32,733 prompts. For this project, a set of 10,000 images were scraped from the web gallery.

Midjourney launched their text-to-image service as a Discord server bot called the MidJourney Bot. The dataset consists of 272 JSON files that capture 250,000 links to images scraped from the Discord server from June 20, 2022 through July 17, 2022. For this project, 10,000 of those 250,000 links were randomly selected.

DiffusionDB is a large-scale prompt gallery dataset that contains over 14 million images generated by StableDiffusion. For this project, a random 10,000 data points were chosen from the DiffusionDB 2M subset provided by Georgia Tech's Polo Club of Data Science.

## 3. Methodology

### 3.1 Data Preprocessing

After scraping 10,000 images from each AI art generator for a total of 30,000 images, each of the images were converted to RGB pixel arrays and labeled by their corresponding generative models so that supervised learning methods could be used for analysis. Initial inspection of a few images determined that many of these images were too high-resolution, so they were downscaled to a resolution of 64 x 64 pixels for easier handling. Because the data from each data source were sized and formatted differently, some of the images gathered during the scraping could not be correctly resized or reformatted to match the rest of the images and were dropped from the dataset. To have an equal number of observations from each AI-generator, a subset of 9,000 correctly-formatted images were selected from each of the AI generator datasets for a total of 27,000 observations.

| AI Art Generator | Class Label | Number of Observations |
|---|---|---|
| DALL-E 2 | 0 | 9,000 |
| Midjourney | 1 | 9,000 |
| StableDiffusion | 2 | 9,000 |

*Table 1: Summary of dataset per AI art generator*

### 3.2 Classification of AI Generators

To answer the first project question of whether the AI "artist" can be identified from a given input image, an 80:20 stratified train/test split was used to partition the dataset proportionally into training and testing datasets. The following supervised learning algorithms were used to perform multiclass classification. Each model's performance was compared against the rest using confusion matrices, precision, and recall scores in answering the project objectives.

1. Naive Bayes
2. K-Nearest Neighbors
3. Random Forest
4. Support Vector Machine

#### 3.2.1 Naïve Bayes

For Naïve Bayes classification, the prior probabilities of each class label are 1/3 based on the setup of the dataset. Since each image in the dataset is represented as an array of RGB color values ranging from 0 to 255, inclusive, a Gaussian distribution is assumed and used for this model.

#### 3.2.2 K-Nearest Neighbors

In order to efficiently fit a K-nearest neighbors model on the training data with reasonable computational cost, the training images were downscaled from 64 x 64-pixel image arrays instead to 32 x 32-pixel image arrays. The optimal number of neighbors for the K-Nearest Neighbors classification model using the hold-out method was selected using a grid search. Furthermore, 5-fold cross validation was used on the entire dataset to compare model performance on unknown data. Tuning for K was not efficient, as the dataset has a large number of observations and features. A secondary examination of the K-nearest neighbors algorithm was implemented to see if predictive performance could be either sped up or improved after performing dimensionality reduction using principal component analysis, which will be discussed in §3.3.

#### 3.2.3 Random Forest

Tuning hyperparameters for the random forest classifier was completed using a grid search, with the best number of decision trees selected to be 200, and the number of features considered for each split was the square root of the total number of features. The performance of the best estimator was compared against that of the default classifier with 100 trees. The random forest classifier chosen for later analysis was constructed using 200 trees, with the number of features considered for each split was decided using the square root of the number of total number of features.

#### 3.2.4 Support Vector Machine

A radial kernel was selected for the support vector machine model. Similar to K-nearest neighbors, the training images were downscaled from a resolution of 64 x 64 pixels to 32 x 32 pixels to save on computational intensity. The support vector machine method was revisited after performing principal component analysis to determine whether classification accuracy could be improved or sped up.

### 3.3 Identifying Patterns

Principal component analysis was carried out to answer the second project question of identifying inherent patterns or features that distinguish AI art made by one generator from another. Performing principal component analysis on the entire training set with 64 x 64 images would result in crashing the runtime environment, so instead it was completed on the downscaled training set of 32 x 32 images mentioned in the K-Nearest Neighbors and Support Vector Machine sections above. A scree plot was generated to determine how many principal components to keep to build secondary K-nearest neighbors and radial support vector machine models.

## 4. Evaluation and Final Results

### 4.1 Performance of Classification Algorithms

Confusion matrices were computed for each of the aforementioned classifiers, as well as the corresponding accuracy, precision, recall, and F1-scores. Discussion and comparison of the performances of each classification model follows.

#### 4.1.1 Naïve Bayes

The Gaussian Naïve Bayes classifier achieved classification accuracy of 0.61, which is much better than randomly guessing. This classification model has notably higher precision for class 0 (DALLE-2) than for class 1 (Midjourney) or 2 (StableDiffusion). Out of all images that the Naïve Bayes classifier predicted as generated by DALL-E 2, 79% were truly generated by DALL-E 2. Recall for class 0 was much lower—out of all 1,800 images actually generated by DALL-E 2, only 52% were correctly labeled by the model as class 0. In fact, about one-third of the DALL-E images were incorrectly classified as Midjourney images. This classifier appears to more commonly classify images in the test dataset as generated by Midjourney rather than either DALL-E or StableDiffusion.

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| 0 | 0.79 | 0.52 | 0.63 |
| 1 | 0.52 | 0.69 | 0.59 |
| 2 | 0.62 | 0.62 | 0.62 |
| **Accuracy** | | | 0.61 |

*Table 2: Gaussian Naïve Bayes classification report*

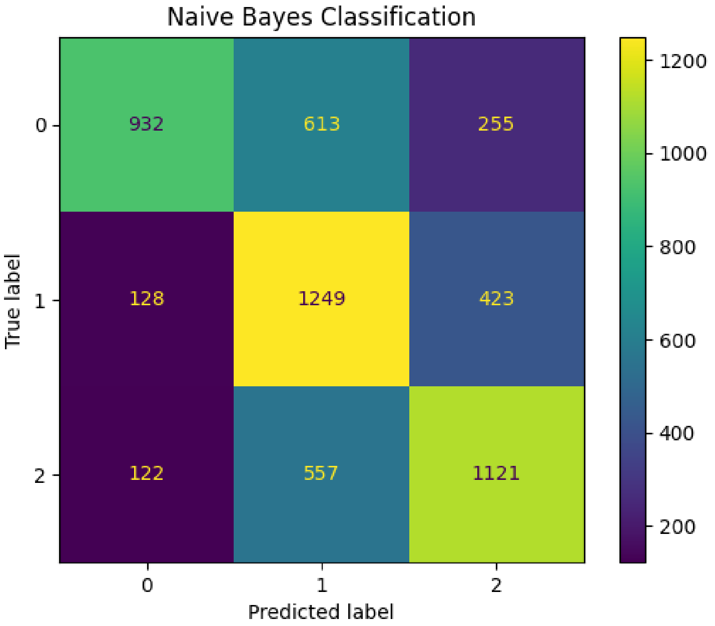

*Figure 1: Gaussian Naïve Bayes confusion matrix*

### 4.1.2 Hold-Out vs. 5-fold Cross Validation K-Nearest Neighbors

Images in the training set were downsized to 32 x 32 pixels for the following analysis. Tuning the number of neighbors using grid search yielded K = 2, which was used for building the hold-out K-nearest neighbors classifier. Surprisingly, this classifier performed slightly worse than the Naïve Bayes, with an overall accuracy of 0.60. Class 2 has notably higher recall than the other two classes—out of all 1,800 images actually generated by StableDiffusion, this KNN classifier was able to correctly predict 81% of them. Compared to the Naïve Bayes classifier above, hold-out KNN was better at correctly identifying images belonging to the StableDiffusion generator than the other two AI art generators.

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| 0 | 0.63 | 0.49 | 0.55 |
| 1 | 0.70 | 0.50 | 0.58 |
| 2 | 0.54 | 0.81 | 0.65 |
| **Accuracy** | | | 0.60 |

*Table 3: Hold-out KNN (K=2) classification report*

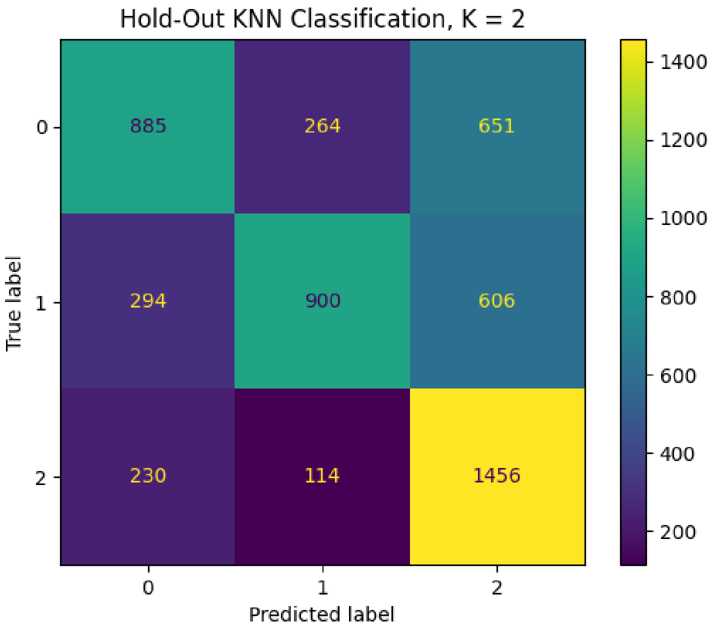

*Figure 2: Hold-out KNN (K = 2) confusion matrix*

To assess how K-nearest neighbors would perform on unseen data, the mean cross-validation score for 5-fold cross-validation using K = 2 was calculated. The mean cross-validation score was 0.58, which demonstrates that this classifier would still be better than randomly guessing to identify AI art generators, but is somewhat worse than using Gaussian Naïve Bayes at this point. To improve classification accuracy, the KNN model will be revisited after applying linear dimensionality reduction to project the training observations to a lower dimensional space.

### 4.1.3 Random Forest

The random forest classifier performed exceptionally well at labeling each of the images in the test dataset. As can be seen in Table 4 and Figure 3, this classifier was able to identify every single one of the 1,800 DALL-E 2 images correctly as DALL-E 2. In addition, out of all 1,810 data points predicted to be generated by DALL-E 2, only 10 of them were not actually generated by DALL-E 2. Besides its exemplary performance on DALL-E 2 images, random forest still does very well on both Midjourney and StableDiffusion images. When it does misclassify an image from either Midjourney or StableDiffusion, it generally misclassifies that image as one belonging to the other generator (e.g. Midjourney as StableDiffusion, or StableDiffusion as Midjourney). This begs the question: What distinctive features of images generated by DALL-E 2 is the random forest algorithm identifying? Further exploration of distinctive generator patterns in §4.2 will answer this question.

| **Class** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| 0 | 0.99 | 1.00 | 1.00 |
| 1 | 0.87 | 0.85 | 0.86 |
| 2 | 0.85 | 0.87 | 0.86 |
| **Accuracy** | | | 0.91 |

*Table 4: Random Forest classification report*

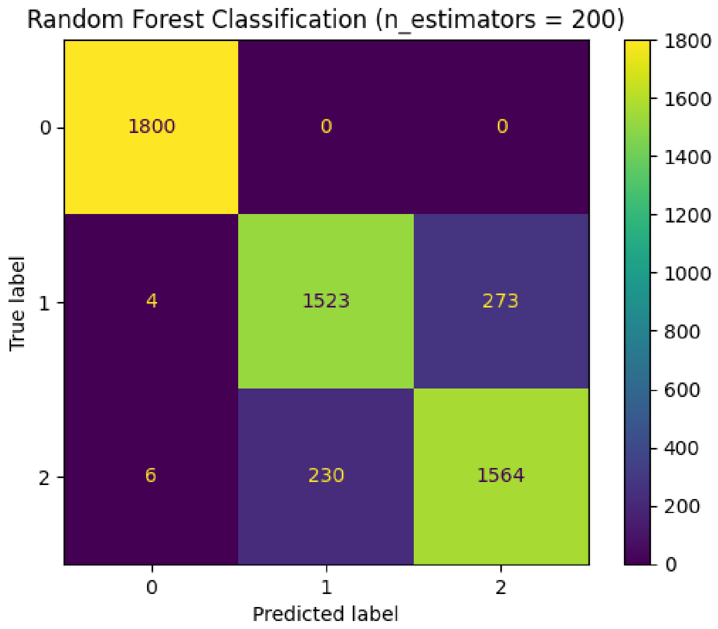

*Figure 3: Random Forest confusion matrix*

With consideration of the computational resources required to tune the number of trees and number of features for each split, the performance of a random forest model using sci-kit learn's default parameters for `RandomForestClassifier()`is almost the same as the results gained from the best estimator model obtained from a grid search, but runs slightly faster.

### 4.1.4 Radial Kernel Support Vector Machine

The support vector machine model using a radial kernel also performed very well, but not as well as the random forest classifier. This classifier achieved an overall accuracy of 0.84. Class 1 (Midjourney) had the highest precision score at 0.91, whereas Class 0 had the highest recall score at 0.90. Like the random forest model above, the support vector machine classifier appears to best at correctly classifying DALL-E 2 images, and would otherwise generally misclassify DALLE-2 images as StableDiffusion images.

For class 1, its high precision indicates that out of 1,513 images predicted to be generated by Midjourney, 1,378 were actually generated by Midjourney. However, out of the 1,800 images that were actually generated by Midjourney, the support vector machine model only correctly classified 77% of them. Those that were misclassified tended to be misclassified as StableDiffusion instead. Misclassification of StableDiffusion-generated images tended to be about the same across the other two generators (approximately 8.3% misclassified as DALL-E 2 versus 6.7% misclassified as Midjourney).

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| 0 | 0.84 | 0.90 | 0.87 |
| 1 | 0.91 | 0.77 | 0.83 |
| 2 | 0.78 | 0.85 | 0.81 |
| **Accuracy** | | | 0.84 |

*Table 5: Radial Kernel SVM classification report*

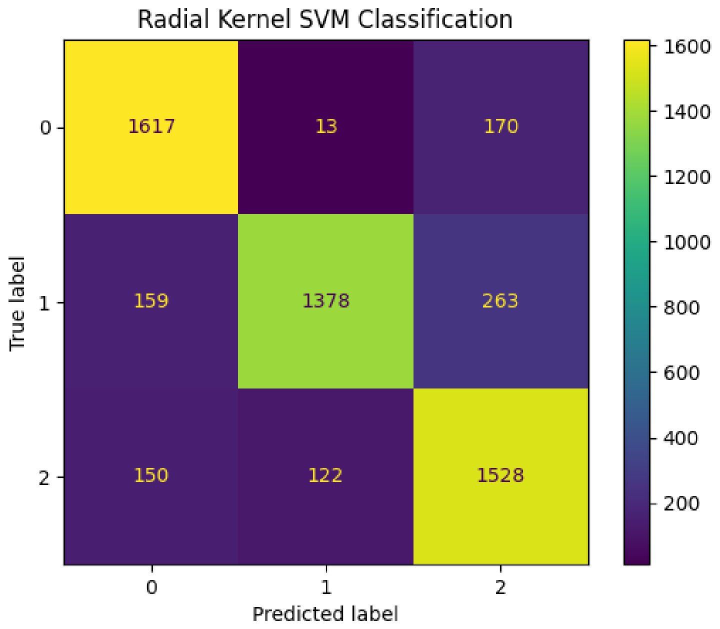

*Figure 4: Radial Kernel SVM confusion matrix*

### 4.2 Principal Component Analysis

Principal component analysis was used to reduce the dimensionality of the training data. A scree plot of the explained variance ratio (Figure 5) highlights how a reasonable number of principal components can be selected and used to transform the original data.

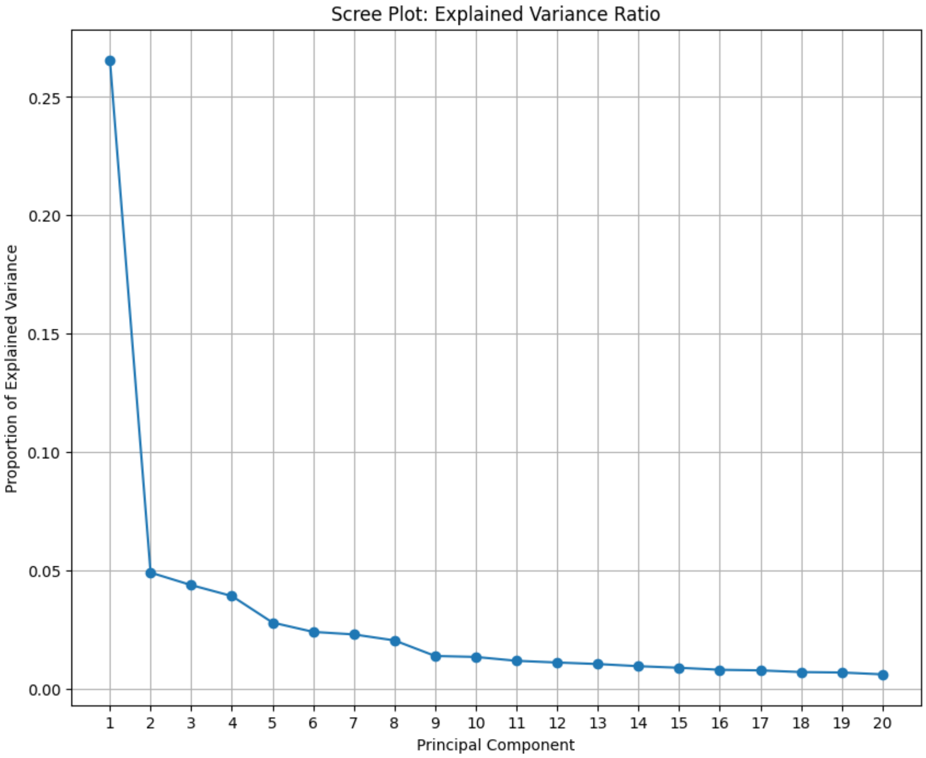

*Figure 5: Scree plot of the proportion of variance explained*

Selecting the first ten principal components captures 52% of the variance, and thus the following plots in Figure 6 illustrate the image patterns captured by each principal component.

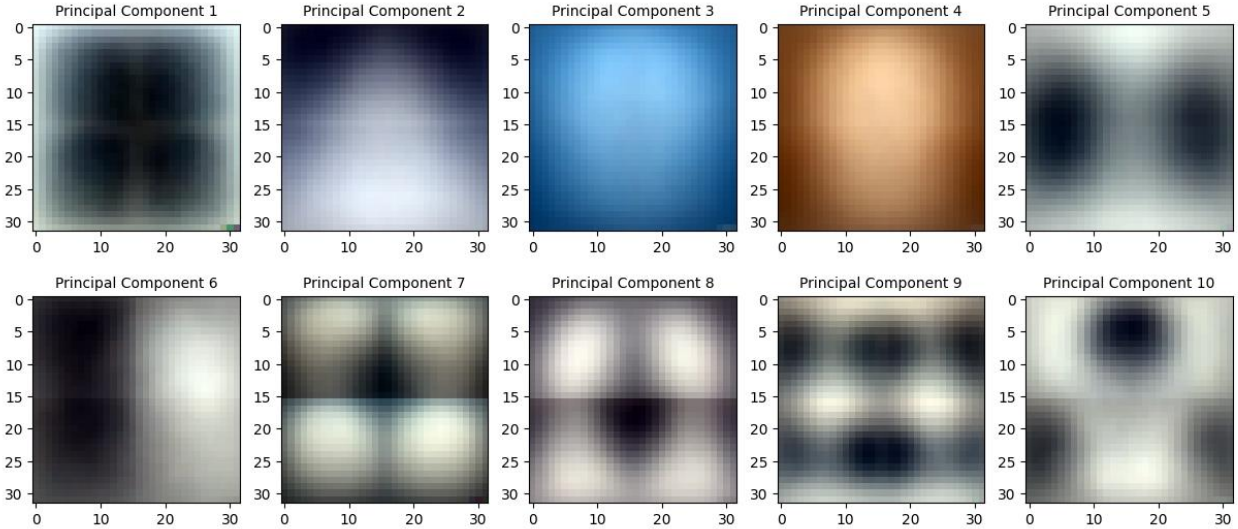

*Figure 6: Plots of the first ten principal components*

A careful examination of Principal Component 1 reveals an array of yellow-green-pink pixels, resembling a watermark in the bottom right corner of the plot. Other notable patterns revealed in Figure 6 include the partitioning of images into one, two, or four quadrants, curtain-style lighting and shading, the use of cool versus warm colors, or the relative positions of dark versus light elements in the image to distinguish between images by different generators. A quick initial inspection (by human eyes) of the original DALL-E 2, Midjourney, and StableDiffusion images would reveal that a watermark does indeed exist on seemingly all the images produced by DALLE-2, and images generated by Midjourney appear to be more frequently separated into four quadrants.

Principal component analysis was able to extract the watermark pattern from the DALL-E 2, which in conjunction with the proportion of variance explained in Figure 5, clearly shows that the first principal component explains over 25% of the variance in the dataset. Considering that images generated by DALL-E 2 make up exactly one-third of the dataset, an increase in K-nearest neighbors classification performance using a PCA-transformed dataset is expected.

### 4.2.1 Revisiting KNN and SVM

A second attempt was made at building a K-nearest neighbors classification model by transforming the training data using the top ten eigenvectors found via principal component analysis. For this model, the optimal number of neighbors was chosen via grid search to be 14, and the resulting classifier showed an increase in prediction accuracy from 0.60 originally to 0.67 using the same method on PCA-transformed data and K = 14.

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| 0 | 0.73 | 0.51 | 0.60 |
| 1 | 0.71 | 0.69 | 0.70 |
| 2 | 0.60 | 0.79 | 0.68 |
| **Accuracy** | | | 0.67 |

*Table 6: KNN after PCA (K=14) classification report*

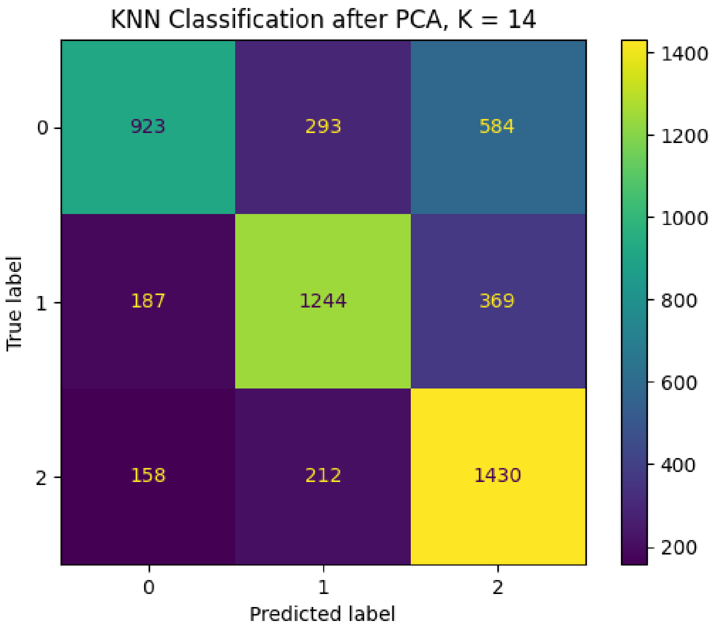

*Figure 7: KNN after PCA (K = 14) confusion matrix*

Reducing dimensionality seems to have greatly improved the performance and runtime of the K-nearest neighbors algorithm, which makes sense as KNN is susceptible to the curse of dimensionality. Creating a second support vector machine classifier using the same PCA-transformed training data ultimately did not improve its performance, although it did greatly reduce computation time. This is not unexpected, as support vector machines are known to be effective in high dimensional spaces as the method has measures against overfitting in the selection of its support vectors. Further investigation into the trade-off between prediction accuracy and computational intensity may benefit those who want to train support vector machine models for image classification.

| **Class** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| 0 | 0.72 | 0.59 | 0.65 |
| 1 | 0.75 | 0.60 | 0.67 |
| 2 | 0.60 | 0.82 | 0.69 |
| **Accuracy** | | | 0.67 |

*Table 7: SVM after PCA classification report*

## 5. Conclusions

### 5.1 Classifying Images by “AI Artist”

Ultimately, given the results of the various classifiers, random forest and radial kernel support vector machines can reliably attribute AI-generated images to their corresponding “AI artists.” The random forest classifier could identify DALL-E 2 images with a 100% true positive rate and near-100% precision, while the other two image generators Midjourney and StableDiffusion can be differentiated reasonably well using either random forest or radial kernel SVM. For effective use in the real world (as artists pretending to create their own works would presumably remove any obvious AI-art watermarks), further exploration must be done by removing the watermark in DALL-E 2 images or separating Midjourney or Stable Diffusion quadrant images into four separate pictures for analysis.

### 5.2 Inherent Patterns or Features

Visual inspection of the top ten principal components reveals several key distinguishing patterns that represent more than half of the variation amongst the training dataset. The most telling signs from images produced by DALL-E 2, Midjourney, or StableDiffusion can be broken down into the existence of watermarks, partitioning of images into blocks or quadrants, the use of cool versus warm tones across the image, and the relative positioning of the subject compared to its background. In particular, the existence of the watermark at the bottom-right of DALL-E 2 generated images contributed to over 25% of the explained variance, and clearly was the distinguishing factor for the random forest and support vector machine classification models. A deeper investigation into color tone, lighting, and positioning of subjects versus backgrounds will need to be done to effectively differentiate between AI-generated art after watermarks and partitioning are removed.

### 5.3 Standardizing Text Prompts

Discovery of the DALL-E 2 watermark and Midjourney image partitioning in the training data leads to the redundancy of classifying images created by AI generators using the same text prompt. For analysis that answers the third project objective to be meaningful, again the watermark must be removed, and Midjourney images must be separated into four individual pictures. Future study or advancement of this project will be needed to fully answer this question, and the study should be done again for the first question as well.

## References

*DALL·E 2 Largest image database*. (2023). Dalle2.Gallery. https://dalle2.gallery/#search

*Midjourney User Prompts & Generated Images (250k)*. (n.d.). Www.kaggle.com. Retrieved April 2, 2023, from https://www.kaggle.com/datasets/da9b9ba35ffbd86a5f97ccd068d3c74f5742cfe5f34f6aaf1f0f458d7694f55e

Penava, E. (2023, January 24). *AI Art Is in Legal Greyscale | The Regulatory Review*. www.theregreview.org. https://www.theregreview.org/2023/01/24/penava-ai-art-is-in-legal-greyscale/

Wang, Z. J., Montoya, E., Munechika, D., Yang, H., Hoover, B., & Chau, D. H. (2022, October 26). *DiffusionDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models*. ArXiv.org. https://arxiv.org/abs/2210.14896