Paint by Generative Adversarial Network (GAN)

Using Neural Networks to Classify Artistic Styles and Generate Novel Artworks

Vanessa Grass
M.S. Data Science
University of New Haven
West Haven, CT

Abstract* — This paper investigates the use of Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) in the classification and generation of artistic images. The study begins with the development of a CNN model designed to distinguish between paintings by Paul Cezanne and Vincent van Gogh, utilizing a dataset of over 1,000 artworks. Through exploratory data analysis, baseline modeling, and iterative refinements, the CNN achieves an accuracy of 80% in this binary classification task. Following the classification, the focus shifts to generating new artwork using a GAN, aiming to produce images that emulate the styles of Cezanne and van Gogh. Initial attempts face challenges with noisy outputs, but the discussion includes the potential of advanced GAN architectures, such as Wasserstein GANs (WGANs), to enhance the quality of generated images. The findings highlight both the strengths and limitations of neural networks at the intersection of AI and art, suggesting future directions for improving generative models in creative applications.

I. INTRODUCTION

Generative Adversarial Networks (GANs) utilize both generative and discriminative models. Generative models describe the process by which data is generated, framed within a probabilistic model. Specifically, a generative model seeks to generate the distribution of P(x|y), where x represents an observation in the dataset and y is a class label. In contrast, discriminative models focus solely on categorizing a given datapoint without concerning themselves with the data generation process. Discriminative models aim to find the distribution of P(y|x), which represents the probability of a class label y given some data x. Essentially, discriminative models define the decision boundary between classes, while generative models offer broader applications, such as imputing missing data or generating new data.

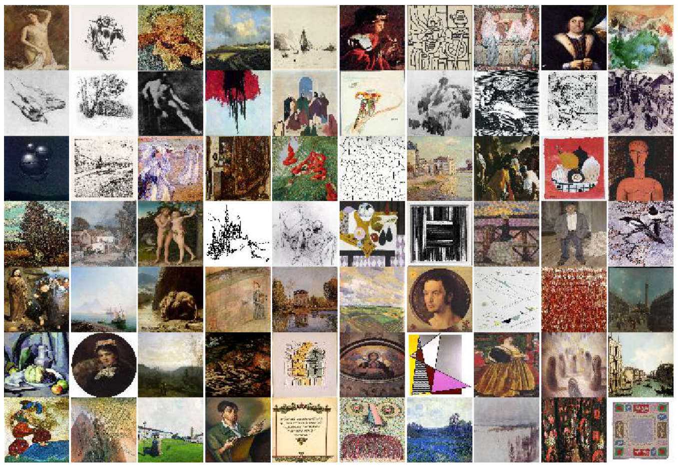
Figure 1. Examples of images in the “Painter by Numbers” dataset

In the context of a GAN, the generator employs a generative model, taking random input values and transforming these into images through a deconvolutional neural network. While the image generation aspect of a GAN is highly intriguing, the role of the discriminator, which utilizes a discriminative model, is equally essential. The discriminator functions as a binary classifier, determining whether an image produced by the generator resembles a real image from the original dataset or an artificially created one. This discriminator is typically implemented as a standard convolutional neural network (CNN).

Throughout the GAN’s training process, the weights and biases in both the discriminator and generator are updated through backpropagation. This enables the discriminator to learn how to distinguish real images from those generated by the generator. Concurrently, the generator utilizes feedback from the discriminator to refine its output, producing increasingly convincing fake images that the discriminator eventually cannot differentiate from real ones.

The initial focus of this research involved developing a comprehensive understanding of GAN architectures, with the ultimate goal of constructing a GAN capable of generating novel art from images of historical paintings. A foundational step in this process involved building a standard CNN to solve a binary classification problem: distinguishing between paintings created by Paul Cezanne and Vincent van Gogh. The underlying premise of a CNN is to process an input image through a series of layers that recognize increasingly complex features, ultimately outputting a single classification—whether the image is a Cezanne or van Gogh painting. This approach serves as a valid starting point, given its architectural similarity to the discriminator in a GAN, which also addresses a binary classification problem by distinguishing real from fake images. The initial steps in creating this binary classification CNN included conducting exploratory data analysis, establishing a non-neural network baseline model, prototyping a simple CNN, and ultimately building a more complex CNN capable of differentiating between Cezanne and van Gogh paintings.

II. DATASET AND EXPLORATORY DATA ANALYSIS

The dataset utilized in this project originates from the Kaggle competition “Painter by Numbers,” with the majority of images sourced from wikiart.org. The dataset consists of a total of 103,250 images, with 79,433 labeled as training data and 23,817 as testing data. The dataset encompasses 42 unique genres, including portraits, landscapes, and abstract works. Additionally, 135 distinct painting styles are represented, ranging from Impressionism and Expressionism to Realism, among others. The dataset includes works from 2,074 different artists.

For the purposes of this project, the focus was narrowed to the works of Paul Cezanne and Vincent van Gogh. The rationale behind this selection lies in the similarities these artists share in terms of color and content, despite differences in style. Both artists are recognized as Post-Impressionists, making them suitable subjects for this analysis.

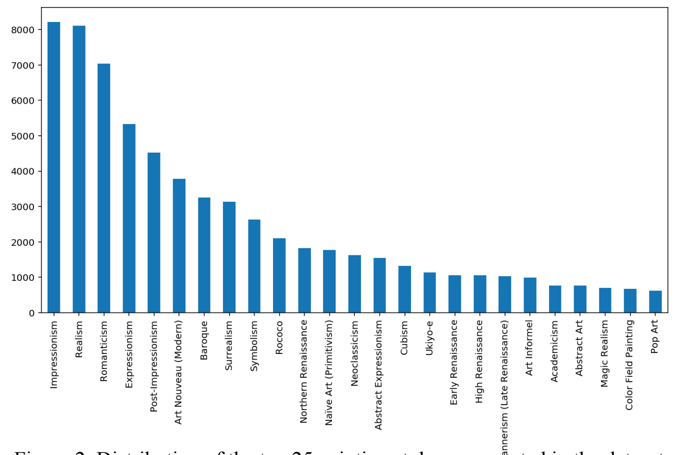

Figure 2. Distribution of the top 25 painting styles represented in the dataset

III. DATA PREPROCESSING

The images in the Kaggle painting dataset exhibit significant variation in pixel size and aspect ratio, ranging from 30,000 by 29,605 pixels on the larger end to 283 by 558 pixels on the smaller end. To make the image data suitable for machine learning applications, a script was developed to resize all images to a uniform square dimension of 72 by 72 pixels. While this resizing approach can cause distortions in non-square images, it aligns with standard practices in similar computer vision tasks. Alternative methods, such as cropping, could potentially mitigate these distortions, though resizing remains a commonly employed solution.

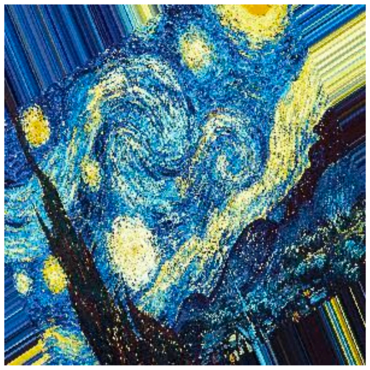

Figure 3. Example of the random transformations applied to dataset images

For the purposes of binary classification, the original dataset of 103,250 images was subsetted to include only the works of Paul Cezanne and Vincent van Gogh. Following the training and testing split provided by Kaggle, this subset resulted in 804 training images (412 Cezanne paintings and 392 van Gogh paintings) and 189 testing images (87 Cezanne paintings and 102 van Gogh paintings). The differing number of images per artist introduces a class imbalance, which is minimal in the training set but more pronounced in the testing set. This issue is addressed in further detail in the modeling sections.

Additional preprocessing steps involved normalizing pixel intensity values to a range of 0 to 255, achieved either through simple NumPy vectorized division or by utilizing the Keras ImageDataGenerator class. The ImageDataGenerator class was also employed to augment the painting images through random transformations, which typically enhance model generalization by introducing variations in orientation, rotation, and other aspects. However, the appropriateness of applying such transformations to this specific dataset warrants further investigation, as these alterations may affect the CNN's ability to accurately discern the artist's style. Specifically, the fill_mode argument in ImageDataGenerator, which determines how empty pixels caused by transformations are filled, can lead to a streaking effect in images when set to "nearest." Using the "reflect" option is likely more suitable, as it avoids this streaking effect, which could be mistakenly interpreted as an artistic style by the CNN.

IV. BASELINE "GENERATIVE" MODEL

Given the project's objective to generate images, a baseline "generative" model was developed by averaging images to create composite representations of data subsets. The analysis revealed that averaging all images within a specific painting category or grouping typically results in a very fuzzy image. There is likely an optimal point at which averaging a certain number of similar images would yield a more refined result. Future experiments may explore the use of KMeans clustering to group similar images within higher-level categories, such as genre, to enhance this approach. Notably, averaging all portrait paintings reveals a discernible pattern, typically showing a central area where faces likely appear and a circular or oval frame common in certain art history periods, such as the Italian Renaissance. However, averaging landscape, Impressionism, Cezanne, and van Gogh paintings results in images that are considerably muddied. This outcome was anticipated, confirming that averaging images serves as a poor baseline "generative" model.

For the baseline classification model, a Random Forest classifier was trained to distinguish between paintings by Cezanne and van Gogh. The default parameters of the scikit-learn implementation were used, yielding a model with an accuracy of 0.65, which is above random guessing. However, given the class imbalance in the data, accuracy is not the most reliable evaluation metric, so the f1-score was also considered, which was similarly 0.65.

An error analysis was conducted to examine which paintings were most easily classified, which were most frequently misclassified, and which caused the most confusion for the model. The Random Forest classifier exhibited a tendency to over-predict Vincent van Gogh, despite the presence of slightly more Cezanne paintings (~20) in the training set. A preliminary examination suggests that the presence of blues and reds may cause the model to mistakenly classify a painting as a van Gogh. Feature importances returned by the Random Forest classifier initially appeared to be of interest, particularly in identifying significant pixel locations within the image. However, the importances are generally low, with approximately 10 locations emerging as slightly more important than others. These pixel locations are suspected to be near the center of the image. Upon further consideration, this interpretation reveals limitations, as focusing on precise pixel values may be too granular to generalize effectively across different images.

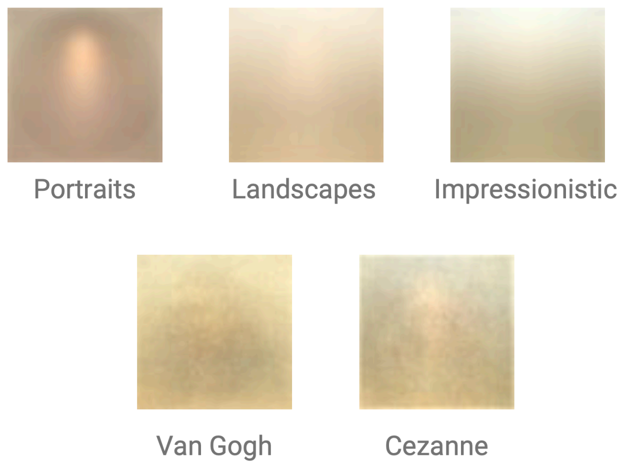

Figure 4. Composite images generated by the baseline "generative" model, illustrating the resulting fuzziness and muddied patterns when averaging subsets of paintings

V. CONVOLUTIONAL NEURAL NETWORK (CNN) MODELS

A. Simple CNN

A simple Convolutional Neural Network (CNN) was implemented and initially overfitted on a small subset of the data, consisting of three randomly selected paintings by Cezanne and van Gogh. This approach facilitated rapid prototyping of the CNN architecture within the dataset. The architecture of the simple CNN comprises a single convolutional layer with 32 filters, utilizing a Rectified Linear Unit (ReLU) as the activation function, followed by a max-pooling layer, a flatten layer, and a final dense layer with a sigmoid activation function, appropriate for the binary classification task. The model was compiled using binary cross-entropy as the loss function and RMSprop as the gradient descent optimizer. The model's accuracy stabilized at 1.0 after approximately seven epochs.

Subsequently, the model was trained using the entire dataset for 50 epochs, resulting in an accuracy of 0.70 and an f1-score of 0.68. This performance was only marginally better than that of the Random Forest classifier. To explore the impact of class imbalance on model performance, the larger class was randomly sampled to match the size of the smaller class. Training the simple CNN on this balanced dataset yielded an accuracy and f1-score of 0.71, demonstrating only a slight improvement over the imbalanced dataset.

B. "Deeper" CNN

Building upon the simple CNN architecture, additional layers were introduced to create a deeper model. Three more convolutional and max-pooling layers were added, initially maintaining 32 convolutional filters and then doubling the filter count for the subsequent two convolutional layers. ReLU activation continued to be used, and an additional flatten layer and a dense layer with 128 units (neurons) were incorporated. A dropout layer, set at 0.25, was also employed to mitigate potential overfitting by randomly deactivating neurons within a given layer. The final activation remained sigmoid, and the model was compiled with binary cross-entropy as the loss

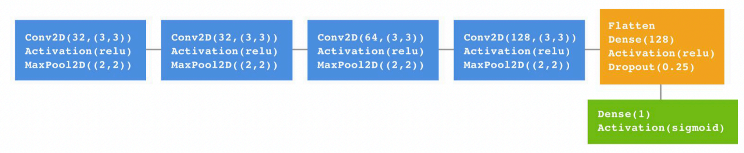

Figure 5. Architecture of the deeper convolutional neural network (CNN) model for the classification task

function.

Although grid search was not performed, manual experimentation with various hyperparameters was conducted, including adjustments to the number of hidden layers and testing different activation functions, such as ELU and Hard Sigmoid. However, these changes did not yield significant improvements in accuracy or f1-score. Notably, using the Adam optimizer for gradient descent resulted in a higher f1-score during experimentation. The diagram below provides an overview of the full architecture.

The model was trained for 100 epochs on a Google Cloud instance equipped with 16 vCPUs, 104 GB of memory, and 2 NVIDIA Tesla K80 GPUs, utilizing the image provided by Stanford's CS231n course. Extending the training beyond 100 epochs did not yield any improvement in accuracy and even introduced the risk of overfitting. During the evaluation of the model's performance, an important detail in Keras was encountered when using the .flow_from_directory method on an ImageDataGenerator object—the class labels are mapped to indices in alphanumeric order. This initially led to an error in decoding the probabilities and class labels on the validation set, as van Gogh had been arbitrarily labeled as class 0 and Cezanne as class 1. Consequently, the evaluation metrics were inverted due to the switched class labels. After resolving this issue, the model achieved an accuracy of 0.80 with an f1-score of 0.76, representing an improvement over the previously mentioned simple CNN.

Error analysis revealed no significant anomalies, though there were a few instances where the model confidently predicted a painting as Cezanne's work when it was actually by van Gogh. These instances highlighted the stylistic similarities between the two artists. Further validation of the model was

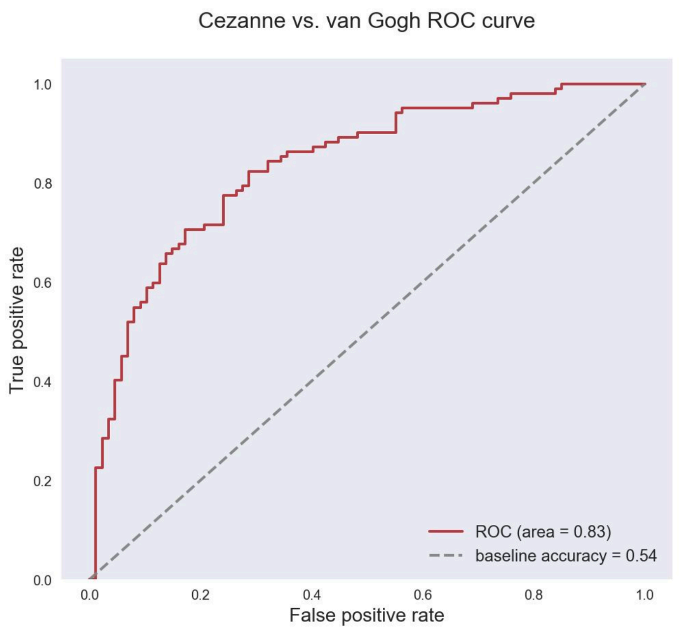

Figure 6. Receiver Operating Characteristic (ROC) curve illustrating the performance of the CNN model in distinguishing between Cezanne and van Gogh paintings

conducted using two additional unseen images that were not actual paintings by either artist but rather "fake" images generated by applying the artist's style to photos through neural style transfer. When testing a canonical but "fake" Cezanne image, the model accurately labeled it as Cezanne with a probability of 0.17, which is appropriate given that a more confident prediction would result in a probability closer to zero, as Cezanne is labeled as class 0. The model also correctly identified a canonical but "fake" van Gogh, though with a lower margin of confidence (a probability of 0.53), indicating that further improvement is needed for more accurate predictions.

Considering the small dataset on which this model was trained, the overall performance is satisfactory. However, training on a larger set of paintings from each artist would likely enhance performance. An additional approach to consider is employing transfer learning, where a model is first trained on a broader subset of painting data, such as the top five genres or painting styles, and then the learned weights are leveraged to improve the binary classification of Cezanne and van Gogh. Another option for transfer learning could involve using larger pre-trained models, such as VGG-16 or Inception, although the effectiveness of these models with non-photo-realistic images remains uncertain.

VI. AN ATTEMPT AT A GENERATIVE ADVERSARIAL NETWORK (GAN)

The final component of this project involved an attempt to build a Generative Adversarial Network (GAN) capable of producing images that closely resemble paintings from the Kaggle painting dataset. A "vanilla" GAN architecture was selected for experimentation, as opposed to more advanced GAN architectures such as AC-GAN, StackGAN, or WGAN. In this vanilla GAN, the discriminator consisted of four convolutional layers, each employing leaky ReLU activation and generous dropout. The use of leaky ReLU is intended to prevent the issue of "saturated gradients," a common problem in GAN architectures. Saturated gradients occur when gradients, representing changes in model weights, become excessively small or large during backpropagation, effectively causing neurons to cease learning. Leaky ReLU addresses the "dying ReLU" problem by allowing small negative values.

Similar to the CNN used in the binary classification task, the discriminator in this GAN outputs a one-dimensional probability vector through a flatten layer and a dense layer with sigmoid activation. The generator, on the other hand, takes a uniformly distributed noise vector as input and generates fake images by essentially performing the inverse of convolution through several transposed convolutional (deconvolutional) layers. The generator also incorporates layers such as upsampling, which increases image dimensions, and batch normalization, which stabilizes learning by normalizing the activations of the previous layer. Like the discriminator, the generator also utilizes leaky ReLU and dropout, concluding with a flatten and dense layer with sigmoid activation. Both models were compiled using RMSprop as the optimizer and binary cross-entropy as the loss function. The generator and discriminator were then combined to create the GAN.

While this GAN architecture produced reasonable results on the MNIST dataset within 1,000 training steps, it failed to deliver satisfactory results on the Kaggle painting dataset. The generated images still resembled noise rather than coherent paintings. It is hypothesized that this issue is related to the aforementioned problem of saturated gradients, as leaky ReLU may be inconsistent in resolving this issue. Further exploration is required to confirm this hypothesis. One promising alternative architecture is the Wasserstein GAN (WGAN), which incorporates the Wasserstein metric—a distance function between probability distributions. This metric is believed to enhance the stability of GANs. In summary, the vanilla GAN architecture did not achieve the desired results, and future work

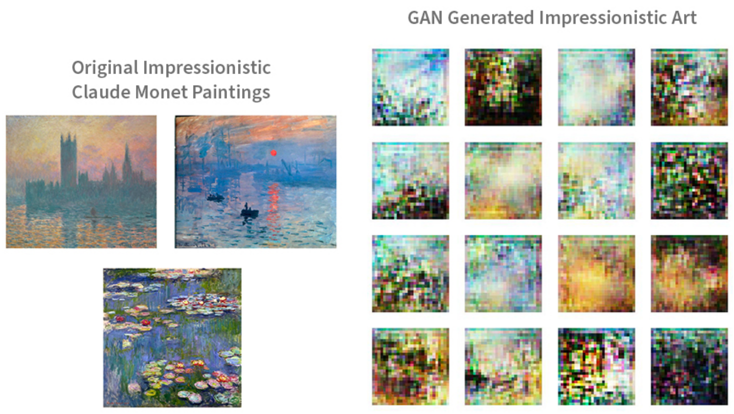

Figure 7. Comparison of original Claude Monet painting (left) and the corresponding GAN-generated Impressionistic version (right)

will involve investigating other GAN architectures, such as the WGAN, to improve image generation quality.

VII. CONCLUSIONS & FUTURE WORK

This research explored the application of Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) in the realm of artistic image classification and generation. The study successfully demonstrated the ability of CNNs to distinguish between paintings by Paul Cezanne and Vincent van Gogh, achieving a classification accuracy of 80% and an f1-score of 0.76. This performance, though satisfactory, highlighted the potential for further improvement, particularly through the use of larger datasets and advanced techniques such as transfer learning.

The experimentation with GANs, specifically a vanilla GAN architecture, revealed significant challenges in generating high-quality images from the Kaggle painting dataset. While the GAN produced reasonable results on simpler datasets such as MNIST, it struggled with the more complex and nuanced task of creating convincing artistic images. The primary issues identified include the problem of saturated gradients and the inconsistency of leaky ReLU in addressing these challenges. Despite these difficulties, the research underscores the potential of GANs in the field of generative art, pointing to the need for more robust and stable architectures.

Future work will focus on addressing the limitations encountered in this study. First, the exploration of advanced GAN architectures, such as the Wasserstein GAN (WGAN), is essential to overcoming the stability issues observed in the vanilla GAN. Additionally, expanding the dataset to include a broader range of artists and styles could enhance the CNN's ability to generalize across different artistic techniques. The incorporation of transfer learning, leveraging pre-trained models like VGG-16 or Inception, also holds promise for improving both classification and generative tasks. Further experimentation with hyperparameter tuning and different activation functions will be necessary to optimize model performance.

In conclusion, while the current models have shown promising results, there remains substantial room for enhancement. The intersection of AI and art is a rapidly evolving field, and continued research in this area will undoubtedly contribute to more sophisticated and creative applications of neural networks in the future.

REFERENCES

1. Chollet, F. (2016, August 5). Building powerful image classification models using very little data. Keras Blog. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
2. Karpathy, A. (2016). Commonly used activation functions. CS231n: Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/neural-networks-1/
3. Bhatt, A. (2017, August 16). GAN by Example using Keras on Tensorflow Backend. Towards Data Science. https://medium.com/towards-data-science/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a60d0
4. Ghosh, P. (2017, December 6). GANGogh: Creating Art with GANs. Towards Data Science. https://medium.com/towards-data-science/gangogh-creating-art-with-gans-8d087d8f74a1
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. arXiv. https://arxiv.org/pdf/1406.2661.pdf
6. Geitgey, A. (2017, March 10). Machine Learning is Fun Part 7: Abusing Generative Adversarial Networks to Make 8-bit Pixel Art. Medium. https://medium.com/@ageitgey/abusing-generative-adversarial-networks-to-make-8-bit-pixel-art-e45d9b96cee7