ArtSee: Art Style Classification Using A CNN

Asha Karmakar, Akhil Iyer, Teresa Luo

Abstract

With the proliferation of digital art due to the increased use of digital tools for self-expression, there has been a surge in the volume of artworks being produced, presenting a challenge for art analysis and categorization. In this study, we address this challenge by leveraging machine learning techniques to classify artworks by style, aiming to facilitate deeper analysis and recommendation systems in art-related platforms. Drawing inspiration from existing literature, we propose a neural network architecture that combines texture and color analysis paths to capture distinctive characteristics of artworks. We trained our network on the Kaggle WikiArt Dataset, focusing on four diverse art genres: Cubism, Late Renaissance, Color Field Painting, and Ukiyo-e. Before feeding these images into the network, we resized and padded the image to a fixed input. We achieved a test accuracy of 72.5%. Our evaluation reveals insights into the model's performance, including its strengths in distinguishing certain styles and areas for improvement. Additionally, we explore visualizations of learned features and feature attribution techniques to gain interpretability into the model's decision-making process.

1 Introduction

As access to self-expression through art has increased, the volume of artworks being produced overtime has increased, particularly in the digital realm. To better understand artwork on a surface level and therefore facilitate deeper analysis, we can use machine learning to classify artwork by style. Thus, the rationale for creating this model was so we can discover the distinctive characteristics of a particular artistic style as well as recommend artwork on art-related platforms, among other uses.

2 Related Work

Most neural networks engineered for the purpose of artwork style classification build upon 2D convolutional networks, and thus, we will opt to do the same.

Cetinic et al. adapted a pre-existing model, the CaffeNet model, created by Jia et al. in 2014 (which was a version of the AlexNet model created by Krizhevsky et al. in 2012) (Cetinic et al. [2018]). The CaffeNet model architecture has five convolutional layers followed by three fully connected layers. They utilize max pooling after the first, second, and fifth convolutional layer and dropout after the first and second fully connected layers. Their input was pre-processed from the full-size image to a $227 \times 227 \times 3$ matrix (3 channels for RGB coloring). They achieved this through resizing the original and cropping the resized image to a square. From this model, we will adopt the convolutional layers, max pooling, and fully connected layers. We will not be able to replicate their exact model, as we have far less computational power, and thus, will downsize while keeping the aspects of the architecture that are critical.

Zhong et al. also use a 2D convolutional network to classify artwork by style (Zhong et al. [2020]). They build upon a simple convolutional network by introducing two channels of convolutional layers. The RGB channel is used to extract patterns in the elements of the image (with the input to the channel being the image matrix) and the brushstroke channel is used to extract patterns in the brush strokes/texture of the image (with the input to the channel being a gray-level co-occurrence matrix, or GLCM). We would like to utilize this technique, but unfortunately do not have the same computational power. Rather than generating all of the GLCMs, we will feed the image into both channels and have them pick up on separate patterns in the image.

3 System

3.1 Dataset

We used the Kaggle WikiArt Dataset (Stefano Morelli [2022]). This dataset has 27 art genres with thousands of images per genre. The total size of the dataset was around 35 GB, which was too large for us to work with, so we chose four genres that we thought represented a good variety of artistic qualities and time periods.

- **Late Renaissance** (1520–1600): human figures in a realistic style
- **Ukiyo-e** (1615–1868): Japanese woodblock prints with detailed figures and landscapes
- **Cubism** (1907–1914): abstract shapes and also distorted representations of objects/figures
- **Color Field Painting** (1950–1960s): minimal details, large swaths of color and abstract shapes

3.2 Data Preprocessing

We began by randomly selecting 1100 images from each of the four listed genres. Then, we shuffled the images and applied an 80-20 train/test split (this code was mostly provided by ChatGPT). The image processing occurred in real-time during training and testing using PyTorch's `transforms.Compose` functionality. Each image was resized (while maintaining aspect ratio) and padded with a reflection of itself to ensure a uniform size of 512 x 512 pixels when fed into the model. We chose to pad the image with itself instead of using a constant value so that the elements of the painting are maintained, and the model learns the actual features of the artwork instead of factoring the aspect ratio into its decision. We used ChatGPT to get the code for the PyTorch Dataset and DataLoader objects, and to get us started with the `transforms.Compose` functionality, but the custom code for padding (`SquarePad` function) was written by us.

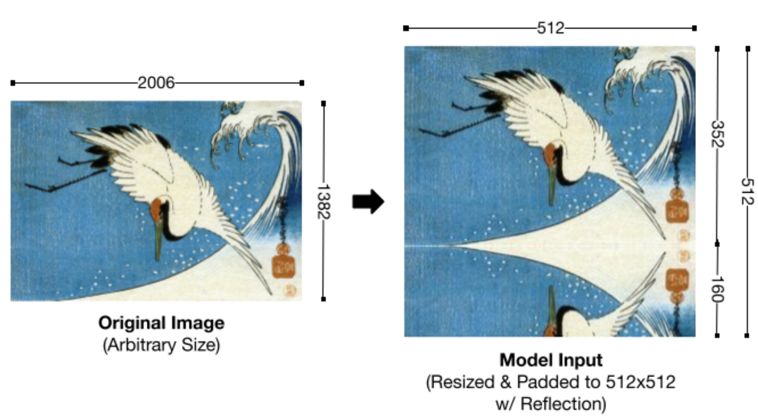


Figure 1: Transformations performed during image preprocessing.

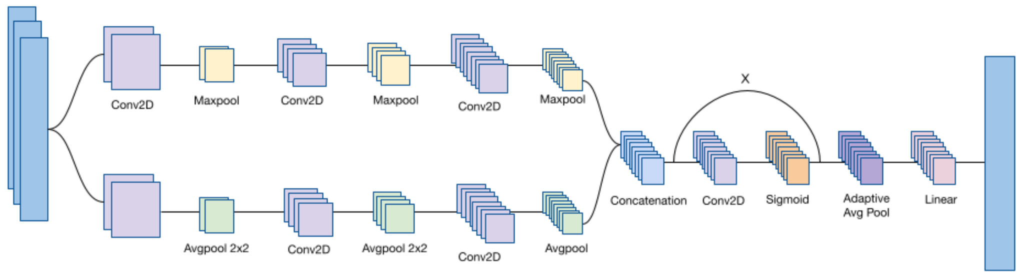


Figure 2: Model Architecture

3.3 Model Architecture

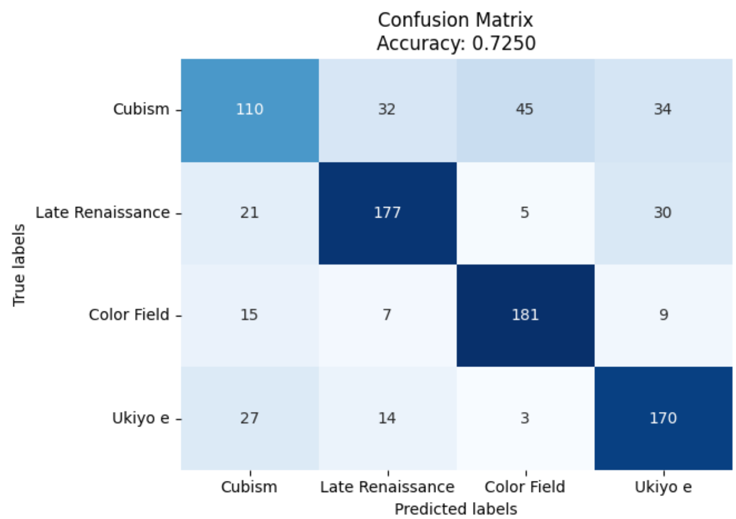
Convolutional Kernel Information	Number of Kernels	Kernel Size	Stride
Color/Texture Path Conv1	16	6x6	1
Color/Texture Path Conv2	32	3x3	1
Color/Texture Path Conv3	64	3x3	1
Texture Path MaxPool	-	2x2	2
Color Path AvgPool	-	2x2	2
Conv4	64	1x1	1
AdaptiveAvgPool	-	1x1	-

We designed our model to understand two main characteristics of any artwork: art textures, which include brushstroke lines, and color usage. To accomplish this, we create two paths in the network: the texture path that puts greater attention on edges and contours, and the color path that puts greater attention to colors in the image (we finalized this idea with the help of ChatGPT). For the color path, we use three pairs of convolution and average pooling layers since this produces output representative of overall color usage in the image patches. For the texture path, we use three pairs of convolution and max pooling layers since these pooling layers downsample information while maintaining emphasis on lines and edges in each image patch. For both of these paths, we downsample information through the use of smaller kernels from 6×6 convolution kernels to 3×3 convolution kernels. Additionally, we accomplish this also by using pooling layers with 2×2 kernels. In between the convolution layers, we used ReLU activation functions to better avoid vanishing gradients. We combine the outputs from both paths through concatenation into a single tensor and then use an attention mechanism implemented with a convolution layer of 1×1 kernels and sigmoid activation function. These weights are then multiplied with the concatenated tensor to put greater attention on the most relevant features from each path. We finally feed the output from our attention mechanism into an adaptive average pooling layer and a linear layer to generate four-dimensional output vectors for each image in the batch.

4 Evaluation

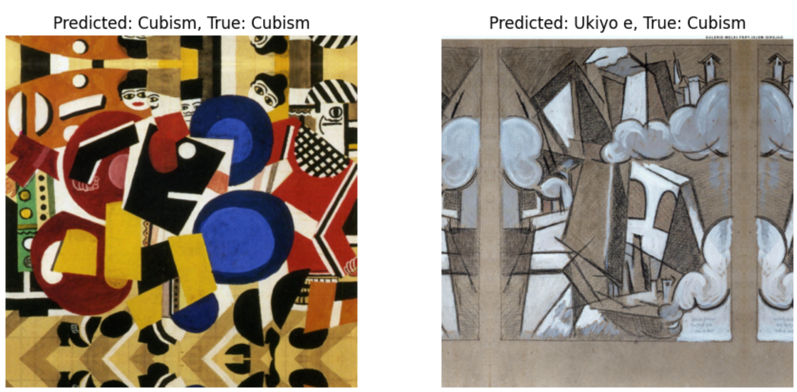
4.1 Overall Results

The model had a test accuracy of 72.5%. The confusion matrix (shown below) suggests that the model does not have heavy class imbalance in its predictions. The model predicts color field painting artwork correctly the most compared to the other classes. This is because the color field painting artwork is the most distinct style relative to the other styles. Unlike the other styles, color field painting is composed of jagged blocks and uses a different color palette, allowing the model to easily distinguish this artwork style. However, there are still instances in which the model confuses color field painting with other styles.



4.2 Examples of Correct/Incorrect Classifications

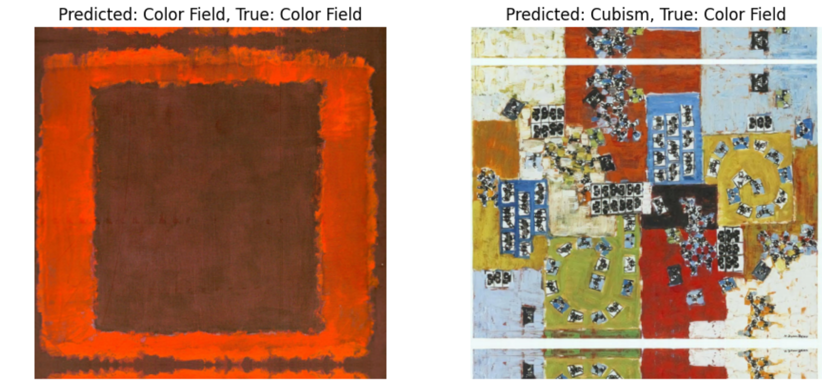
Typically, the model will misclassify an image if it contains characteristics commonly present in other styles. In all of the misclassified examples, we can identify features of other styles that may have led the model to misclassify the image.



(a) Correctly Classified (b) Incorrectly Classified

Figure 3: Correct/Incorrect Classifications for Cubism

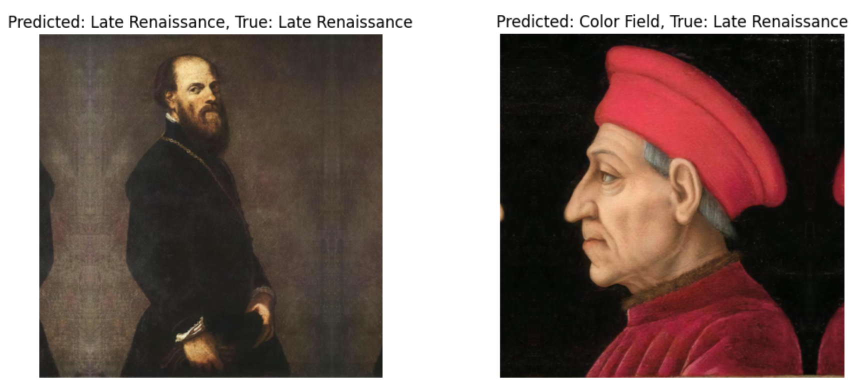
Cubism typically contains jagged blocks of uniform color to create abstractions of figures as shown in Figure 3. As a result, Cubism shares a lot of attributes with other styles and thus is the most incorrectly classified class. In the correctly classified Cubism painting, we can see the more complex objects with jagged shapes. In the misclassified Cubism painting, it features a color palette commonly used in Ukiyo-e paintings, with light blues and a beige background. The color palette, combined with the positioning of the central objects placed on a beige border/background, likely led the model to classify it as Ukiyo-e.



(a) Correctly Classified (b) Incorrectly Classified

Figure 4: Correct/Incorrect Classifications for Color Field Painting

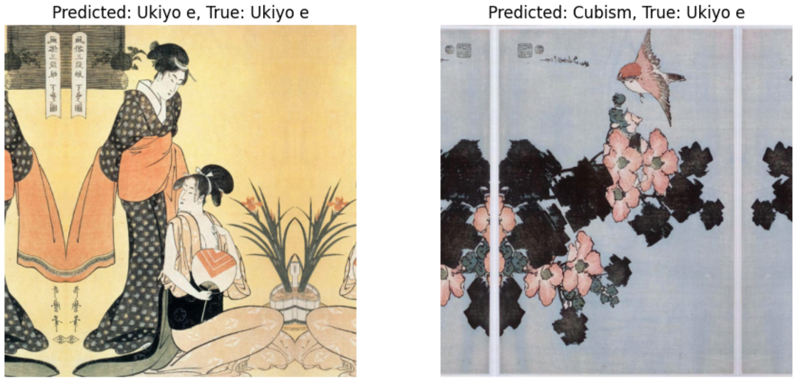
Color Field Painting typically contains large blocks of color in regular, uncomplex shapes as shown in Figure 4. The correctly classified Color Field Painting features nested squares of solid shades of red. The painting that the model misclassified has more complex color-block objects with more jagged shapes, a common element of Cubism.



(a) Correctly Classified (b) Incorrectly Classified

Figure 5: Correct/Incorrect Classifications for Late Renaissance

Late Renaissance typically features darker, more muted and somber color palettes than the other styles as shown in Figure 5. Compositionally, it will highlight a couple of central figures in either a landscape or a plain background. In the incorrectly classified Late Renaissance painting, the cap and background are both large blocks of untextured color. Typically, Late Renaissance paintings will feature more texture and shading in their large blocks of color. However, this painting has relatively little texture, likely leading the model to believe that it is a Color Field Painting.



(a) Correctly Classified (b) Incorrectly Classified

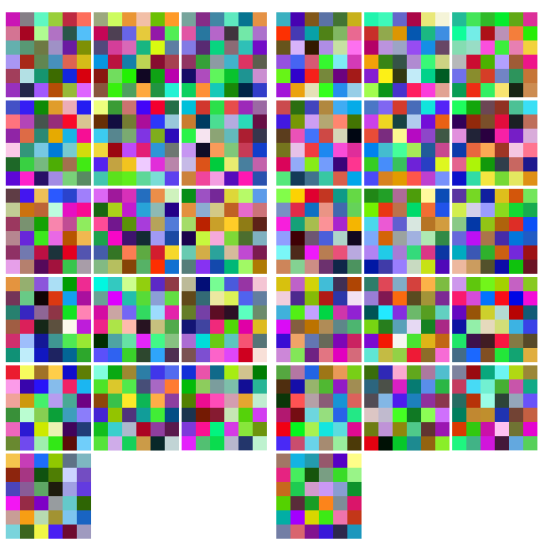
Figure 6: Correct/Incorrect Classifications for Ukiyo-e

Ukiyo-e typically features central figures depicted using line art, strongly defining the objects in the painting as shown in Figure 6. In the correctly classified Ukiyo-e painting, two central figures are depicted using strong, black brush strokes on a solid, cream background. In the incorrectly classified Ukiyo-e painting, the composition is shared with other Ukiyo-e paintings, however, the figures of the flowers are solid and jagged. This style of depicting the flowers is reminiscent of the jagged blocks in Cubism, likely leading the model to classify it as Cubism.

4.3 Experiments

4.3.1 Kernel Visualization

To visualize what features of the artwork the kernels in our first convolutional layer are detecting, we will display their learned values. We hope to see specific kernel patterns common across convolutional kernels (such as edge detection of various orientations or color detection in certain patterns). Below are the visualizations of the kernels on the first convolutional layer of our two convolutional paths.



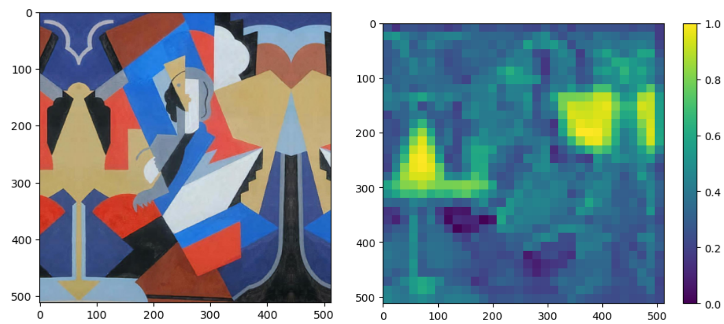
(a) Path 1 Kernels (b) Path 2 Kernels

Figure 7: Kernels of the first convolutional layer.

The kernels do not seem to be picking up on any discernible patterns, but rather, seem random and are hard to interpret. We believe this is because our input images are 512 x 512 while our kernels are 6 x 6. Thus, relative to the features in the image, our kernels are not large enough to pick up on specific edges or color patterns. If our kernels were larger, they would likely be more interpretable and may even improve the accuracy of our model, as it could pick up on actual features rather than pixel by pixel patterns.

4.3.2 Attribution

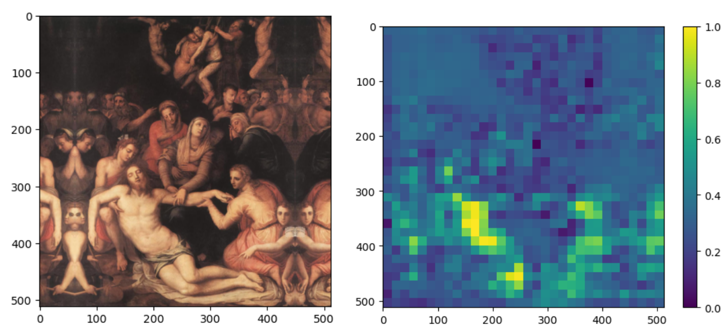
What specific artistic features differentiate certain art styles from others? We would like to visualize which parts of the image the model uses to make its prediction about the style of the input image. We will use perturbation, a feature attribution technique, to identify which features of the input data contribute the most to the model's prediction (the code for this task was generated using ChatGPT). We believe that the model will use the shape and color of patterns in the image to determine the style of the art. We removed 16 x 16 chunks from the input and analyzed how much the removal of that region impacted the model's probability of predicting the correct genre. Yellow means the change was significant and dark purple means the change was not significant.



(a) Cubism Padded (b) Cubism Attribution

Figure 8: Cubism sample image attribution.

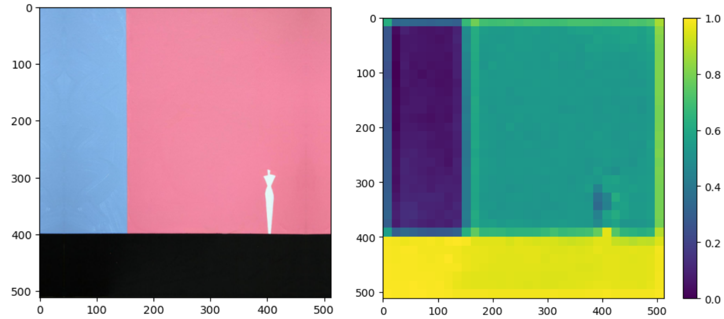
The model's decision was heavily influenced by the jagged blocks of beige. This may be because beige blocks are used differently in other art styles. In Late Renaissance, beige blocks are typically architectural features or human skin or light fabric, and are given far more texture than the color blocking in Cubism. In Color Field Painting, beige blocks are typically more regular in shape and less angular. In Ukiyo-e, beige blocks are used as a background color, appearing in wide expanses rather than jagged forms.



(a) Late Renaissance Padded (b) Late Renaissance Attribution

Figure 9: Late Renaissance sample image attribution.

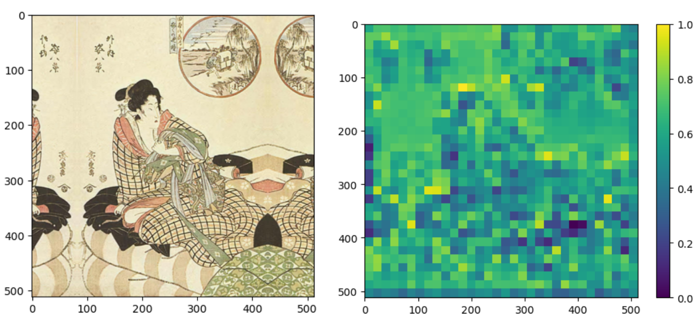
The model's decision seems to be based on the lighter skin and fabric of the central figures. This may be because textured patches of lighter color in the center of the piece commonly appear in Late Renaissance paintings, while other styles either do not have as much texture or presence of this type of composition.



(a) Color Field Painting Padded (b) Color Field Painting Attribution

Figure 10: Color Field Painting sample image attribution.

The blocks that influence the model's decision the most are the bottom block of black and the upper right block of pink. The black block likely has such high importance because it is a feature unique to color field paintings. None of the other styles include such large, uniform blocks of black. The blue block on the upper left is likely the least important to the model's decision because other styles include blue backgrounds either as sky or water.



(a) Ukiyo-e Padded (b) Ukiyo-e Attribution

Figure 11: Ukiyo-e sample image attribution.

The attribution for Ukiyo-e is a bit more ambiguous, but it looks like the model heavily bases its decision on sharp boundaries/edges between dark and light colors. This is likely unique to Ukiyo-e because it is a style which utilizes solid brush strokes to define figures and objects. Other styles do not use solid black lines to define the form of their figures and objects. There may be defined borders between subjects in Cubism and Color Field Painting, but they are not delineated by solid black lines.

The model also seems to utilize the background, as Ukiyo-e is the only genre to heavily feature central figures on a wide cream background. Other styles have more crowded compositions and fill the background space.

5 Conclusion

In this paper, we presented ArtSee, a convolutional neural network for art style classification. In order to create a model that equally emphasizes information related to color usage and brushstroke patterns in the original artwork, we created two paths that learn these two types of information separately. Using the WikiArt dataset containing artworks from Ukiyo-e, Mannerism Late Renaissance, Cubism, and Color Field Painting, we found an overall accuracy of 72.5%. Although our kernels were too hard to visualize due to their small size, from our perturbation-based methods, we found that ArtSee successfully puts the most importance on the artistic features unique to their historical periods.

6 Model Card

Model Details

Model Description:

Given an image of artwork, ArtSee classifies the artwork as one of four styles: Ukiyo-e, Mannerism Late Renaissance, Color Field Painting, Cubism.

Developed by:

Asha Karmakar, Teresa Luo, Akhil Iyer

Model type:

Convolutional Neural Network

Model Sources:

Fine-art painting classification via two-channel dual path networks | International Journal of Machine Learning and Cybernetics

Repository:

`https://github.com/akarmakar87/CS342_Final_Project`

Training Details

Training Data:

WikiArt is downsampled to use only 4 genres, and for each genre, we sampled 1100 examples. We sampled 80% of this data to include in our training set.

Training Procedure

Preprocessing:

For each image, we reduce its original resolution into a 512x512 image. If the image is not originally square, we perform reflection padding on the image to retain the artwork's color usage and brushstrokes. We finally normalize the RGB values of each pixel in the padded image.

Training Hyperparameters

Training regime:

We used SGD with a mini-batch size of 128 using a cross-entropy loss function. We initially set the learning rate to 0.0005 and decreased it by a factor of 0.10 every 2 epochs. We train the model for a total of 5 epochs.

Evaluation

Testing Data, Factors & Metrics

Testing Data:

WikiArt is downsampled to use only 4 genres, and for each genre, we sampled 1100 examples. Whatever data that was not included in the training set is part of the testing set.

Metrics:

We used accuracy (i.e., number of correct predictions / total number of predictions) to measure performance on the testing set. For a finer-grained analysis, we used a confusion matrix to assess how well the model was able to predict examples of each class.

Results:

We got a test set accuracy of 72.5%. More specifically, the model was able to get color field painting predictions correct the most relative to the other styles. The model struggled, however, in predicting paintings as a cubism style relative to the other styles.

7 Datasheet

The datasheet paper linked on the the Canvas assignment had many questions that didn't apply to the dataset we used, so we've compiled and answered the questions that are most applicable to our dataset:

Dataset title: WikiArt
Dataset location: `https://www.kaggle.com/datasets/steubk/wikiart/data`

Who created the dataset and on behalf of which entity?
The dataset we used was created by Stefano Morelli (under the username steubk) on Kaggle.com. The dataset description states that it was "upscaled and resized, originally from `https://github.com/cs-chan/ArtGAN/tree/master/WikiArt%20Dataset`".

What do the instances that comprise the dataset represent?
The instances that comprise the dataset represent artwork of various genres created by various artists throughout different time periods.

How many instances are there in total?
This dataset contains 80020 unique images from 1119 different artists in 27 styles.

What data does each instance consist of?
Each instance is a JPG file.

Is there a label or target associated with each instance?
There are 27 folders (one per art style). The images under each folder are therefore implicitly given a genre label. Additionally, the images are named with the artist and title of the artwork. There is also a CSV file with the following metadata: filename, artist, genre, description, phash, width, height, genre_count, subset. However, because a lot of the filenames do not match those of the actual image files, we do not end up using the CSV file.

Are there recommended data splits?
The CSV file recommends an 80-20 train/test split.

Are there any errors, sources of noise, or redundancies in the dataset?
Many of the filenames in the provided CSV file do not match the names of the actual image files in the different art style folders, so the CSV is essentially unusable. However, this is not a problem because the most important information we need (genre label for each image) can be determined based on the folder the image is in, so the CSV is not necessary.

Is the dataset self-contained, or does it link to or otherwise rely on external resources?
The dataset is self-contained.

Does the dataset contain data that might be considered confidential?
No.

Does the dataset contain data that might be considered sensitive in any way?
No.

How was the data associated with each instance acquired? Was the data directly observable, reported by subjects, or indirectly inferred/derived from other data?
This original data was downloaded from `https://www.wikiart.org/`.

Has the dataset been used for any tasks already?
Not this exact dataset, but the original dataset this one was built off was used in the following project to generate art (trained on WikiArt images): `https://archive.org/details/wikiart-stylegan2-conditional-model/`

What (other) tasks could the dataset be used for?
In addition to generative uses, this dataset can be used to predict the style of an artwork, which is what we are doing in our project.

8 Group Member Contributions

Asha Karmakar

- Organized the 10 GB of data on my Google Drive account and set up an environment for the model in Colab.
- Wrote code for selecting and loading in the images and worked with Teresa to do the image transformations.
- Extensively trained the model and tuned hyperparameters (with input from group members).
- Generated the confusion matrix and selected correctly/incorrectly classified images for analysis (analysis done by Teresa).
- Wrote Data Preprocessing and Datasheet sections of the paper, and finalized formatting and references. Cleaned up and formatted code for final submission.

Teresa Luo

- Helped Asha debug image transformation in the Data Preprocessing section.
- Designed and implemented both experiments (kernel visualization and feature attribution).
- Experimented on various images from all styles to understand the model.
- Wrote Abstract, Experiments, Related Works, and Introduction sections of the paper, and helped with the analysis for Examples of Correctly/Incorrectly Classified Images.

Akhil Iyer

- Wrote PyTorch code for model architecture.
- Helped Asha train the model and tune hyperparameters of the model as well as hyperparameters for the training process.
- Wrote the Model Architecture, Overall Results, and Model Card sections of the paper.

References

E. Cetinic, T. Lipic, and S. Grgic. Fine-tuning convolutional neural networks for fine art classification. *Expert Systems with Applications*, 114:107–118, 2018. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.07.026. URL `https://www.sciencedirect.com/science/article/pii/S0957417418304421`.

Stefano Morelli. Kaggle WikiArt Dataset. https://www.kaggle.com/datasets/steubk/wikiart/data, 2022. Accessed: April 2024.

S.-h. Zhong, X. Huang, and Z. Xiao. Fine-art painting classification via two-channel dual path networks. *International Journal of Machine Learning and Cybernetics*, 11(1):137–152, 2020. ISSN 1868-808X. doi: 10.1007/s13042-019-00963-0. URL https://doi.org/10.1007/s13042-019-00963-0.