Art Style Classification Through Neural Networks: Refining High-Level & Artist-Specific Classifications

Hadley Griffin[1,†]

[1]Maadhoo Distributors Inc.
118 North Bedford Road, Suite 100, Mount Kisco, NY 10549

[†]Address to which correspondence should be addressed:
hadley@maadho.com

Abstract:

Classifying artworks by both style and artist remains a significant challenge in computer vision. While deep learning models, particularly Convolutional Neural Networks (CNNs), have shown promise in style recognition, distinguishing individual artists within a style remains difficult due to overlapping visual characteristics and nuanced stylistic variations. Existing approaches struggle to generalize across diverse artistic expressions, limiting their effectiveness in fine-grained classification tasks.

A two-step classification framework using a multi-layer CNN was used to help address the current generalization struggle. The first network categorizes artworks into broad stylistic groups, while a second network refines the classification by identifying the specific artist within each style.

The model was trained on the WikiArt dataset, a diverse collection of over 80,000 artworks, with 2500 images samples for manageable computational requirements. The dataset's images, resized to 256x256 pixels and normalized, were processed for classification tasks with integer labels that were one-hot encoded. The artist names were extracted from structured image filename and also one-hot encoded for consistency. The custom CNN utilized techniques like L2 regularization, dropout, early stopping, and a learning rate scheduler to help enhance training stability and efficiency.

The model's performance was evaluated with metrics including accuracy, recall, and F1 scores. Results showed that the model struggled with generalization, especially regarding underrepresented genres and artists, achieving low genre classification accuracy and poor artist classification performance. The model's tendency to cluster prediction around a few frequently predicted artist suggests a reliance on low-level features, such as color and texture of an artwork, instead of higher-level compositional structures.

While the CNN performed relatively well during training, its ability to generalize was drastically reduced by dataset imbalance and overfitting. Future research will explore dynamic neurons and additional preprocessing techniques, such as class weighting or oversampling, to improve performance and address the current limitations of the model. This study represents a step forward in advancing automated recognition of artistic styles and artists.

Keywords:

Art Style Classification, Neural Networks, CNN, Abstraction, Nodes, Impressionism, Realism, Baroque, Wikiart, Accuracy, Classification.

Table of Contents

Introduction & Problem Statement:

Introduction:

The field of image classification has advanced significantly with the adoption of deep learning techniques, particularly Convolutional Neural Networks (CNNs). In the domain of art, automatic classification of artworks into styles and identifying artists has practical applications in digital curation, art authentication, and cultural analysis. However, achieving high accuracy remains challenging due to the complex and often overlapping visual characteristics of different artistic styles.

While prior research has demonstrated CNNs' effectiveness in recognizing broad art styles, distinguishing between individual artists within a style remains a difficult task. This study aims to address this challenge by implementing a multi-step classification approach. A subnet will first classify the artwork into a broad style category, and a subsequent main network, using the subnet's output as additional input, will classify the specific artist. The study will evaluate accurately the designed model can characterize the artist and genres of a given artwork, through accuracy, precision, recall, and F1 scores.

Problem Statement:

The objective of the study is to accurately classify artworks into both broad styles and their corresponding artists. While pre-trained CNN models are highly effective in general object classification, their application to art style and artist classification has not been extensively explored. The classification challenge becomes even more complex when introducing a dataset that covers a diverse selection of artists and genres over a 300-hundred-year period. This study will explore these factors to design a neural network that optimally balances model simplicity and classification accuracy, providing a robust framework for future advancements in fine-grained art classification.

Background:

The classification of artistic styles using neural networks has been a topic of increasing interest in the art world in recent years, driven by advancements in deep learning as well as the increase in public domain artworks. Prior research has established theoretical and practical foundations for this work, highlighting key methodologies and challenges in art style classification.

Lelievre and Neri explore how deep learning can be used to model the human perception of artistic composition, focusing on abstract art (Lelievre and Neri, 2021). Unlike previous research with models that relied on explicit art theory rules or low-level image statistics, their model learned composition implicitly, and outperformed previous methods across various styles. One insight from their work was that abstract paintings, which lack clear object boundaries and familiar patterns, often require deeper network layers for orientation judgements. This challenge aligns with difficulties encountered when using datasets like WikiArt, where traditional labels like genre and artist, may not correspond to easily identifiable visual features that a model can learn.

In 2015, He et al. proposed Residual Networks (ResNet) in their paper *Deep Residual Learning for Image Recognition,* as a way to address the vanishing gradient problem, through the introduction of residual connections (He 2015). This finding enabled deeper networks to be trained without losing stability. While ResNet offer advantages in deep hierarchical feature extraction, this project will focus on leveraging VGG-based architecture due to their well-documented success in style recognition.

Jeremiah Johnson's publication, *Neural Style Representations and the Large-Scale Classification of Artistic Style,* explored how deep neural networks encode artistic styles and how these representations can be used for large scale classification of artworks (Johnson 2017). Their work demonstrated that CNNs can effectively classify paintings based on artistic style, while placing emphasis on the importance of fine-tuning pretrained models for style classification instead of object recognition. However, a key limitation identified in their study was the difficulty in distinguishing individual artists within overlapping artistic styles. This model will build upon these findings by incorporating dynamic architectural adjustments to enhance artistic classification.

The study, *Deep Ensemble Art Style Recongition*, by Sofou, Stamou, and Mastromichalakis further examines the application of several CNN architectures on large-scale art datasets (Sofou, Stamou, and Mastromichalakis, 2024). The study compares VGG, ResNet, and DenseNet architecture, with the focus on using transfer learning and fine-tuning in order to get optimal style classification. The results suggest that data preprocessing techniques like resizing, instead of cropping, can significantly help performance by preserving essential features of an artistic style.

By integrating these insights, this study seeks to process art data and develop a neural network that adapts its architecture based on classification complexity, ultimately improving accuracy in both style and artist recognition. This approach contributes to a more context-aware and flexible method for fine-grained art classification.

Data:

The dataset used for this study is the opensource WikiArt dataset, which is made up of 13,343 training images, 1,300 validation images, and 631 test images. These images are categorized into 27 different genres, covering a wide range of 19^{th} and 20^{th} century artistic movements. Example images of the five most common art genres within the dataset are shown in Figure 1 below. All images have associated genre classes, ensuring a clean and structured classification approach.

Figure 1. Example Images of Five Most Common Art Genres within Dataset

In order to ensure consistency in the input data, all images were resized to 256x256 pixels. A normalization layer was also applied to scale the pixel values to the range [0,1] by dividing each pixel value by 255. The dataset was then processed using TensorFlow's *image_dataset_from_directory* function, where the images were assigned integer labels corresponding to their genre. These integer labels were then one-hot encoded in order to streamline classification tasks. Figure 3 below shows the distribution of all genre classes within the dataset. The least common genres were visualized, in order to compare the coloring and artistic styles to the five most common genres (Figure 2).

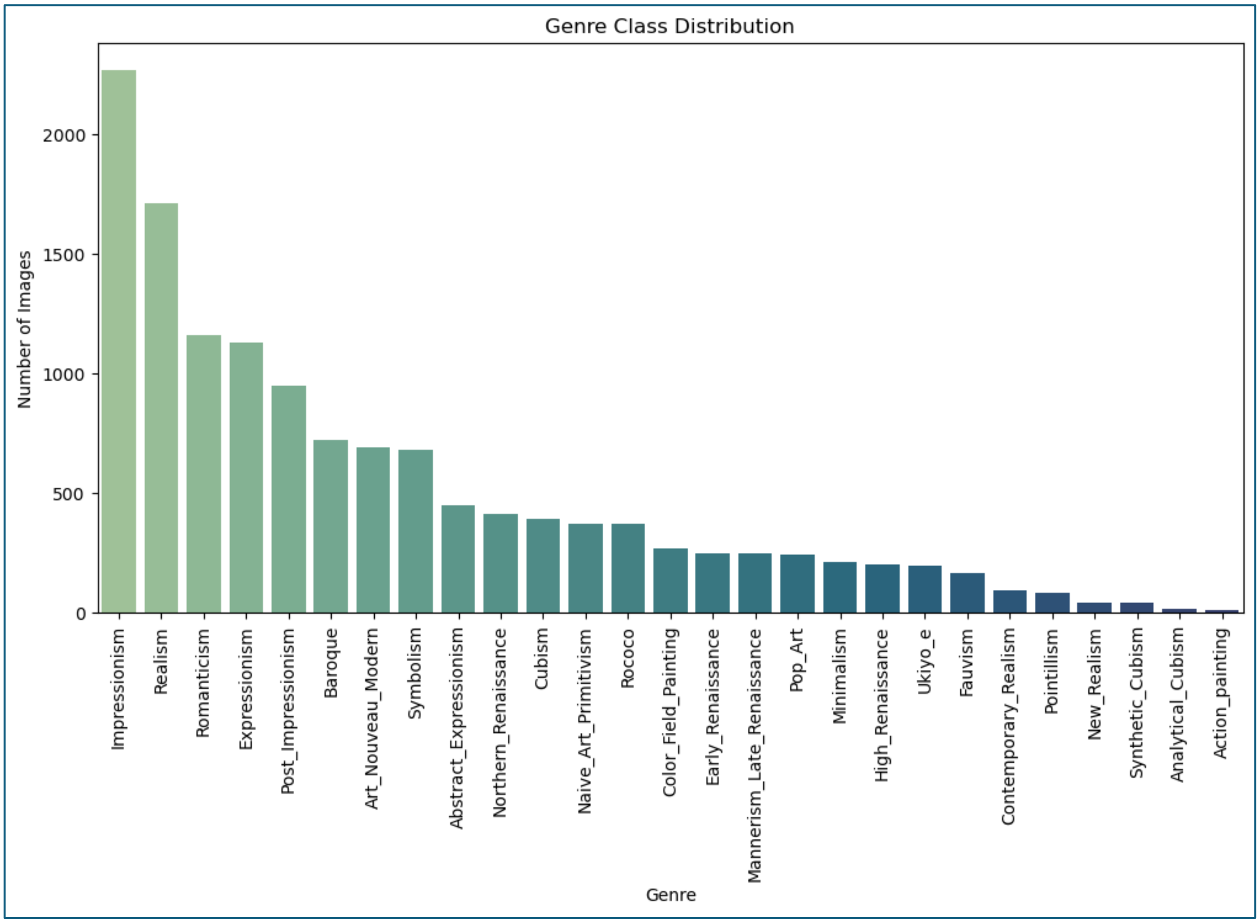


Figure 2. Distribution of Genre Classes

As described by the researchers, Lelievre and Neri, the model may have difficulty in characterizing artworks into the lesser common genres (Lelievre and Neri, 2021). This is not only due having less images of those images within the training dataset but also because many of the less common genres are abstract in design nature and thus tend to be more difficult for ML models to classify.

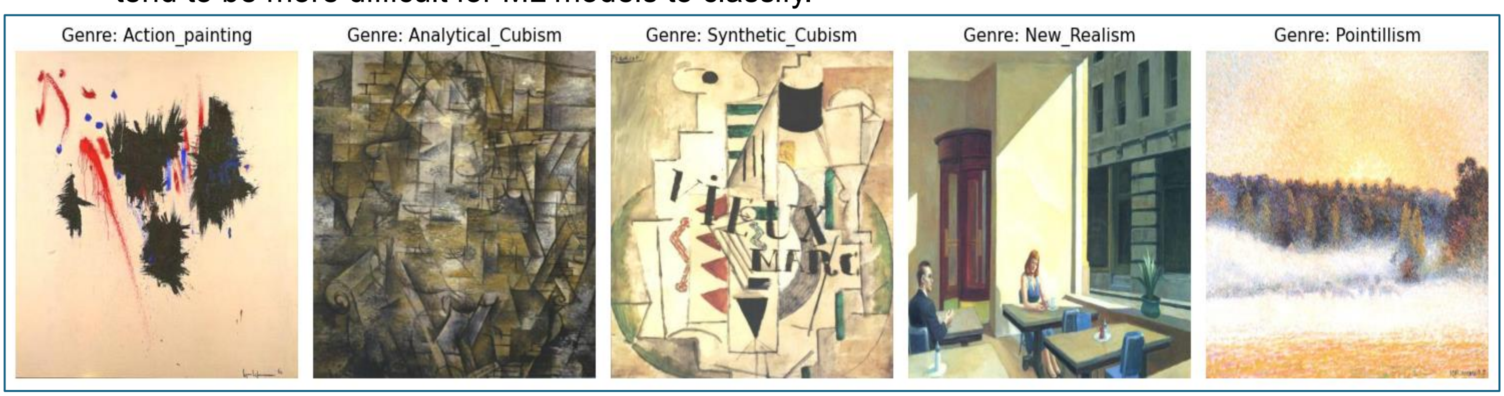


Figure 3. Example Images of Five Least Common Art Genres within Dataset

While the dataset already contained the genre class associated with each art image, the artist's name was not. Each image filename follows a structured naming convention of 'artist-name_artwork-name-year.jpg". The artist names were extracted from the filenames through the function, *extract_artist_from_filename*, in order to ensure consistent labeling across the dataset. The artist labels were then one-hot encoded for data standardization. The distribution of all artists within the dataset is shown in Figure 4. Figure 5 shows example works of the top five most common artists throughout the training dataset.

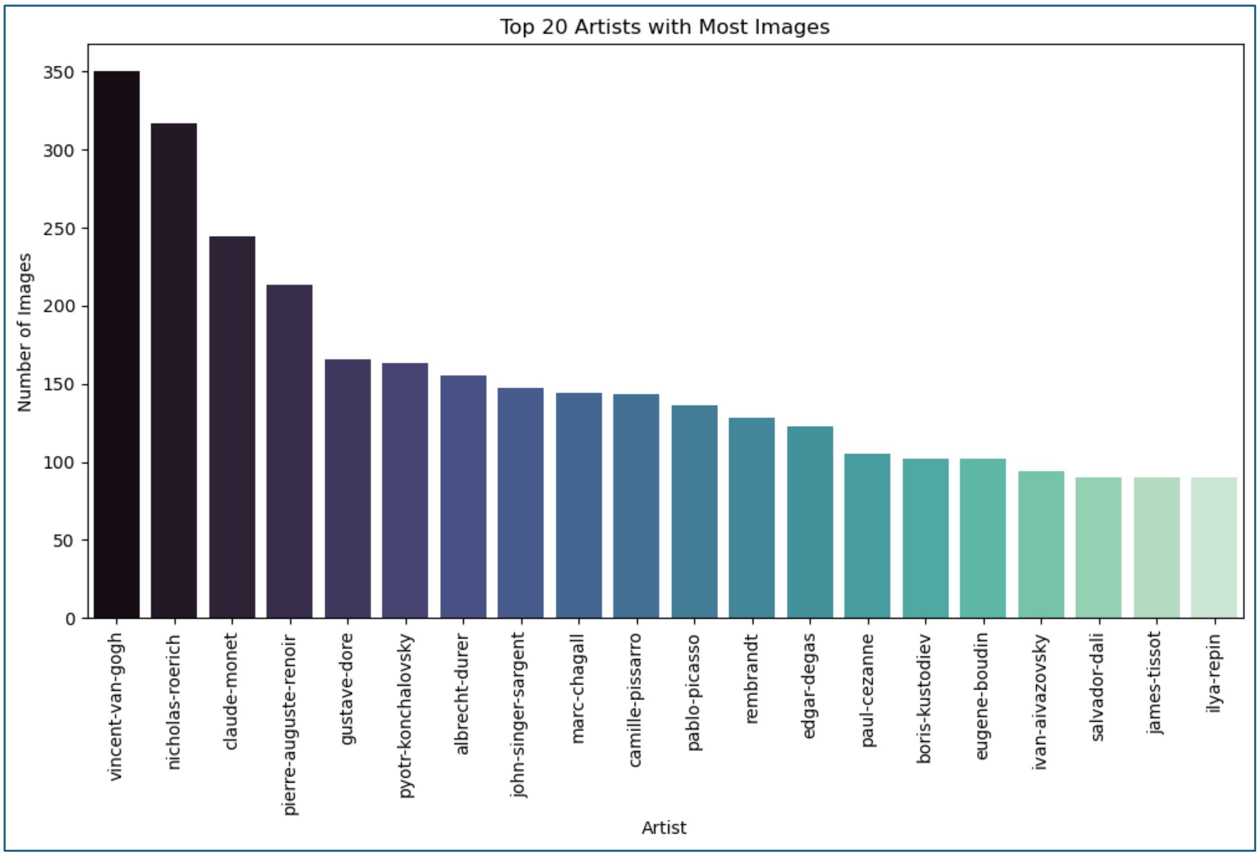


Figure 4. Distribution of Artists within Dataset

The example images highlight the additional struggle for art characterization through neural networks. Artists can have styles that change throughout their artistic career. A model that is training on a broad dataset of various artists, may not include all works of a given artist and therefore not full encapsulate all the styles of a given artist.

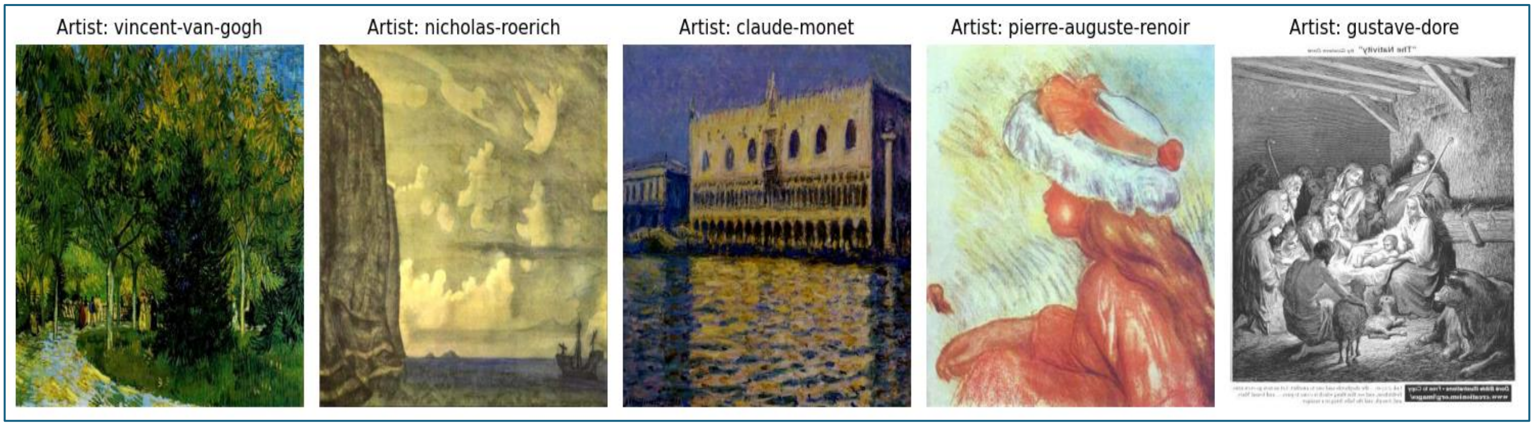


Figure 5. Example Images of the Five Most Common Artists within Dataset

Methods:

A custom Convolutional Neural Network (CNN) was designed to classify both art genre and artist. The architecture begins with an input layer that accepts images with the shape (256, 256, 3), followed by two convolutional layers. The first convolutional layer applies 32 filters with a 3x3 kernel and ReLU activation, while the second layer uses 64 filters with the same configuration. Both layers incorporate L2 regularization in order to reduce overfitting. Batch Normalization is then applied after each convolutional layer to stabilize learning of the model. MaxPooling, with a 2x2 pooling size, follows each convolutional layer in order to reduce spatial dimensions. After feature extraction, a Global Average Pooling Layer then compresses the feature maps into a single feature vector. This is then followed by a Dropout layer, with 0.5 probability, to further prevent any overfitting. The model has two output layers. One layer is for genre classification, made up of 27 neurons with SoftMax activation. The second layer is for artist classification, consisting of 851 neurons with SoftMax activation.

The model was then compiled using the Adam optimizer with a learning rate of 0.0001. Adam combines the benefits of both momentum-based optimization and adaptive learning rate, making it great for deep learning tasks with a large dataset, such as WikiArts. The learning rate was set low enough to achieve stable convergence but high enough to still ensure the model learns efficiently. Categorical cross-entropy loss was used for both genre and artistic classification. This loss function calculates the difference between the predicted probability distribution and the true distribution. This makes sure that the model learns to correctly assign probabilities to each class. Accuracy was used as the primary evaluation metric for both tasks. In order to enhance performance and further prevent overfitting, L2 regularization was applied to the convolutional layers, which encourages the model to learn more generalizable features by penalizing large weight values. A Dropout of 0.5 was added to randomly deactivate neurons during training. This further helps prevent the model from relying too heavily on specific features, forcing it to develop more robust feature representations.

Early stopping was also added to the model in order to monitor validation loss, halting training if there was no improvement observed after three consecutive epochs. This prevents unnecessary training, that could possibly lead to overfitting. Early stopping makes sure that the model stops at the optimal time, before performance on any unseen data begins to decrease. Also, a learning rate scheduler was included to reduce the learning rate by 50%, if validation loss did not improve for two epochs. This helps the model transition from a faster learning phase to a fine-tuning phase, making the model make more precise updates as it converges.

These design choices together optimize training stability, efficiency, and generalization. This ensures that the model can accurately differentiate between a large number of artists while still maintaining stability and avoiding overfitting.

Results:

The model's performance was evaluated using various metrics, including accuracy, recall, and F1 scores. The following results highlight the model's strengths and limitations, such as its tendency to overfit and struggle with generalizations.

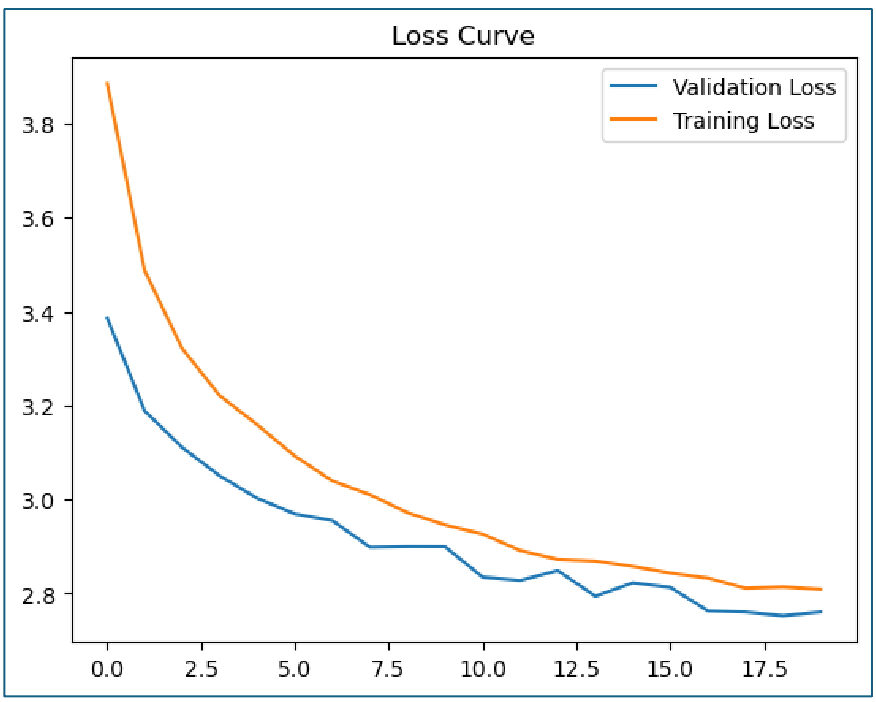


Figure 6. Loss curve for Training and Validation

The loss curve for both the training and validation sets are shown in Figure 6 above. The training loss steadily decreased while the validation loss plateaued early. However, despite the early stopping, the validation loss stayed high, possibly indicating the model having

difficulty in generalizing unseen data. The genre accuracy curve, shown in Figure 7, illustrates the model's genre classification performance over time. The training accuracy steadily increased, while the validation accuracy remained quite low throughout the whole training process.

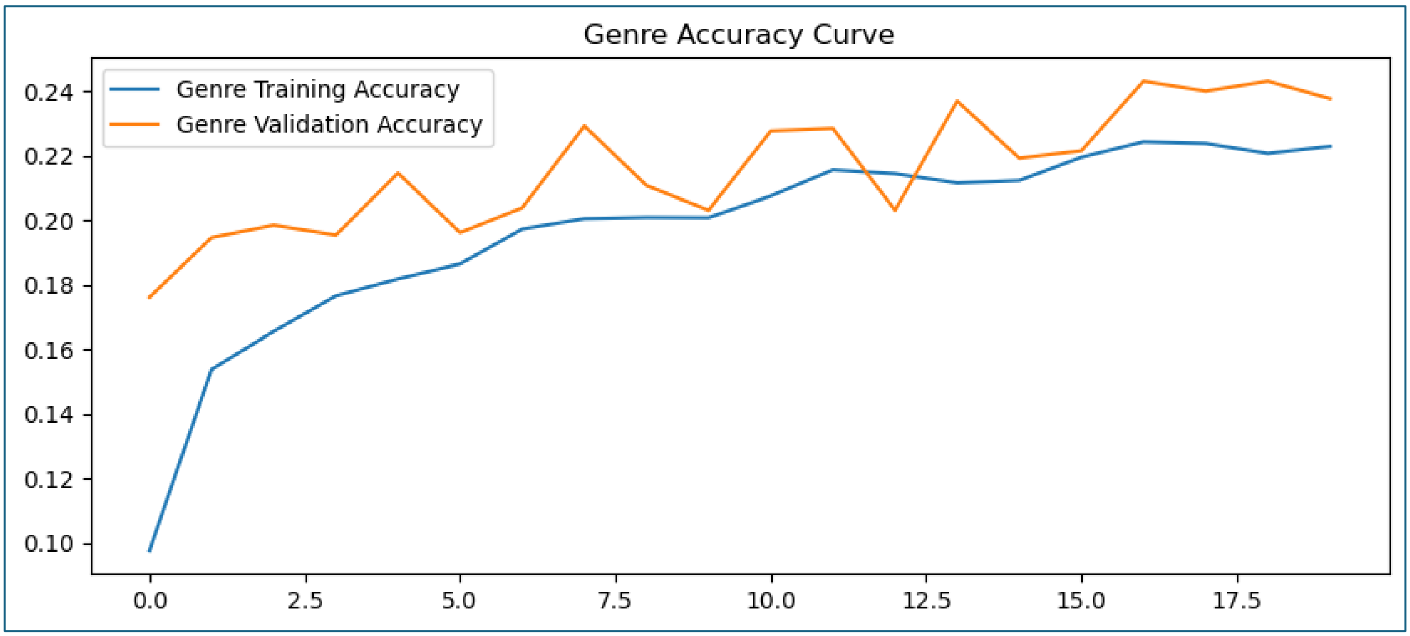


Figure 7. Genre Accuracy Curve

The genre classification accuracy was assessed using both the top genre and top five genres average accuracy. The top 1 genre accuracy reached 14%, while the top 5 genre accuracy increased to 47% (Table 1). The top performing genres regarding recall were Impressionism, Realism, and Expressionism (Table 2).

Table 1. Model Accuracy Metrics Overview

Metric	Accuracy (%)
Top-1 Genre Accuracy	14
Top-5 Genre Accuracy	47
F1- Score	8

Table 2. Genres with Highest Recall

Genre	Recall (%)
Impressionism	55
Realism	23
Expressionism	10

The genre confusion matrix, shown below in Figure 8, highlights significant misclassifications between visually similar genres. For example, genres like Post-Impressionism and Expressionism were commonly misclassified as Impressionism, likely due to their shared visual features. Also, many underrepresented genres were misclassified into more frequently represented genres, like Impressionism.

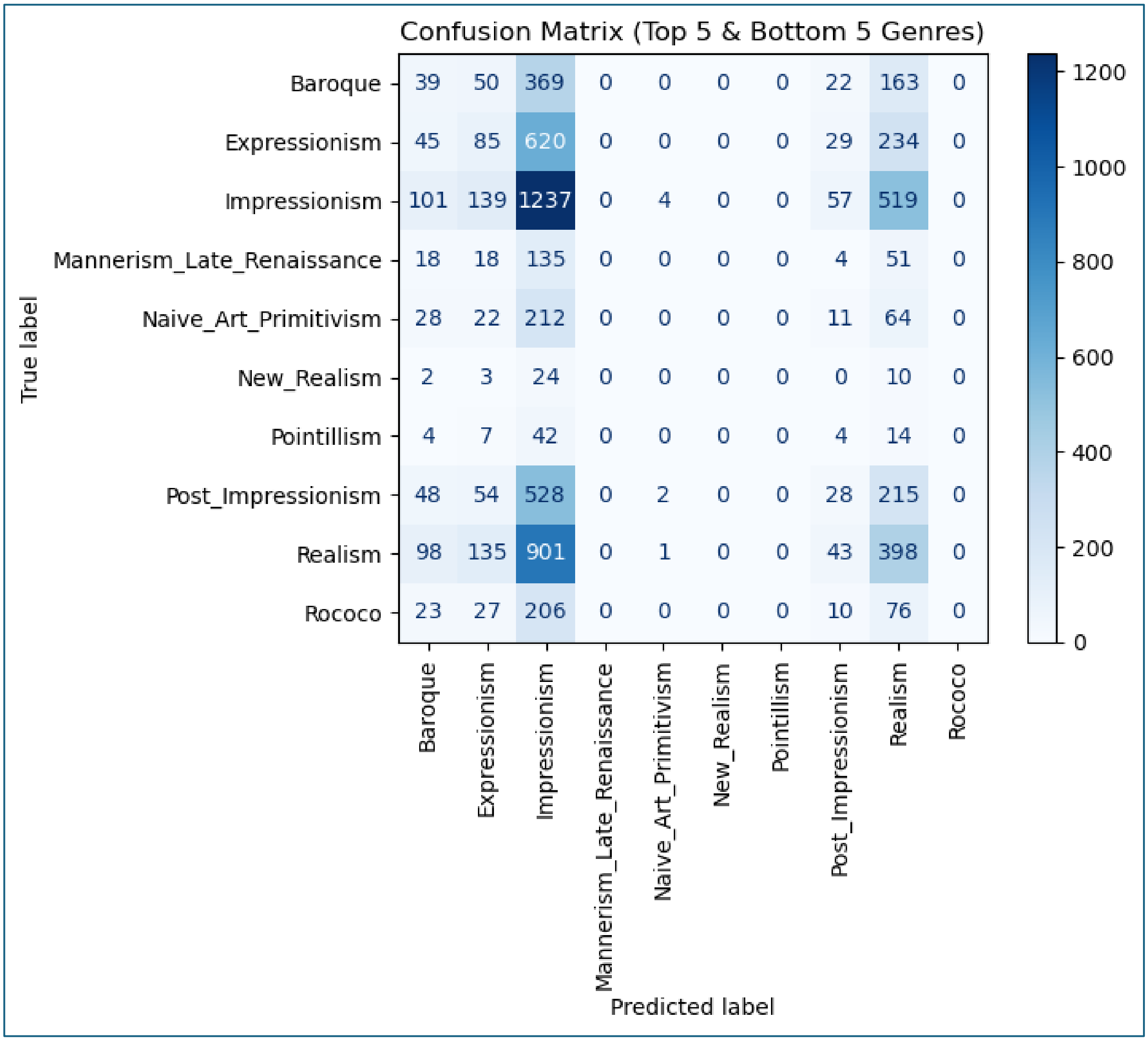


Figure 8. Genre Confusion Matrix

The distribution of artists prediction confidence is shown in Figure 9 below. Similar to the genre predictions, the confidence levels for artists classifications were low, with few exceptions.

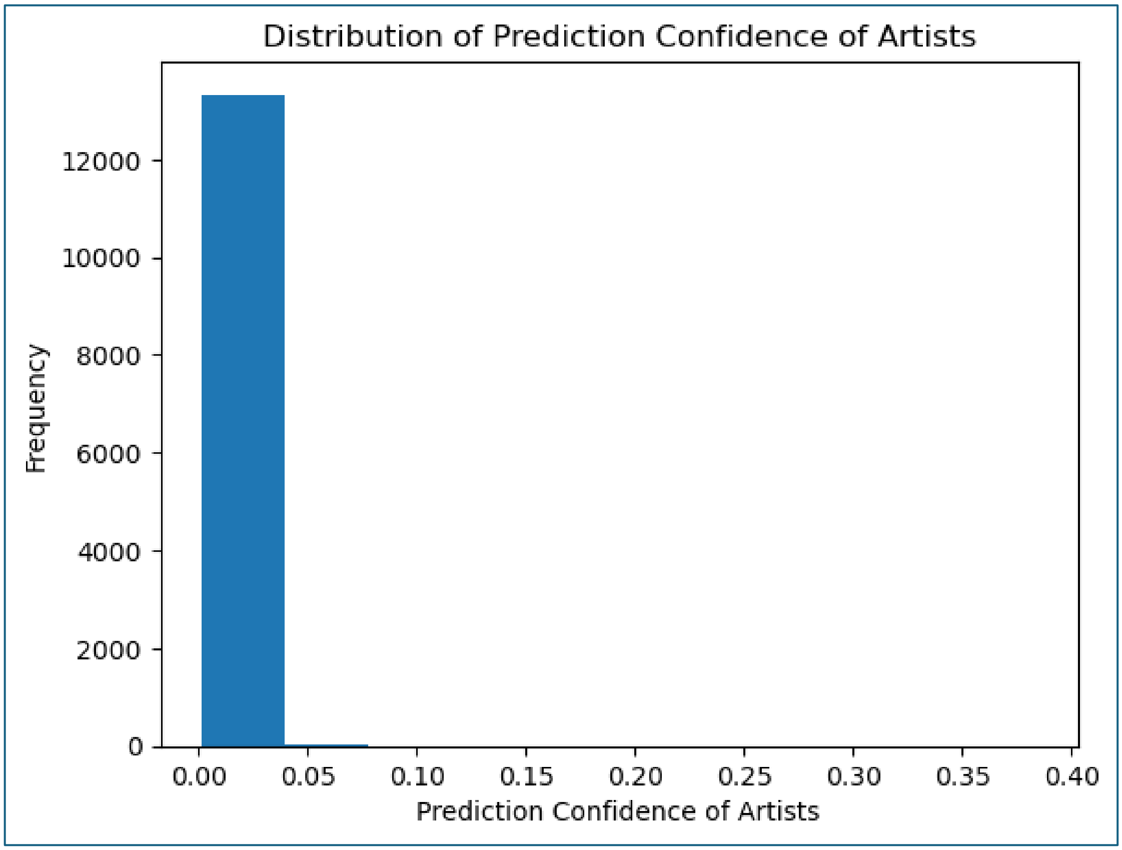


Figure 9. Distribution of Artists Prediction Confidence

Table 3 below shows the most commonly predicted artists by the model. It should be noted that none of these artists were frequently seen in the training dataset, shown in Figure 4.

Table 3. Most Commonly Predicted Artists

Artist	Prediction Counts
Keith Haring	2386
Anita Nio de Carvalo da Silva Porto	1769
William Turner	1107
Francesco Hayez	829
Tano Festa	729

Figures 10 and 11 are two examples of misclassified artworks, including both the predicted and actual genres and artists. Figure 10, a Baroque painting by Ivan Nikitin, was incorrectly classified as an Impressionist work. While Figure 10 is a Fauvist painting but was misclassified by the model as a Realism. These examples highlight the classification inaccuracies of the trained model.

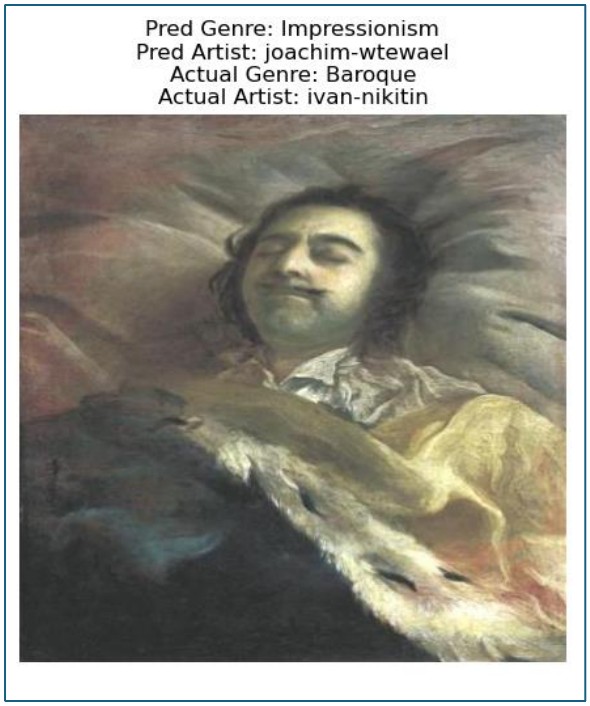


Figure 10. Example of Misclassified Art by Ivan Nikitin

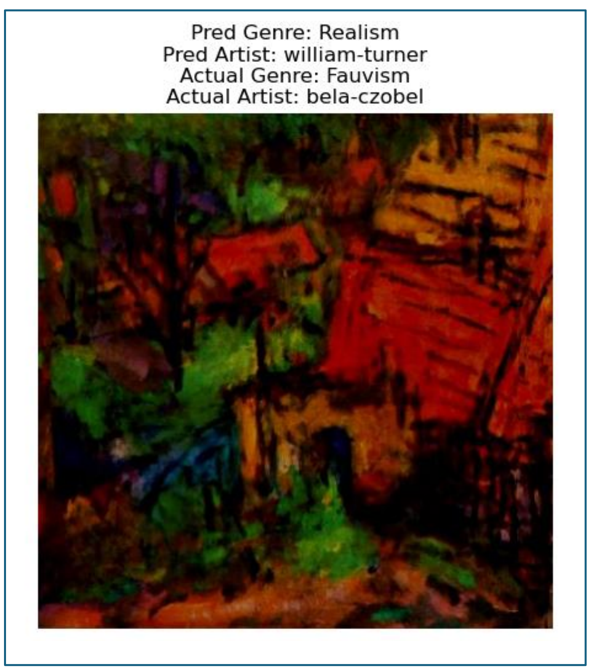


Figure 11. Example of Misclassified Art By Bela Czobel

Discussion:

The model consisted of two convolutional layers, followed by batch normalization, max pooling, and a fully connected layer that branches into two outputs: one for genre classification and one for artist classification.

The loss curve also revealed that the training loss steadily decreased, while the validation loss plateaued early, further indicating overfitting of the model (Figure 6). The early stopping function prevented excessive overfitting. However, the validation loss still stayed proportionately high, indicating that the model also struggled with generalization. The genre accuracy curves show an increasing trend in the training accuracy, while the validation accuracy stayed consistently low (Figure 7). Also, the sharp fluctuations seen in validation accuracy indicates that the model had a difficult time learning stable representations for the various genres, especially for the genres with less training samples. Both genre accuracy curves as well as loss curves indicate overfitting, suggesting that the model was most likely memorizing superficial visual patterns from the training set, instead of learning generalizable features

The model's genre classification accuracy was 14%, with a top 5 genre accuracy of 47% (Table 1). The best performing genre was Impressionism (55% recall) (Table 2). However, many underrepresented genres had recall scores close to 0%. The model seemed to perform well on highly recognizable styles like Impressionism, but it struggled with the more nuanced artistic movements that share visual similarities, such as Post-Impressionism and Symbolism. The confusion matrix showed that there was significant misclassification between historically related styles, most likely due to overlapping feature representations as well as class imbalances (Figure 8). The confusion matrix results also indicates that the CNN filters were possibly detecting only low-level features like color schemes and brushstrokes, rather than higher-level compositional structures, which would be necessary for more accurate genre differentiation.

In terms of art classification, performance was extremely poor, with F1 scores near zero for most artists (Figure 9). Even very famous artists, found frequently within the training dataset, like Van Gogh and Picasso, were frequently misclassified. Also, the dataset was made up of many artists that had only a few samples to represent their works, most likely contributing to the poor generalization and the issue of overfitting. The model seems to have consistently clustered predictions around a few artists, while less represented artists had almost zero classification success (Table 3). This is curious because, Keith Haring was the most frequently predicted artist, however, he along with the four other most commonly predicted artists were not even within the top twenty artists with images within the training dataset. This pattern indicates that the model possibly defaulted to certain artists, even when their style did not match the input images. Though it is unique that none of the commonly predicted artists commonly appeared in the training it, their styles were possibly generalized by the network. For example, the over representation of Keith Haring, a Pop Art artist with a highly distinct style, shows that the model may be overfitting to artists with very recognizable visual patterns. Similarly, Turner's prominence suggests that the model be also be prioritizing well-known historical artists, even in cases where their style does not align with the input image.

When examining a few misclassified images, a few recurring patterns were observed in the model's errors. Figure 10, a Baroque painting by Ivan Nikitin, was incorrectly misclassified in both genre and artist. The misclassification suggests that the model was influenced by the soft brushwork as well as the light tonal balance of the work, which both are typical also of Impressionist works. Similarly, with Figure 11, where the model mistook the bold, expressive coloring of a Fauvist painting for a Realism painting, suggesting that the model may not be able to completely differentiate color-based abstraction from representational realism. From these errors, it seems that the key issues are surrounding the model's heavy reliance on color and texture, as well as its confusion between related or visually overlapping genres. Additionally, the same set of artists appeared frequently in the misclassifications.

The poor performance of the model seems to be in part due to the imbalanced nature of the training dataset. There being a large imbalance between the number of images for each genre and artists seemed to cause issues when training the model. This is especially seen in regard to the model's attempt at genre classification, where only the genres most frequently found within the training dataset, were used by the model to classify the validation set. A difficulty faced when building the model was how computationally expensive and time consuming each model run was. This limited the total number of iterations to optimize the model's performance.

Conclusions:

The study aimed to classify artworks from the WikiArt dataset into genres and artists using a custom Convolutional Neural Network (CNN). Though the model had consistent training performance, it struggled with generalization, especially for underrepresented genres and artists. The genre classification accuracy was low, end the model had difficulty distinguishing between visually similar styles. For artist classification, the model showed a tendency to cluster predictions around a few select artists, often less common artists from the training data. These issues largely attributed to the imbalance in the dataset, overfitting, and the model's reliance on low-level features.

Directions for Future Works:

For future work, several paths will be explored in order to enhance the model's performance and address its current limitations. The use of dynamic neurons in the network will be explored more. By incorporating dynamic neurons, the model would be able to adaptively adjust its capacity to learn and process different features at various layers of the network. This addition could potentially improve its ability to generalize across the diverse genres and artists within the dataset. Further preprocessing techniques will also be explored, in order to improve the model's performance. This includes the addition of either class weight or oversampling for the less represented genres and artists, in order to help reduce the impact of class imbalance. This additional preprocessing step could also help the model learning to recognize these categories more accurately.

References:

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition.” *arXiv Preprint arXiv:1512.03385*. https://arxiv.org/abs/1512.03385.

Johnson, Jerimiah. 2017. “Neural Style Representations and the Large-Scale Classification of Artistic Style.” *Future Technologies Conference Proceedings*. https://saiconference.com/Downloads/FTC2017/Proceedings/39_Paper_197-Neural_Style_Representations_and_the_Large-Scale.pdf.

Lelièvre, Pierre, Neri, Peter. 2021. “A deep-learning framework for human perception of abstract art composition.” *Journal of Vision*, 21(5):9. https://doi.org/10.1167/jov.21.5.9.

Menis-Mastromichalakis, O., Sofou, N., & Stamou, G. 2024. “Deep ensemble art style recognition” *International Joint Conference on Neural Networks,* May 19. *(IJCNN)* (pp. 1-8). IEEE. https://arxiv.org/abs/2405.11675v1

Mezina, Anzhelika, Burget, Radim. 2025. “EnsArtNet: Ensemble Neural Network Architectures for Identifying Art Styles from Paintings.” *Journal of Cultural heritage,* March 1. Volume 72, Pages 71-80, ISSN 1296-2074, https://doi.org/10.1016/j.culher.2025.01.005.

Hugging Face. 2023. “WikiArt.” *HuggingFace*, https://huggingface.co/datasets/Artificio/WikiArt

Ng, Andrew. NA. “Multi-Layer Neural Networks.” *UFLDL Tutorial.* *http://ufldl.stanford.edu/tutorial/*

Artland Editors. NA. “The Most Famous Art Movements and Styles.” *Artland Magazine.* *https://magazine.artland.com/art-movements-and-styles/*

Appendix 1

In [125…

```
import os
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sns
import cv2
from roboflow import Roboflow
from collections import Counter
from tensorflow.keras import layers, models, regularizers
from tensorflow.keras.applications import VGG16
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Flatten, Dense, Dropout, GlobalAveragePoolin
from tensorflow.keras.utils import to_categorical
from keras.models import Model
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
```

In [2]:

```
## reading in wikiarts data set through roboflow
rf = Roboflow(api_key="Gc5mrHmsGPLcdqDOE5nM")
project = rf.workspace("art-dataset").project("wiki-art")
version = project.version(2)
dataset = version.download("folder")
print(f"Dataset downloaded to: {dataset.location}")
```

```
loading Roboflow workspace...
loading Roboflow project...
Dataset downloaded to: /Users/hadleygriffin/MSDS 458/wiki-art-2
```

In [152…

```
#creating directories for each file
dataset_dir = '/Users/hadleygriffin/MSDS 458/wiki-art-2'
train_dir = os.path.join(dataset_dir, "train")
valid_dir = os.path.join(dataset_dir, "valid")
test_dir  = os.path.join(dataset_dir, "test")
```

In [153…

```
print(os.path.exists(train_dir))
print(os.listdir(train_dir)[:5])
```

```
True
['Early_Renaissance', 'Analytical_Cubism', 'Mannerism_Late_Renaissance', 'Expressionism', 'Contemporary_Reali
sm']
```

In [154…
```
#setting image & batch size
batch_size = 32
img_size = (256, 256)
```

In [155…
```
# processing directories, and converting into datasets for easy of analysis
def process_dataset(directory, shuffle=True):
    ds = tf.keras.preprocessing.image_dataset_from_directory(
        directory,
        image_size=img_size, #256,256
        batch_size=batch_size, #32
        label_mode='categorical',  #one hot encoding
        shuffle=shuffle
    )
    return ds
```

In [156…
```
train_ds = process_dataset(train_dir, shuffle=True)
valid_ds = process_dataset(valid_dir, shuffle=False)
test_ds = process_dataset(test_dir, shuffle=False)
```

```
Found 13343 files belonging to 27 classes.
Found 1300 files belonging to 27 classes.
Found 631 files belonging to 27 classes.
```

In [163…
```
#genre classes for images
genre_classes = sorted([d for d in os.listdir(train_dir) if not d.startswith('.')])
print("Genre classes:", genre_classes)
```

```
Genre classes: ['Abstract_Expressionism', 'Action_painting', 'Analytical_Cubism', 'Art_Nouveau_Modern', 'Baro
que', 'Color_Field_Painting', 'Contemporary_Realism', 'Cubism', 'Early_Renaissance', 'Expressionism', 'Fauvis
m', 'High_Renaissance', 'Impressionism', 'Mannerism_Late_Renaissance', 'Minimalism', 'Naive_Art_Primitivism',
'New_Realism', 'Northern_Renaissance', 'Pointillism', 'Pop_Art', 'Post_Impressionism', 'Realism', 'Rococo',
'Romanticism', 'Symbolism', 'Synthetic_Cubism', 'Ukiyo_e']
```

In [164…
```
genre_to_index = {genre: idx for idx, genre in enumerate(genre_classes)}
```

In [165…
```
#getting artist name from .jpg file name
def extract_artist_from_filename(file_path):
    filename = os.path.basename(file_path)  #getting filepath names
    artist_name = filename.split('_')[0]  #splitting file names before _ (bw artist name & artwork name )
    return artist_name
```

In [166…
```
#pulling image file paths
image_files = [os.path.join(root, file)
```

```
                for root, _, files in os.walk(train_dir)
                for file in files if file.lower().endswith(('.jpg', '.jpeg', '.png'))]
```

In [167…

```
artists = list(set([extract_artist_from_filename(f) for f in image_files])) #artist names into list
artist_to_index = {artist: idx for idx, artist in enumerate(artists)} #indexing artist names
```

In [168…

```
artist_labels = [extract_artist_from_filename(f) for f in image_files]  #creating artist label for dataset
artist_indices = [artist_to_index[artist] for artist in artist_labels]
artist_one_hot_labels = to_categorical(artist_indices, num_classes = len(artist_to_index)) #one hot encoding
```

In [169…

```
def extract_genre_from_directory(file_path, genre_to_index): #pulling genre name to get label
    genre_name = os.path.basename(os.path.dirname(file_path))  #genre folder path
    return genre_to_index.get(genre_name, -1)
```

In [170…

```
genre_labels = [extract_genre_from_directory(f, genre_to_index) for f in image_files] #creating genre label f
genre_one_hot_labels = to_categorical(genre_labels, num_classes=len(genre_to_index)) #one hot encodng
```

In [171…

```
print(f"total images found: {len(image_files)}")
print(f"total artists: {len(artists)}")
print(f"total genre labels: {len(genre_labels)}")
print(f"onehot encoded genre labels shape: {len(genre_one_hot_labels)}, {genre_one_hot_labels[0].shape}")
```

```
total images found: 13343
total artists: 851
total genre labels: 13343
onehot encoded genre labels shape: 13343, (27,)
```

In [172…

```
#artists
```

Data Visualization:

In [173…

```
def plot_sample_images(num_samples=5):
    sample_files = random.sample(image_files, num_samples)  #randomize selection
    fig, axes = plt.subplots(1, num_samples, figsize=(15, 5))

    for i, img_path in enumerate(sample_files):
        if not os.path.exists(img_path): #error add in
            print(f"File not found: {img_path}")
            continue
        img = cv2.imread(img_path)
        if img is None:
            print(f"cant find image path: {img_path}")
```

```
            continue
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  #converting to RGB
        axes[i].imshow(img)
        axes[i].axis("off")
    plt.tight_layout()
    plt.show()
plot_sample_images()
```

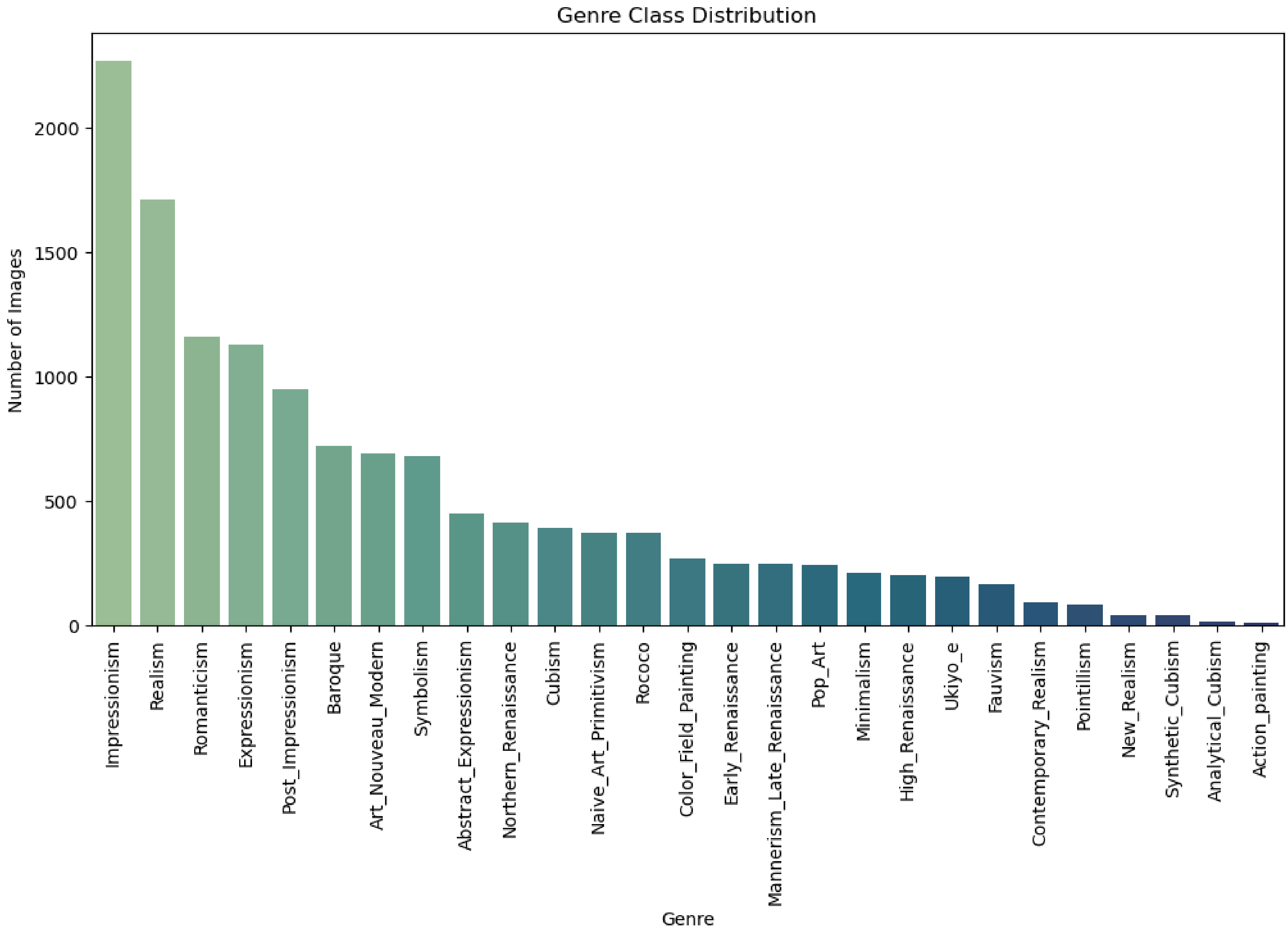
In [176…

```
genre_counts = {genre: len(os.listdir(os.path.join(train_dir, genre))) for genre in genre_classes}
sorted_genre_counts = dict(sorted(genre_counts.items(), key=lambda item: item[1], reverse=True))
```

In [178…

```
plt.figure(figsize=(12, 6))
sns.barplot(x=list(sorted_genre_counts.keys()), y=list(sorted_genre_counts.values()), palette="crest")
plt.xticks(rotation=90)
plt.xlabel("Genre")
plt.ylabel("Number of Images")
plt.title("Genre Class Distribution")
plt.show()
```

```
unique with argument that is not not a Series, Index, ExtensionArray, or np.ndarray is deprecated and will ra
ise in a future version.
```



In [80]:
```
top_5_genres = sorted(genre_counts.items(), key=lambda x: x[1], reverse=True)[:5]
top_5_genre_names = [genre for genre, _ in top_5_genres]
```

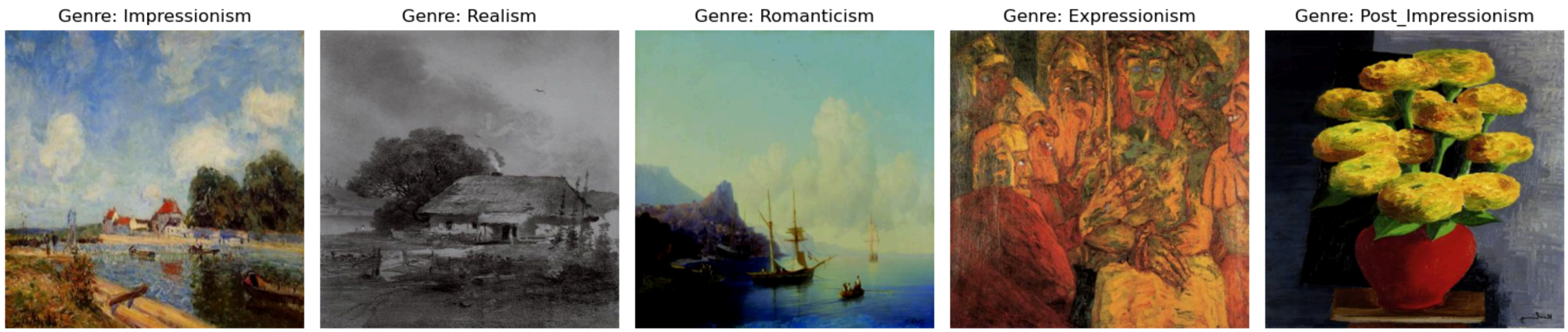
In [81]:
```
#top five genres
def visualize_top_5_genres(top_5_genre_names):
```

```
    fig, axes = plt.subplots(1, 5, figsize=(15, 5))  #subplot for five images
    for i, genre in enumerate(top_5_genre_names):
        genre_dir = os.path.join(train_dir, genre)
        genre_images = [os.path.join(genre_dir, img) for img in os.listdir(genre_dir) if img.lower().endswith
        random_img_path = random.choice(genre_images) #pulling random image from path
        img = cv2.imread(random_img_path) #reading image
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  #BGR to RGB
        axes[i].imshow(img) #plotting w. genre labels
        axes[i].axis("off")  #hiding axis
        axes[i].set_title(f"Genre: {genre}") #setting title to show example of genre

    plt.tight_layout()
    plt.show()
```

In [83]:
```
print(f'Example works for top 5 genres:')
visualize_top_5_genres(top_5_genre_names)
```

```
Example works for top 5 genres:
```

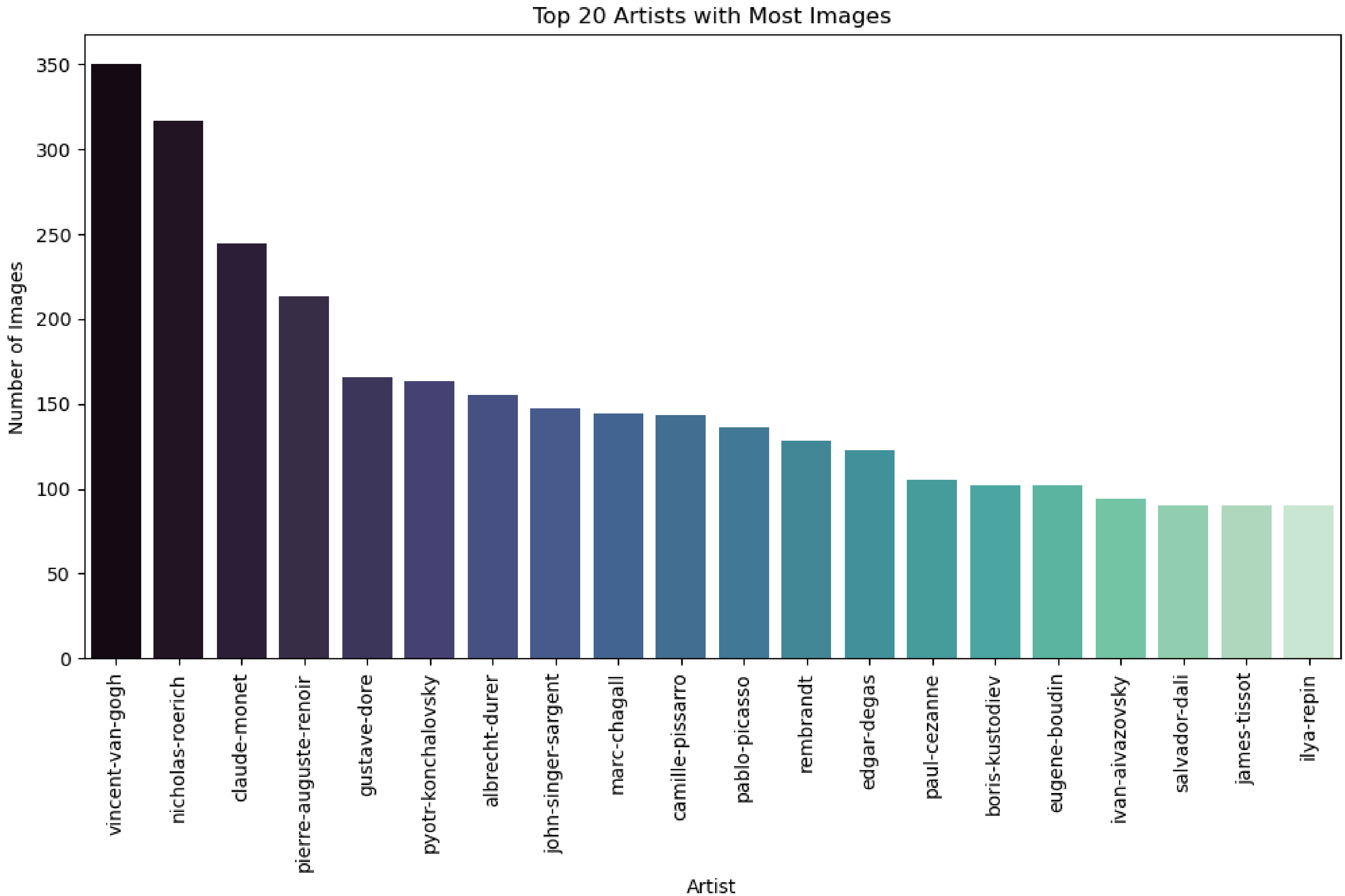


In [23]:
```
artist_counts = Counter([extract_artist_from_filename(f) for f in image_files])
```

In [24]:
```
top_artists = dict(sorted(artist_counts.items(), key=lambda x: x[1], reverse=True)[:20])
```

In [149…
```
#Top 20 artist plot
plt.figure(figsize=(12, 6))
sns.barplot(x=list(top_artists.keys()), y=list(top_artists.values()), palette="mako")
plt.xticks(rotation=90)
plt.xlabel("Artist")
plt.ylabel("Number of Images")
plt.title("Top 20 Artists with Most Images")
plt.show()
```

```
unique with argument that is not not a Series, Index, ExtensionArray, or np.ndarray is deprecated and will ra
ise in a future version.
```



In [26]:
```
#image_files
```

Model Build:

In [ ]:

In [29]:
```
early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True) #prevent overfittin
lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2, verbose=1) #reduce LR when val l
```

In [30]:
```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001) #stabalize training w. Adam
```

In [31]:
```
input_layer = Input(shape=(256, 256, 3)) #3 color channels
```

In [32]:
```
#CV Layer 1: 32 filters, 3x3 kernal, ReLu, L2 reg
x = Conv2D(32, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(0.01))(input_layer)
x = BatchNormalization()(x) #normalizeing
x = MaxPooling2D(pool_size=(2, 2))(x) #downsampling to reduce dimensions
#CV Layer 2: 64 filters, 3x3 kernal, Reul, L2 reg
x = Conv2D(64, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(0.01))(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2, 2))(x) #futher downsampling
x = GlobalAveragePooling2D()(x) #reduce feature maps to single vector
x = Dropout(0.5)(x)  #reduce overfitting w. 50% dropout
```

In [33]:
```
genre_output = Dense(len(genre_classes), activation='softmax', name='genre_output')(x) #genre outer layer
artist_output = Dense(len(artist_to_index), activation='softmax', name='artist_output')(x) #artists outer lay
```

In [34]:
```
model = Model(inputs=input_layer, outputs=[genre_output, artist_output]) #defining model (1 input layer, 2 ou
#compiling model
model.compile(
    optimizer=optimizer,
    loss={'genre_output': 'categorical_crossentropy', 'artist_output': 'categorical_crossentropy'},
    loss_weights={'genre_output': 1.0, 'artist_output': 1.0},
    metrics={'genre_output': 'accuracy', 'artist_output': 'accuracy'}
)
```

In [35]:
```
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 256, 256, 3)	0	-
conv2d (Conv2D)	(None, 254, 254, 32)	896	input_layer[0][0]
batch_normalization (BatchNormalizatio…	(None, 254, 254, 32)	128	conv2d[0][0]
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0	batch_normalizat…
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,496	max_pooling2d[0]…
batch_normalizatio… (BatchNormalizatio…	(None, 125, 125, 64)	256	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0	batch_normalizat…
global_average_poo… (GlobalAveragePool…	(None, 64)	0	max_pooling2d_1[…
dropout (Dropout)	(None, 64)	0	global_average_p…
genre_output (Dense)	(None, 27)	1,755	dropout[0][0]
artist_output (Dense)	(None, 851)	55,315	dropout[0][0]

Total params: 76,846 (300.18 KB)

Trainable params: 76,654 (299.43 KB)

Non-trainable params: 192 (768.00 B)

In [36]:
```
#artist_to_index
```

In [37]:
```
history = model.fit(
    train_ds,
    validation_data=valid_ds,
```

```
    epochs=20,
    callbacks=[early_stopping, lr_scheduler]  #callbacks for early stopping, Lr
)
```

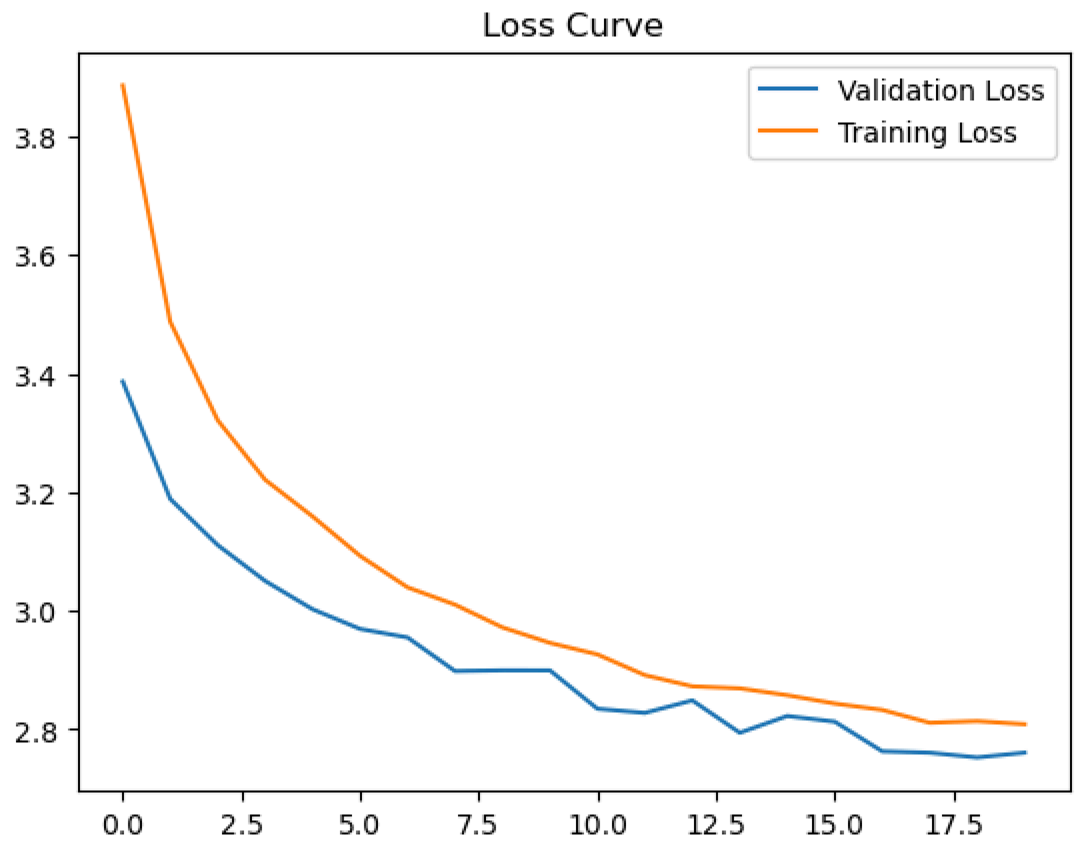
```
Epoch 1/20
Gradients do not exist for variables ['kernel', 'bias'] when minimizing the loss. If using `model.compile()`,
did you forget to provide a `loss` argument?
```

```
409/417 ━━━━━━━━━━━━━━━━━━━━ 39s 5s/step - genre_output_accuracy: 0.2180 - loss: 2.8168Batch 410 - Artist: gi
orgio-vasari
410/417 ━━━━━━━━━━━━━━━━━━━━ 34s 5s/step - genre_output_accuracy: 0.2180 - loss: 2.8167Batch 411 - Artist: ma
erten-van-heemskerck
411/417 ━━━━━━━━━━━━━━━━━━━━ 29s 5s/step - genre_output_accuracy: 0.2180 - loss: 2.8167Batch 412 - Artist: pi
ero-di-cosimo
412/417 ━━━━━━━━━━━━━━━━━━━━ 24s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8167Batch 413 - Artist: ag
nolo-bronzino
413/417 ━━━━━━━━━━━━━━━━━━━━ 19s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8167Batch 414 - Artist: ti
ntoretto
414/417 ━━━━━━━━━━━━━━━━━━━━ 14s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8167Batch 415 - Artist: ti
tian
415/417 ━━━━━━━━━━━━━━━━━━━━ 9s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8166 Batch 416 - Artist: ti
ntoretto
416/417 ━━━━━━━━━━━━━━━━━━━━ 4s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8166Batch 417 - Artist: pie
ro-di-cosimo
417/417 ━━━━━━━━━━━━━━━━━━━━ 0s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8166
Epoch 20 - Artists in this batch: william-h--johnson, william-h--johnson, amedeo-modigliani, toyen, pyotr-kon
chalovsky, stanley-pinker, saul-steinberg, henri-matisse, viorel-marginean, paula-modersohn-becker, egon-schi
ele, dimitris-mytaras, ernst-ludwig-kirchner, oswaldo-guayasamin, georges-braque, martiros-saryan, pyotr-konc
halovsky, kathe-kollwitz, wassily-kandinsky, albert-bloch, kees-van-dongen, amedeo-modigliani, salvador-dali,
martiros-saryan, mark-rothko, jean-helion, william-h--johnson, abraham-manievich, vasile-kazar, istvan-farka
s, amedeo-modigliani, ramon-oviedo
417/417 ━━━━━━━━━━━━━━━━━━━━ 2067s 5s/step - genre_output_accuracy: 0.2181 - loss: 2.8166 - val_genre_output_
accuracy: 0.2377 - val_loss: 2.7605 - learning_rate: 2.5000e-05
```

In [38]:
```
print(history.history.keys()) #history keys
```

```
dict_keys(['genre_output_accuracy', 'loss', 'val_genre_output_accuracy', 'val_loss', 'learning_rate'])
```

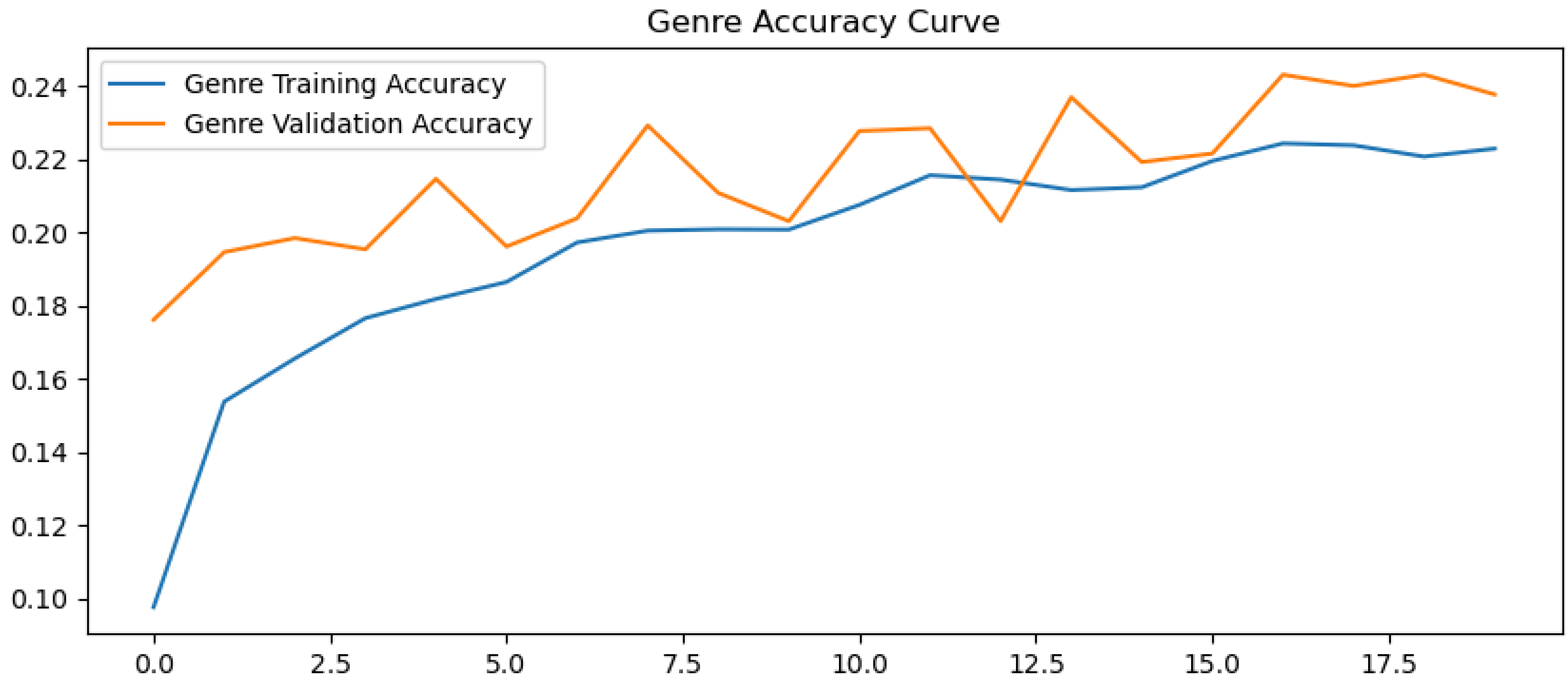
In [39]:
```
#loss curve
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.plot(history.history['loss'], label='Training Loss')
plt.legend()
plt.title("Loss Curve")
plt.show()
```



In [40]:
```
#evaluating model on valid_ds
results = model.evaluate(valid_ds)
print(f"Validation Results: {results}")
```

```
41/41 ━━━━━━━━━━━━━━━━━━━━ 34s 831ms/step - genre_output_accuracy: 0.2230 - loss: 2.9102
Validation Results: [2.752408742904663, 0.24307692050933838]
```

In [41]:
```
#genre_accuracy curve
plt.figure(figsize = (10, 4))
plt.plot(history.history['genre_output_accuracy'], label = 'Genre Training Accuracy')
plt.plot(history.history['val_genre_output_accuracy'], label = 'Genre Validation Accuracy')
plt.legend()
plt.title("Genre Accuracy Curve")
plt.show()
```



In [71]:
```
genre_predictions = model.predict(train_ds)
```

```
417/417 ━━━━━━━━━━━━━━━━━━━━ 423s 1s/step
```

In [72]:
```
print("Shape of first array:", genre_predictions[0].shape) #genre predictions
print("Shape of second array:", genre_predictions[1].shape) #artists predictions
```

```
Shape of first array: (13343, 27)
Shape of second array: (13343, 851)
```

In [73]:
```
genre_predictions, artist_predictions = genre_predictions #pulling artists_predictions from genre_predictions
```

In [74]:
```
print("Genre Predictions Shape:", genre_predictions.shape)  # Should be (13343, 27)
print("Artist Predictions Shape:", artist_predictions.shape)  # Should be (13343, 851)
```

```
Genre Predictions Shape: (13343, 27)
Artist Predictions Shape: (13343, 851)
```

In [75]:
```
predicted_genres = np.argmax(genre_predictions, axis = 1)
predicted_genre_labels = [genre_classes[idx] for idx in predicted_genres]
```

In [140…

```
#classification report for genre
genre_report = classification_report(genre_labels, predicted_genres, target_names=genre_classes, zero_divisio
print(genre_report)
```

```
                           precision    recall  f1-score   support

     Abstract_Expressionism       0.03      0.01      0.01       449
            Action_painting       0.00      0.00      0.00        11
          Analytical_Cubism       0.00      0.00      0.00        15
         Art_Nouveau_Modern       0.04      0.01      0.01       688
                    Baroque       0.06      0.05      0.06       721
         Color_Field_Painting       0.00      0.00      0.00       268
       Contemporary_Realism       0.00      0.00      0.00        91
                     Cubism       0.08      0.00      0.00       390
          Early_Renaissance       0.00      0.00      0.00       246
              Expressionism       0.09      0.08      0.08      1127
                    Fauvism       0.14      0.01      0.01       163
           High_Renaissance       0.00      0.00      0.00       198
              Impressionism       0.17      0.55      0.26      2269
 Mannerism_Late_Renaissance       0.00      0.00      0.00       246
                 Minimalism       0.01      0.01      0.01       212
      Naive_Art_Primitivism       0.00      0.00      0.00       373
                New_Realism       0.00      0.00      0.00        41
        Northern_Renaissance       0.02      0.01      0.01       413
                Pointillism       0.00      0.00      0.00        80
                    Pop_Art       0.01      0.00      0.01       244
         Post_Impressionism       0.08      0.03      0.04       946
                    Realism       0.13      0.23      0.17      1712
                     Rococo       0.00      0.00      0.00       369
                Romanticism       0.06      0.01      0.01      1157
                  Symbolism       0.05      0.01      0.02       679
           Synthetic_Cubism       0.00      0.00      0.00        38
                    Ukiyo_e       0.00      0.00      0.00       197

                   accuracy                           0.14     13343
                  macro avg       0.04      0.04      0.03     13343
               weighted avg       0.08      0.14      0.08     13343
```

In [105…

```
top_5_predicted_genres = np.argsort(genre_predictions, axis = 1)[:, -5:]
top_5_correct = [actual in preds for actual, preds in zip(genre_labels, top_5_predicted_genres)]
top_5_accuracy = np.mean(top_5_correct)
print(f"Top 5 Genre Accuracy: {top_5_accuracy:.2%}")
```

```
Top 5 Genre Accuracy: 46.92%
```

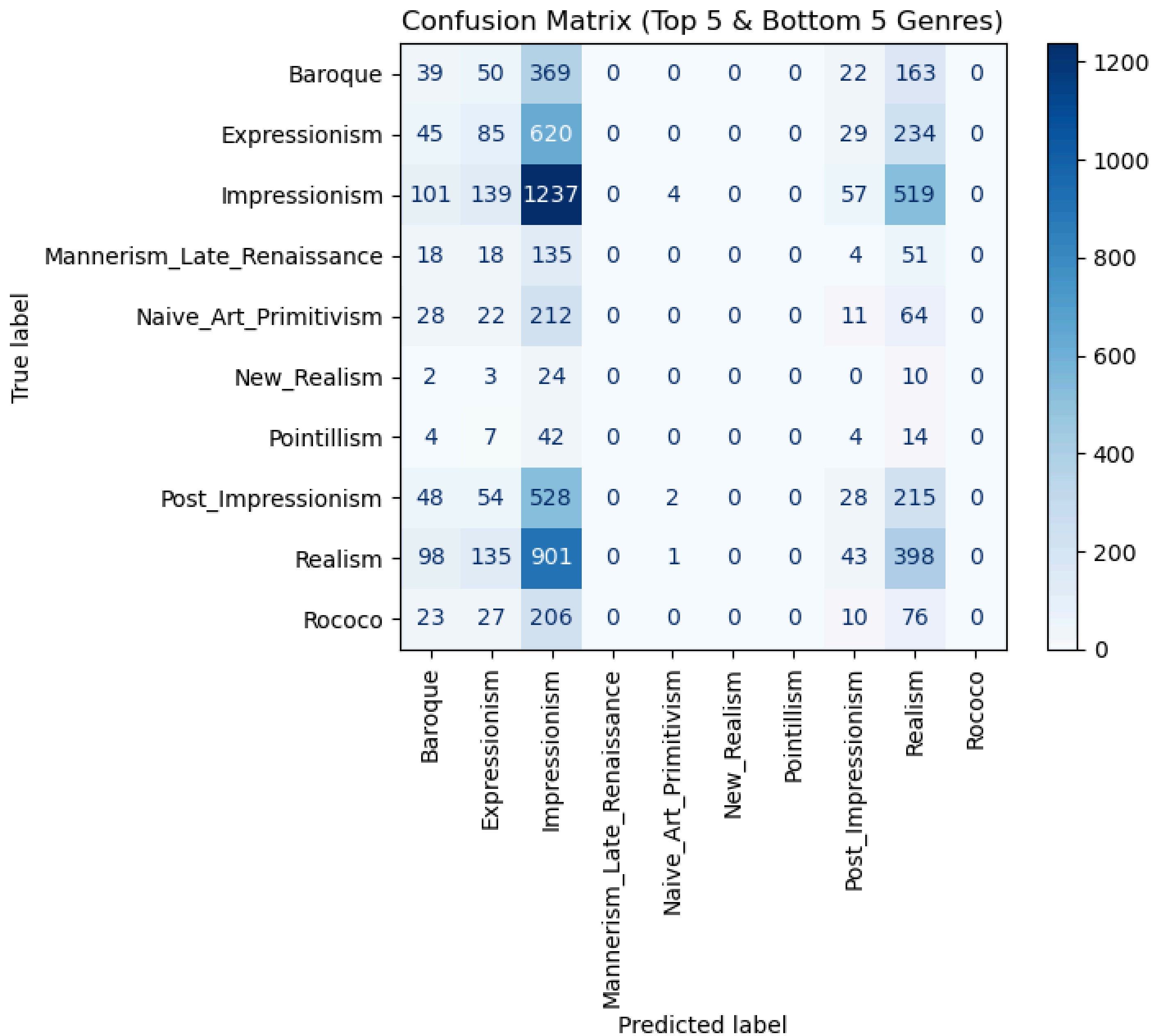
In [115…

```
#confusion matrix
genre_cm = confusion_matrix(genre_labels, predicted_genres) #confusion matrix for genre
genre_accuracy = np.diag(genre_cm) / genre_cm.sum(axis = 1)  #true positive/total class
sorted_genres = np.argsort(genre_accuracy) #sorting by accuracy
top_5_indices = sorted_genres[-5:]  #5 best performing genres
bottom_5_indices = sorted_genres[:5]  #5 worst performing genres
selected_indices = np.sort(np.concatenate([top_5_indices, bottom_5_indices]))
filtered_cm = genre_cm[selected_indices][:, selected_indices]
filtered_labels = [genre_classes[idx] for idx in selected_indices]
```

In [179…

```
#Plot of Confusion Matrix
plt.figure(figsize=(10, 15))
disp = ConfusionMatrixDisplay(confusion_matrix = filtered_cm, display_labels = filtered_labels)
disp.plot(cmap = "Blues", values_format=".0f", xticks_rotation = 90)
plt.title("Confusion Matrix (Top 5 & Bottom 5 Genres)")
plt.show()
```

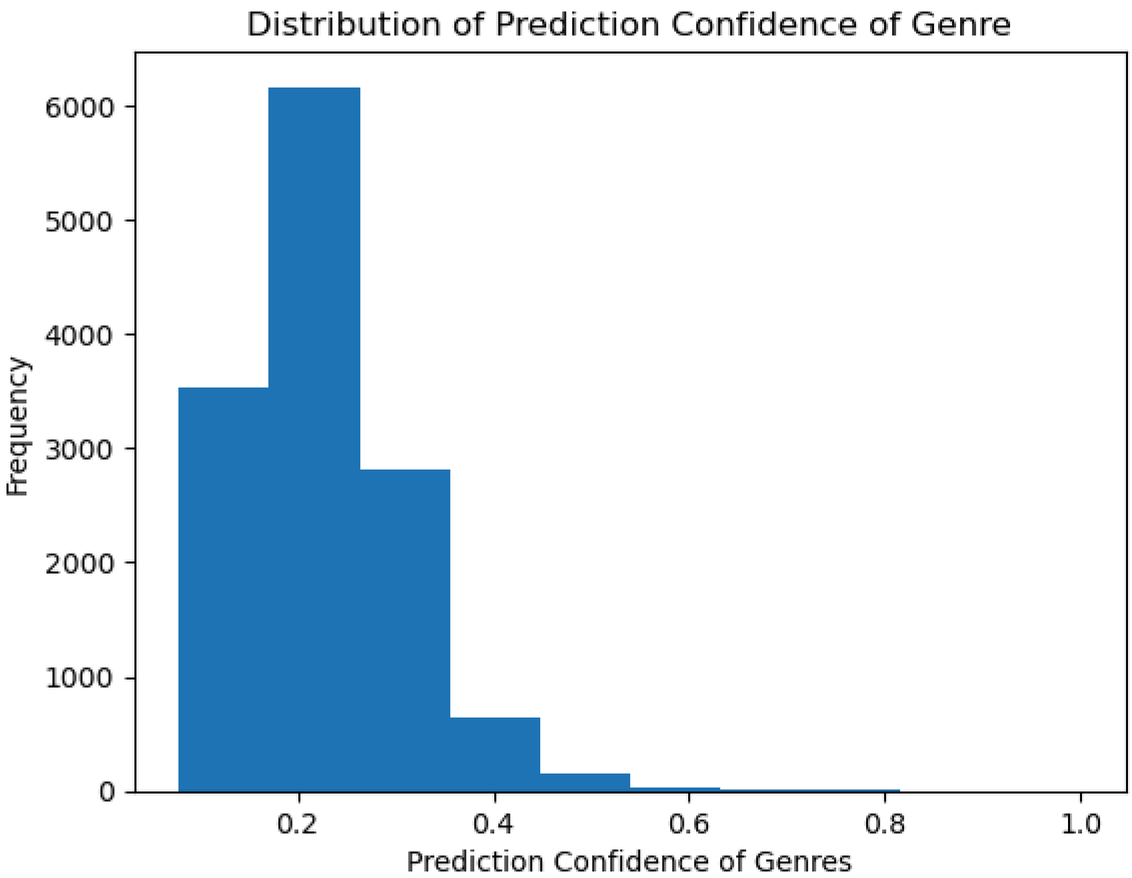
```
<Figure size 1000x1500 with 0 Axes>
```



In [146…
```
max_probs_genre = np.max(genre_predictions, axis=1) #max of genre predictions
```

In [147…
```
#plot of distribution of pred of genre
plt.hist(max_probs_genre, bins = 10)
```

```
plt.xlabel("Prediction Confidence of Genres")
plt.ylabel("Frequency")
plt.title("Distribution of Prediction Confidence of Genre")
plt.show()
```



In [182…

```
predicted_artists = np.argmax(artist_predictions, axis=1)
predicted_artist_labels = [artists[idx] for idx in predicted_artists]
```

In [183…

```
#Classification report for artists
artist_to_index = {artist: idx for idx, artist in enumerate(artists)}
true_artist_indices = [artist_to_index[artist] for artist in artist_labels]
artist_report = classification_report(true_artist_indices, predicted_artists, target_names=artists, zero_divi
print(artist_report)
```

	precision	recall	f1-score	support
jacopo-pontormo	0.00	0.00	0.00	12
jules-pascin	0.00	0.00	0.00	3
janos-mattis-teutsch	0.00	0.00	0.00	6
george-frederick-watts	0.00	0.00	0.00	6
francesco-hayez	0.00	0.00	0.00	21
olexandr-archipenko	0.00	0.00	0.00	18
iosif-iser	0.00	0.00	0.00	3
robert-nickle	0.00	0.00	0.00	3
jean-fautrier	0.00	0.00	0.00	3
willem-de-kooning	0.00	0.00	0.00	6
euga-ne-grasset	0.00	0.00	0.00	6
ivan-generalic	0.00	0.00	0.00	3
parmigianino	0.00	0.00	0.00	17
larry-zox	0.00	0.00	0.00	3
louis-vivin	0.00	0.00	0.00	9
paul-mathiopoulos	0.00	0.00	0.00	9
auguste-herbin	0.00	0.00	0.00	3
marie-laurencin	0.00	0.00	0.00	3
filipp-malyavin	0.00	0.00	0.00	6
antoine-pesne	0.00	0.00	0.00	21
saul-steinberg	0.00	0.00	0.00	12
konstantin-somov	0.00	0.00	0.00	33
arturo-souto	0.00	0.00	0.00	3
marjorie-strider	0.00	0.00	0.00	15
theodore-rousseau	0.00	0.00	0.00	18
theodoor-van-thulden	0.00	0.00	0.00	9
chronis-botsoglou	0.00	0.00	0.00	12
diego-rivera	0.00	0.00	0.00	3
stefan-dimitrescu	0.00	0.00	0.00	6
a-y--jackson	0.00	0.00	0.00	12
andre-bauchant	0.00	0.00	0.00	3
aleksey-savrasov	0.00	0.00	0.00	33
fyodor-rokotov	0.00	0.00	0.00	9
camille-bombois	0.00	0.00	0.00	3
alphonse-mucha	0.00	0.00	0.00	41
cy-twombly	0.00	0.00	0.00	25
bernhard-strigel	0.00	0.00	0.00	6
roy-lichtenstein	0.00	0.00	0.00	12
jean-marc-nattier	0.00	0.00	0.00	3
henry-raeburn	0.00	0.00	0.00	15
donatello	0.00	0.00	0.00	3
nikolaos-lytras	0.00	0.00	0.00	3
carlos-merida	0.00	0.00	0.00	3

henri-laurens	0.00	0.00	0.00	3
ion-theodorescu-sion	0.00	0.00	0.00	9
billy-apple	0.00	0.00	0.00	6
endre-bartos	0.00	0.00	0.00	6
anta-nio-de-carvalho-da-silva-porto	0.00	0.33	0.00	3
carlo-crivelli	0.00	0.00	0.00	21
dadamaino	0.00	0.00	0.00	3
vasily-sadovnikov	0.00	0.00	0.00	3
kateryna-bilokur	0.00	0.00	0.00	8
victor-brauner	0.00	0.00	0.00	6
efim-volkov	0.00	0.00	0.00	3
abdullah-suriosubroto	0.00	0.00	0.00	3
jean-helion	0.00	0.00	0.00	6
oswaldo-guayasamin	0.00	0.00	0.00	5
rudolf-von-alt	0.00	0.00	0.00	6
perle-fine	0.00	0.00	0.00	6
istvan-ilosvai-varga	0.00	0.00	0.00	6
alfred-manessier	0.00	0.00	0.00	3
juan-gris	0.00	0.00	0.00	24
brice-marden	0.00	0.00	0.00	12
homer-watson	0.00	0.00	0.00	6
ogata-gekko	0.00	0.00	0.00	3
albert-marquet	0.00	0.00	0.00	6
franz-kline	0.00	0.00	0.00	7
theophrastos-triantafyllidis	0.00	0.00	0.00	9
corneliu-baba	0.00	0.00	0.00	3
max-beckmann	0.00	0.00	0.00	8
sidney-nolan	0.00	0.00	0.00	3
gerard-terborch	0.00	0.00	0.00	6
utagawa-kuniyoshi	0.00	0.00	0.00	51
anna-ostroumova-lebedeva	0.00	0.00	0.00	12
sam-gilliam	0.00	0.00	0.00	15
eliseu-visconti	0.00	0.00	0.00	9
chuck-close	0.00	0.00	0.00	6
caspar-david-friedrich	0.00	0.00	0.00	3
taro-yamamoto	0.00	0.00	0.00	3
oleksandr-bogomazov	0.00	0.00	0.00	3
salvador-dali	0.00	0.00	0.00	90
tsukioka-yoshitoshi	0.00	0.00	0.00	12
jean-baptiste-simeon-chardin	0.00	0.00	0.00	12
vincent-van-gogh	0.00	0.00	0.00	350
mikhail-vrubel	0.00	0.00	0.00	35
filippo-lippi	0.00	0.00	0.00	15
filippo-brunelleschi	0.00	0.00	0.00	3
carlos-botelho	0.00	0.00	0.00	3

ion-andreescu	0.00	0.00	0.00	6
roger-fry	0.00	0.00	0.00	3
rafa-nasiri	0.00	0.00	0.00	3
pisanello	0.00	0.00	0.00	3
cornelis-de-vos	0.00	0.00	0.00	3
lyonel-feininger	0.00	0.00	0.00	11
william-holman-hunt	0.00	0.00	0.00	12
t--c--steele	0.00	0.00	0.00	9
leon-spilliaert	0.00	0.00	0.00	3
george-mavroides	0.00	0.00	0.00	3
jan-siberechts	0.00	0.00	0.00	6
august-macke	0.00	0.00	0.00	12
john-mccracken	0.00	0.00	0.00	3
william-congdon	0.00	0.00	0.00	6
thomas-gainsborough	0.00	0.00	0.00	21
louis-cane	0.00	0.00	0.00	6
zinaida-serebriakova	0.00	0.00	0.00	66
konstantin-korovin	0.00	0.00	0.00	39
peter-paul-rubens	0.00	0.00	0.00	69
fikret-mualla-saygi	0.00	0.00	0.00	8
gosta-adrian-nilsson	0.00	0.00	0.00	6
leon-berkowitz	0.00	0.00	0.00	9
berthe-morisot	0.00	0.00	0.00	26
john-french-sloan	0.00	0.00	0.00	8
jose-clemente-orozco	0.00	0.00	0.00	3
willard-metcalf	0.00	0.00	0.00	9
constantin-piliuta	0.00	0.00	0.00	6
stuart-davis	0.00	0.00	0.00	3
manabu-mabe	0.00	0.00	0.00	3
johannes-itten	0.00	0.00	0.00	6
friedel-dzubas	0.00	0.00	0.00	17
maxime-lalanne	0.00	0.00	0.00	3
ivan-bilibin	0.00	0.00	0.00	30
hiro-yamagata	0.00	0.00	0.00	21
andrea-solario	0.00	0.00	0.00	6
wilhelm-kotarbinski	0.00	0.00	0.00	21
meijer-de-haan	0.00	0.00	0.00	3
hans-holbein-the-younger	0.00	0.00	0.00	18
giovanni-bellini	0.00	0.00	0.00	12
aladar-korosfoi-kriesch	0.00	0.00	0.00	6
pavel-fedotov	0.00	0.00	0.00	3
vajda-lajos	0.00	0.00	0.00	6
titian	0.00	0.00	0.00	36
alexander-liberman	0.00	0.00	0.00	3
geta-bratescu	0.00	0.00	0.00	9

domingos-sequeira	0.00	0.00	0.00	3
julio-gonzalez	0.00	0.00	0.00	9
afro	0.00	0.00	0.00	3
nicolas-tournier	0.00	0.00	0.00	3
nicolae-tonitza	0.00	0.00	0.00	27
koloman-moser	0.00	0.00	0.00	32
petre-abrudan	0.00	0.00	0.00	3
dirk-bouts	0.00	0.00	0.00	18
hubert-robert	0.00	0.00	0.00	6
samuel-mutzner	0.00	0.00	0.00	9
katsushika-hokusai	0.00	0.00	0.00	27
fyodor-solntsev	0.00	0.00	0.00	3
henry-herbert-la-thangue	0.00	0.00	0.00	9
micaela-eleutheriade	0.00	0.00	0.00	3
pietro-perugino	0.00	0.00	0.00	36
robert-silvers	0.00	0.00	0.00	6
doug-ohlson	0.00	0.00	0.00	3
mabuse	0.00	0.00	0.00	3
vicente-manansala	0.00	0.00	0.00	9
li-yuan-chia	0.00	0.00	0.00	3
paul-signac	0.00	0.00	0.00	6
jose-guerrero	0.00	0.00	0.00	3
vasily-tropinin	0.00	0.00	0.00	12
rogier-van-der-weyden	0.00	0.00	0.00	21
vasily-perov	0.00	0.00	0.00	12
philip-wilson-steer	0.00	0.00	0.00	12
thomas-moran	0.00	0.00	0.00	11
albert-bierstadt	0.00	0.00	0.00	3
felix-vallotton	0.00	0.00	0.00	21
ion-tuculescu	0.00	0.00	0.00	12
hans-hofmann	0.00	0.00	0.00	19
aubrey-beardsley	0.00	0.00	0.00	24
mostafa-dashti	0.00	0.00	0.00	3
vladimir-tatlin	0.00	0.00	0.00	3
serge-sudeikin	0.00	0.00	0.00	9
franz-richard-unterberger	0.00	0.00	0.00	12
miklos-barabas	0.00	0.00	0.00	3
vasile-popescu	0.00	0.00	0.00	3
yayoi-kusama	0.00	0.00	0.00	3
jules-lefranc	0.00	0.00	0.00	6
bernardo-strozzi	0.00	0.00	0.00	6
kitagawa-utamaro	0.00	0.00	0.00	33
anita-malfatti	0.00	0.00	0.00	12
marevna--marie-vorobieff-	0.00	0.00	0.00	9
adriaen-van-de-venne	0.00	0.00	0.00	3

tsuguharu-foujita	0.00	0.00	0.00	9
christen-kobke	0.00	0.00	0.00	6
maxime-maufra	0.00	0.00	0.00	30
dumitru-ghiatza	0.00	0.00	0.00	6
maria-primachenko	0.00	0.00	0.00	9
gerrit-dou	0.00	0.00	0.00	21
fred-yates	0.00	0.00	0.00	3
jose-de-almada-negreiros	0.00	0.00	0.00	3
agnolo-bronzino	0.00	0.00	0.00	15
johannes-vermeer	0.00	0.00	0.00	12
joachim-wtewael	0.00	0.00	0.00	3
carlos-saenz-de-tejada	0.00	0.00	0.00	3
hans-von-aachen	0.00	0.00	0.00	6
alekos-kontopoulos	0.00	0.00	0.00	6
david-batchelor	0.00	0.00	0.00	6
rik-wouters	0.00	0.00	0.00	6
konstantin-vasilyev	0.00	0.00	0.00	9
marc-chagall	0.03	0.01	0.01	144
lee-ufan	0.00	0.00	0.00	6
piero-manzoni	0.00	0.00	0.00	3
pedro-calapez	0.00	0.00	0.00	3
sol-lewitt	0.00	0.00	0.00	3
m-c--escher	0.00	0.00	0.00	24
theodoros-stamos	0.00	0.00	0.00	16
andrea-mantegna	0.00	0.00	0.00	23
andra--lhote	0.00	0.00	0.00	6
guntis-strupulis	0.00	0.00	0.00	3
richard-diebenkorn	0.00	0.00	0.00	15
ad-reinhardt	0.00	0.00	0.00	6
julio-pomar	0.00	0.00	0.00	12
michelangelo	0.00	0.00	0.00	24
ronald-davis	0.00	0.00	0.00	3
aelbert-cuyp	0.00	0.00	0.00	6
keisai-eisen	0.00	0.00	0.00	24
john-hoppner	0.00	0.00	0.00	9
karl-schrag	0.00	0.00	0.00	6
hans-memling	0.00	0.00	0.00	9
fyodor-vasilyev	0.00	0.00	0.00	18
john-lewis-krimmel	0.00	0.00	0.00	6
max-pechstein	0.00	0.00	0.00	24
john-mclaughlin	0.00	0.00	0.00	6
odilon-redon	0.00	0.00	0.00	81
robert-goodnough	0.00	0.00	0.00	6
karoly-ferenczy	0.00	0.00	0.00	12
richard-gerstl	0.00	0.00	0.00	18

emily-carr	0.00	0.00	0.00	3
emil-nolde	0.00	0.00	0.00	6
pauline-boty	0.00	0.00	0.00	6
alexander-ivanov	0.00	0.00	0.00	15
hercules-seghers	0.00	0.00	0.00	3
willem-kalf	0.00	0.00	0.00	3
rosso-fiorentino	0.00	0.00	0.00	3
richard-pousette-dart	0.00	0.00	0.00	18
leon-bakst	0.00	0.00	0.00	12
john-william-waterhouse	0.00	0.00	0.00	18
taras-shevchenko	0.00	0.00	0.00	21
robert-julian-onderdonk	0.00	0.00	0.00	18
jean-paul-lemieux	0.00	0.00	0.00	3
michel-carrade	0.00	0.00	0.00	3
fyodor-bronnikov	0.00	0.00	0.00	2
paolo-veronese	0.00	0.00	0.00	47
alfred-kubin	0.00	0.00	0.00	3
thomas-eakins	0.00	0.00	0.00	48
adriaen-van-de-velde	0.00	0.00	0.00	3
joan-miro	0.00	0.00	0.00	15
nutzi-acontz	0.00	0.00	0.00	6
hans-baldung	0.00	0.00	0.00	12
radi-nedelchev	0.00	0.00	0.00	3
vasily-surikov	0.00	0.00	0.00	35
edward-hopper	0.00	0.00	0.00	18
david-teniers-the-younger	0.00	0.00	0.00	11
pierre-bonnard	0.00	0.00	0.00	15
park-seo-bo	0.00	0.00	0.00	6
benozzo-gozzoli	0.00	0.00	0.00	18
giacomo-balla	0.00	0.00	0.00	3
pericles-pantazis	0.00	0.00	0.00	12
john-singer-sargent	0.00	0.00	0.00	147
max-liebermann	0.00	0.00	0.00	15
henri-martin	0.00	0.00	0.00	15
george-inness	0.00	0.00	0.00	3
raoul-ubac	0.00	0.00	0.00	3
hendrick-cornelisz-vroom	0.00	0.00	0.00	3
alfred-stevens	0.00	0.00	0.00	9
david-burliuk	0.00	0.00	0.00	51
klavdy-lebedev	0.00	0.00	0.00	3
albrecht-altdorfer	0.00	0.00	0.00	17
cima-da-conegliano	0.00	0.00	0.00	3
julian-alden-weir	0.00	0.00	0.00	6
jack-tworkov	0.00	0.00	0.00	3
alexander-calder	0.00	0.00	0.00	6

victor-borisov-musatov	0.00	0.00	0.00	12
sandro-botticelli	0.00	0.00	0.00	27
bui-xuan-phai	0.00	0.00	0.00	3
theodor-pallady	0.00	0.00	0.00	18
jan-provoost	0.00	0.00	0.00	3
joseph-wright	0.00	0.00	0.00	30
morris-louis	0.00	0.00	0.00	9
candido-portinari	0.00	0.00	0.00	3
frans-hals	0.00	0.00	0.00	35
jacob-jordaens	0.00	0.00	0.00	33
wassily-kandinsky	0.00	0.00	0.00	12
red-grooms	0.00	0.00	0.00	3
edward-avedisian	0.00	0.00	0.00	6
ellsworth-kelly	0.00	0.00	0.00	18
andrei-ryabushkin	0.00	0.00	0.00	12
kazimir-malevich	0.00	0.00	0.00	9
sergey-solomko	0.00	0.00	0.00	9
dimitris-mytaras	0.00	0.00	0.00	21
konstantin-makovsky	0.00	0.00	0.00	45
olga-rozanova	0.00	0.00	0.00	3
burhan-dogancay	0.00	0.00	0.00	9
gerard-fromanger	0.00	0.00	0.00	3
richard-tuttle	0.00	0.00	0.00	15
pietro-da-cortona	0.00	0.00	0.00	3
henryk-siemiradzki	0.00	0.00	0.00	3
nikola-tanev	0.00	0.00	0.00	3
lynd-ward	0.00	0.00	0.00	3
piero-di-cosimo	0.00	0.00	0.00	6
lucian-freud	0.00	0.00	0.00	51
panayiotis-tetsis	0.00	0.00	0.00	3
felicien-rops	0.00	0.00	0.00	15
niko-pirosmani	0.00	0.00	0.00	24
peter-max	0.00	0.00	0.00	6
alonzo-cano	0.00	0.00	0.00	3
albert-bloch	0.00	0.00	0.00	15
lucas-cranach-the-elder	0.00	0.00	0.00	35
man-ray	0.00	0.00	0.00	12
piet-mondrian	0.00	0.00	0.00	3
grigoriy-myasoyedov	0.00	0.00	0.00	12
gilles-aillaud	0.00	0.00	0.00	3
john-atkinson-grimshaw	0.00	0.00	0.00	21
gwen-john	0.00	0.00	0.00	9
chaim-soutine	0.00	0.00	0.00	6
kees-van-dongen	0.00	0.00	0.00	9
daniel-buren	0.00	0.00	0.00	3

franz-stuck	0.00	0.00	0.00	9
tintoretto	0.00	0.00	0.00	57
ivan-kramskoy	0.00	0.00	0.00	21
jacopo-bellini	0.00	0.00	0.00	6
hiroshige	0.00	0.00	0.00	18
pietro-longhi	0.00	0.00	0.00	3
janet-fish	0.00	0.00	0.00	3
robert-ryman	0.00	0.00	0.00	3
barnett-newman	0.00	0.00	0.00	21
morris-graves	0.00	0.00	0.00	5
arshile-gorky	0.00	0.00	0.00	6
angelo-de-sousa	0.00	0.00	0.00	3
allen-jones	0.00	0.00	0.00	3
pierre-auguste-renoir	0.05	0.00	0.01	213
blinky-palermo	0.00	0.00	0.00	3
fairfield-porter	0.00	0.00	0.00	11
jose-gutierrez-solana	0.00	0.00	0.00	6
gustave-courbet	0.00	0.00	0.00	39
john-chamberlain	0.00	0.00	0.00	3
peder-severin-kroyer	0.00	0.00	0.00	6
pierre-soulages	0.00	0.00	0.00	4
helene-schjerfbeck	0.00	0.00	0.00	3
andy-warhol	0.00	0.00	0.00	56
le-nain-brothers	0.00	0.00	0.00	12
jean-hey	0.00	0.00	0.00	3
john-constable	0.00	0.00	0.00	33
ivan-aivazovsky	0.00	0.00	0.00	94
joshua-reynolds	0.00	0.00	0.00	35
francois-boucher	0.00	0.00	0.00	9
rudolf-schweitzer-cumpana	0.00	0.00	0.00	12
paul-klee	0.00	0.00	0.00	36
francisco-goya	0.00	0.00	0.00	54
raoul-dufy	0.00	0.00	0.00	27
francesco-guardi	0.00	0.00	0.00	9
eva-hesse	0.00	0.00	0.00	3
mihaly-munkacsy	0.00	0.00	0.00	6
michelangelo-pistoletto	0.00	0.00	0.00	3
el-greco	0.00	0.00	0.00	15
aldemir-martins	0.00	0.00	0.00	5
jury-annenkov	0.00	0.00	0.00	9
martial-raysse	0.00	0.00	0.00	3
isaac-levitan	0.00	0.00	0.00	82
henri-edmond-cross	0.00	0.00	0.00	21
yves-klein	0.00	0.00	0.00	9
frank-lobdell	0.00	0.00	0.00	3

georges-vantongerloo	0.00	0.00	0.00	3
frank-johnston	0.00	0.00	0.00	3
ernst-ludwig-kirchner	0.00	0.00	0.00	60
boris-kustodiev	0.00	0.00	0.00	102
charles-francois-daubigny	0.00	0.00	0.00	12
miriam-schapiro	0.00	0.00	0.00	6
bela-czobel	0.00	0.00	0.00	12
maxim-vorobiev	0.00	0.00	0.00	3
asgrimur-jonsson	0.00	0.00	0.00	6
amedeo-modigliani	0.00	0.00	0.00	69
valentin-serov	0.00	0.00	0.00	42
dante-gabriel-rossetti	0.00	0.00	0.00	29
martiros-saryan	0.00	0.00	0.00	81
richard-hamilton	0.00	0.00	0.00	3
sir-lawrence-alma-tadema	0.00	0.00	0.00	48
john-lavery	0.00	0.00	0.00	3
jan-steen	0.00	0.00	0.00	18
lev-lagorio	0.00	0.00	0.00	18
niki-de-sainte-phalle	0.00	0.00	0.00	3
j--e--h--macdonald	0.00	0.00	0.00	3
gerhard-richter	0.00	0.00	0.00	6
santiago-rusinol	0.00	0.00	0.00	12
edmund-charles-tarbell	0.00	0.00	0.00	8
walasse-ting	0.00	0.00	0.00	15
jean-metzinger	0.00	0.00	0.00	3
frantisek-kupka	0.00	0.00	0.00	6
armand-guillaumin	0.00	0.00	0.00	24
n-c--wyeth	0.00	0.00	0.00	3
valerio-adami	0.00	0.00	0.00	3
william-merritt-chase	0.00	0.00	0.00	81
jozsef-rippl-ronai	0.00	0.00	0.00	9
edgar-degas	0.00	0.00	0.00	123
le-pho	0.00	0.00	0.00	3
juan-de-valdes-leal	0.00	0.00	0.00	9
adolphe-joseph-thomas-monticelli	0.00	0.00	0.00	6
giovanni-boldini	0.00	0.00	0.00	60
constant-troyon	0.00	0.00	0.00	3
benjamin-west	0.00	0.00	0.00	17
franklin-carmichael	0.00	0.00	0.00	6
jack-bush	0.00	0.00	0.00	21
joachim-patinir	0.00	0.00	0.00	3
arthur-hughes	0.00	0.00	0.00	6
gustave-loiseau	0.00	0.00	0.00	44
otto-gustav-carlsund	0.00	0.00	0.00	6
yov-kondzelevych	0.00	0.00	0.00	3

jackson-pollock	0.00	0.00	0.00	6
augustus-john	0.00	0.00	0.00	15
william-hogarth	0.00	0.00	0.00	29
eugene-boudin	0.00	0.00	0.00	102
marie-bracquemond	0.00	0.00	0.00	6
constantine-maleas	0.00	0.00	0.00	6
medi-wechsler-dinu	0.00	0.00	0.00	6
albrecht-durer	0.00	0.00	0.00	155
jan-van-hemessen	0.00	0.00	0.00	9
john-collier	0.00	0.00	0.00	9
kenzo-okada	0.00	0.00	0.00	3
john-everett-millais	0.00	0.00	0.00	20
thomas-cole	0.00	0.00	0.00	30
isaac-van-ostade	0.00	0.00	0.00	6
robert-indiana	0.00	0.00	0.00	3
frank-stella	0.00	0.00	0.00	26
jean-honore-fragonard	0.00	0.00	0.00	21
edward-ruscha	0.00	0.00	0.00	3
yves-gaucher	0.00	0.00	0.00	9
ossip-zadkine	0.00	0.00	0.00	9
georges-lemmen	0.00	0.00	0.00	6
milton-resnick	0.00	0.00	0.00	3
thomas-theodor-heine	0.00	0.00	0.00	9
theodore-chasseriau	0.00	0.00	0.00	3
frederic-bazille	0.00	0.00	0.00	15
walter-battiss	0.00	0.00	0.00	23
vasily-vereshchagin	0.00	0.00	0.00	24
caravaggio	0.00	0.00	0.00	15
leonardo-da-vinci	0.00	0.00	0.00	27
edouard-cortes	0.00	0.00	0.00	33
giovanni-domenico-tiepolo	0.00	0.00	0.00	9
jan-matejko	0.00	0.00	0.00	20
david-wilkie	0.00	0.00	0.00	6
romare-bearden	0.00	0.00	0.00	3
pieter-bruegel-the-elder	0.00	0.00	0.00	12
octav-angheluta	0.00	0.00	0.00	3
martin-johnson-heade	0.00	0.00	0.00	3
vladimir-borovikovsky	0.00	0.00	0.00	9
dmitry-levitzky	0.00	0.00	0.00	18
julius-leblanc-stewart	0.00	0.00	0.00	6
alfred-sisley	0.00	0.00	0.00	90
vasile-kazar	0.00	0.00	0.00	9
istvan-nagy	0.00	0.00	0.00	3
paolo-uccello	0.00	0.00	0.00	12
lovis-corinth	0.00	0.00	0.00	33

gustav-klimt	0.00	0.00	0.00	18
jay-defeo	0.00	0.00	0.00	3
bartolome-esteban-murillo	0.00	0.00	0.00	23
abraham-manievich	0.00	0.00	0.00	6
rene-magritte	0.00	0.00	0.00	3
josefa-de-obidos	0.00	0.00	0.00	3
alexey-venetsianov	0.00	0.00	0.00	9
andrea-del-verrocchio	0.00	0.00	0.00	9
pyotr-konchalovsky	0.00	0.00	0.00	163
dan-flavin	0.00	0.00	0.00	9
keith-sonnier	0.00	0.00	0.00	6
adam-baltatu	0.00	0.00	0.00	9
ivan-nikitin	0.00	0.00	0.00	9
henri-matisse	0.00	0.00	0.00	76
eric-fischl	0.00	0.00	0.00	14
fernando-calhau	0.00	0.00	0.00	3
andrea-del-castagno	0.00	0.00	0.00	15
guy-rose	0.00	0.00	0.00	24
pablo-picasso	0.00	0.00	0.00	136
walter-de-maria	0.00	0.00	0.00	3
mykola-yaroshenko	0.00	0.00	0.00	6
michael-bell	0.00	0.00	0.00	3
ben-nicholson	0.00	0.00	0.00	6
paul-jenkins	0.00	0.00	0.00	24
ralph-hotere	0.00	0.00	0.00	9
dan-christensen	0.00	0.00	0.00	12
paula-modersohn-becker	0.00	0.00	0.00	11
lourdes-castro	0.00	0.00	0.00	3
victor-meirelles	0.00	0.00	0.00	3
mario-zanini	0.00	0.00	0.00	3
piero-dorazio	0.00	0.00	0.00	5
joaqua-n-sorolla	0.00	0.00	0.00	39
nikolay-bogdanov-belsky	0.00	0.00	0.00	48
edvard-munch	0.00	0.00	0.00	27
toyen	0.00	0.00	0.00	12
edwin-henry-landseer	0.00	0.00	0.00	6
nicolae-darascu	0.00	0.00	0.00	9
eduardo-paolozzi	0.00	0.00	0.00	3
nikos-hadjikyriakos-ghikas	0.00	0.00	0.00	12
kay-nielsen	0.00	0.00	0.00	3
gra-goire-michonze	0.00	0.00	0.00	12
fernand-khnopff	0.00	0.00	0.00	9
eugene-delacroix	0.00	0.00	0.00	15
gregoire-boonzaier	0.00	0.00	0.00	29
anne-appleby	0.00	0.00	0.00	6

istvan-farkas	0.00	0.00	0.00	3
antoine-watteau	0.00	0.00	0.00	24
lajos-tihanyi	0.00	0.00	0.00	3
edward-burne-jones	0.00	0.00	0.00	27
marcel-janco	0.00	0.00	0.00	3
tarsila-do-amaral	0.00	0.00	0.00	9
nzante-spee	0.00	0.00	0.00	3
francisco-de-zurbaran	0.00	0.00	0.00	18
egon-schiele	0.00	0.00	0.00	54
correggio	0.00	0.00	0.00	12
james-ensor	0.00	0.00	0.00	6
matthias-stom	0.00	0.00	0.00	3
maurice-de-vlaminck	0.00	0.00	0.00	3
franz-marc	0.00	0.00	0.00	24
luca-signorelli	0.00	0.00	0.00	9
james-tissot	0.00	0.00	0.00	90
frits-thaulow	0.00	0.00	0.00	8
karl-bodmer	0.00	0.00	0.00	9
george-catlin	0.00	0.00	0.00	6
horia-damian	0.00	0.00	0.00	3
frederic-edwin-church	0.00	0.00	0.00	6
louis-schanker	0.00	0.00	0.00	6
george-bouzianis	0.00	0.00	0.00	24
kazuo-nakamura	0.00	0.00	0.00	2
utagawa-kunisada	0.00	0.00	0.00	12
milton-avery	0.00	0.00	0.00	12
johan-christian-dahl	0.00	0.00	0.00	3
umberto-boccioni	0.00	0.00	0.00	6
paul-serusier	0.00	0.00	0.00	6
amadeo-de-souza-cardoso	0.00	0.00	0.00	14
karl-bryullov	0.00	0.00	0.00	24
ion-nicodim	0.00	0.00	0.00	3
octav-bancila	0.00	0.00	0.00	18
ferdinand-georg-waldma-ller	0.00	0.00	0.00	6
nicolae-vermont	0.00	0.00	0.00	17
jose-de-guimaraes	0.00	0.00	0.00	3
paul-brach	0.00	0.00	0.00	12
lorenzo-lotto	0.00	0.00	0.00	24
piero-della-francesca	0.00	0.00	0.00	20
vasily-polenov	0.00	0.00	0.00	36
menez	0.00	0.00	0.00	6
derek-boshier	0.00	0.00	0.00	6
natalia-goncharova	0.01	0.33	0.01	3
jusepe-de-ribera	0.00	0.00	0.00	3
pat-lipsky	0.00	0.00	0.00	3

paul-gauguin	0.00	0.00	0.00	59
frank-bowling	0.00	0.00	0.00	3
stanley-pinker	0.00	0.00	0.00	3
anthony-van-dyck	0.00	0.00	0.00	30
jose-malhoa	0.00	0.00	0.00	3
john-marin	0.00	0.00	0.00	6
rufino-tamayo	0.00	0.00	0.00	6
maerten-van-heemskerck	0.00	0.00	0.00	14
laszlo-mednyanszky	0.00	0.00	0.00	9
otto-dix	0.00	0.00	0.00	12
pierre-paul-prud-hon	0.00	0.00	0.00	9
ferdinand-hodler	0.00	0.00	0.00	45
norman-bluhm	0.00	0.00	0.00	3
joan-snyder	0.00	0.00	0.00	3
masaccio	0.00	0.00	0.00	3
maurice-prendergast	0.00	0.00	0.00	48
louis-marcoussis	0.00	0.00	0.00	3
joan-mitchell	0.00	0.00	0.00	3
jimmy-ernst	0.00	0.00	0.00	15
auguste-rodin	0.00	0.00	0.00	15
frederic-remington	0.00	0.00	0.00	5
mary-fedden	0.00	0.00	0.00	9
canaletto	0.00	0.00	0.00	26
henri-fantin-latour	0.00	0.00	0.00	71
vilmos-aba-novak	0.00	0.00	0.00	6
adriaen-brouwer	0.00	0.00	0.00	6
jamie-wyeth	0.00	0.00	0.00	18
ito-shinsui	0.00	0.00	0.00	2
corneliu-michailescu	0.00	0.00	0.00	3
annibale-carracci	0.00	0.00	0.00	15
elaine-de-kooning	0.00	0.00	0.00	18
imi-knoebel	0.00	0.00	0.00	3
vladimir-makovsky	0.00	0.00	0.00	21
spyros-papaloukas	0.00	0.00	0.00	8
theo-van-doesburg	0.00	0.00	0.00	16
jacques-villon	0.00	0.00	0.00	6
francis-picabia	0.00	0.00	0.00	12
guido-reni	0.00	0.00	0.00	18
raphael	0.00	0.00	0.00	33
gustave-moreau	0.00	0.00	0.00	15
irma-stern	0.00	0.00	0.00	12
childe-hassam	0.04	0.01	0.02	90
william-h--johnson	0.00	0.00	0.00	33
antonio-palolo	0.00	0.00	0.00	3
konstantinos-parthenis	0.00	0.00	0.00	12

mstislav-dobuzhinsky	0.00	0.00	0.00	26
ivan-milev	0.00	0.00	0.00	6
anders-zorn	0.00	0.00	0.00	15
lyubov-popova	0.00	0.00	0.00	3
jean-david	0.00	0.00	0.00	9
maurice-denis	0.00	0.00	0.00	15
henri-de-toulouse-lautrec	0.00	0.00	0.00	81
william-scott	0.00	0.00	0.00	9
carl-bloch	0.00	0.00	0.00	3
ivan-shishkin	0.00	0.00	0.00	87
raphael-kirchner	0.00	0.00	0.00	68
volodymyr-orlovsky	0.00	0.00	0.00	21
john-roddam-spencer-stanhope	0.00	0.00	0.00	9
theodore-gericault	0.00	0.00	0.00	26
howard-mehring	0.00	0.00	0.00	3
m--h--maxy	0.00	0.00	0.00	3
john-hoyland	0.00	0.00	0.00	9
thalia-flora-karavia	0.00	0.00	0.00	9
giovanni-paolo-panini	0.00	0.00	0.00	3
giorgio-vasari	0.00	0.00	0.00	9
wilhelm-leibl	0.00	0.00	0.00	9
cornelis-springer	0.00	0.00	0.00	15
antonio-ligabue	0.00	0.00	0.00	12
rafael-zabaleta	0.00	0.00	0.00	3
max-ernst	0.00	0.00	0.00	6
patrick-caulfield	0.00	0.00	0.00	6
pierre-tal-coat	0.00	0.00	0.00	3
konstantinos-volanakis	0.00	0.00	0.00	9
vasile-dobrian	0.00	0.00	0.00	12
sa-nogueira	0.00	0.00	0.00	3
leopold-survage	0.00	0.00	0.00	6
henri-rousseau	0.00	0.00	0.00	9
tano-festa	0.00	0.33	0.00	3
arman-manookian	0.00	0.00	0.00	9
theodor-severin-kittelsen	0.00	0.00	0.00	12
sebastien-bourdon	0.00	0.00	0.00	3
antoine-blanchard	0.00	0.00	0.00	24
pierre-puvis-de-chavannes	0.00	0.00	0.00	3
gotthard-graubner	0.00	0.00	0.00	3
joan-hernandez-pijuan	0.00	0.00	0.00	6
martin-schongauer	0.00	0.00	0.00	6
pavel-svinyin	0.00	0.00	0.00	2
marcus-larson	0.00	0.00	0.00	3
hieronymus-bosch	0.00	0.00	0.00	33
kimon-loghi	0.00	0.00	0.00	15

edmund-dulac	0.00	0.00	0.00	18
ronnie-landfield	0.00	0.00	0.00	15
giovanni-battista-tiepolo	0.00	0.00	0.00	15
francisc-sirato	0.00	0.00	0.00	6
sam-francis	0.00	0.00	0.00	51
francisco-bayeu-y-subias	0.00	0.00	0.00	3
carl-larsson	0.00	0.00	0.00	9
arthur-segal	0.00	0.00	0.00	12
jim-dine	0.00	0.00	0.00	3
fra-angelico	0.00	0.00	0.00	21
ilya-repin	0.00	0.00	0.00	90
keith-haring	0.00	0.12	0.00	8
heorhiy-narbut	0.00	0.00	0.00	27
henrique-pousao	0.00	0.00	0.00	3
ding-yanyong	0.00	0.00	0.00	3
tivadar-kosztka-csontvary	0.00	0.00	0.00	6
alexandre-benois	0.00	0.00	0.00	6
eva-gonzales	0.00	0.00	0.00	6
paul-delaroche	0.00	0.00	0.00	6
allan-d-arcangelo	0.00	0.00	0.00	3
charles-demuth	0.00	0.00	0.00	6
orest-kiprensky	0.00	0.00	0.00	36
rembrandt	0.00	0.00	0.00	128
william-shayer	0.00	0.00	0.00	15
ipolit-strambu	0.00	0.00	0.00	3
stefan-popescu	0.00	0.00	0.00	6
roger-de-la-fresnaye	0.00	0.00	0.00	3
william-baziotes	0.00	0.00	0.00	3
jan-sluyters	0.00	0.00	0.00	3
george-segal	0.00	0.00	0.00	3
richard-whitney	0.00	0.00	0.00	3
jean-degottex	0.00	0.00	0.00	6
hugo-van-der-goes	0.00	0.00	0.00	6
howard-hodgkin	0.00	0.00	0.00	9
william-blake	0.00	0.00	0.00	18
conrad-marca-relli	0.00	0.00	0.00	9
winston-churchill	0.00	0.00	0.00	15
pieter-de-hooch	0.00	0.00	0.00	22
jean-paul-riopelle	0.00	0.00	0.00	3
victor-pasmore	0.00	0.00	0.00	3
matej-sternen	0.00	0.00	0.00	3
max-slevogt	0.00	0.00	0.00	9
margareta-sterian	0.00	0.00	0.00	6
sven-lukin	0.00	0.00	0.00	3
hugo-simberg	0.00	0.00	0.00	3

marcel-duchamp	0.00	0.00	0.00	9
bertalan-por	0.00	0.00	0.00	3
richard-artschwager	0.00	0.00	0.00	3
kathe-kollwitz	0.00	0.00	0.00	6
damien-hirst	0.00	0.00	0.00	3
clyfford-still	0.00	0.00	0.00	3
michel-simonidy	0.00	0.00	0.00	3
amedee-ozenfant	0.00	0.00	0.00	6
patrick-procktor	0.00	0.00	0.00	9
ivan-vladimirov	0.00	0.00	0.00	9
louay-kayyali	0.00	0.00	0.00	3
gustave-caillebotte	0.00	0.00	0.00	21
viorel-marginean	0.00	0.00	0.00	9
georgios-jakobides	0.00	0.00	0.00	8
william-james-glackens	0.00	0.00	0.00	9
jean-hugo	0.00	0.00	0.00	3
constantin-guys	0.00	0.00	0.00	3
federico-zandomeneghi	0.00	0.00	0.00	6
stefan-luchian	0.00	0.00	0.00	18
maurice-quentin-de-la-tour	0.00	0.00	0.00	18
lucian-grigorescu	0.00	0.00	0.00	6
ito-jakuchu	0.00	0.00	0.00	3
utagawa-toyokuni	0.00	0.00	0.00	6
victor-hugo	0.00	0.00	0.00	3
oskar-kokoschka	0.00	0.00	0.00	6
camille-corot	0.00	0.00	0.00	78
vittore-carpaccio	0.00	0.00	0.00	18
camille-pissarro	0.00	0.00	0.00	143
anne-truitt	0.00	0.00	0.00	15
maria-helena-vieira-da-silva	0.00	0.00	0.00	3
frans-snyders	0.00	0.00	0.00	6
akseli-gallen-kallela	0.00	0.00	0.00	18
james-mcneill-whistler	0.00	0.00	0.00	24
hoca-ali-riza	0.00	0.00	0.00	2
john-miller	0.00	0.00	0.00	18
helen-frankenthaler	0.00	0.00	0.00	24
jean-francois-millet	0.00	0.00	0.00	30
princess-fahrelnissa-zeid	0.00	0.00	0.00	3
marko-pogacnik	0.00	0.00	0.00	3
gustave-dore	0.00	0.00	0.00	166
aleksey-antropov	0.00	0.00	0.00	3
adriaen-van-ostade	0.00	0.00	0.00	18
ford-madox-brown	0.00	0.00	0.00	9
endre-balint	0.00	0.00	0.00	6
neil-welliver	0.00	0.00	0.00	6

horace-pippin	0.00	0.00	0.00	3
carl-ludwig-johann-christineck	0.00	0.00	0.00	3
willi-baumeister	0.00	0.00	0.00	3
antonello-da-messina	0.00	0.00	0.00	3
harry-clarke	0.00	0.00	0.00	15
bruce-nauman	0.00	0.00	0.00	3
paul-feeley	0.00	0.00	0.00	3
grace-cossington-smith	0.00	0.00	0.00	9
ferdynand-ruszczyc	0.00	0.00	0.00	3
robert-mangold	0.00	0.00	0.00	9
ion-pacea	0.00	0.00	0.00	6
andre-dunoyer-de-segonzac	0.00	0.00	0.00	6
john-russell	0.00	0.00	0.00	6
edith-vonnegut	0.00	0.00	0.00	3
john-henry-twachtman	0.00	0.00	0.00	44
evelyne-axell	0.00	0.00	0.00	6
alexey-bogolyubov	0.00	0.00	0.00	15
vladimir-dimitrov	0.00	0.00	0.00	3
aaron-siskind	0.00	0.00	0.00	3
georges-braque	0.00	0.00	0.00	30
yiannis-tsaroychis	0.00	0.00	0.00	18
ilka-gedo	0.00	0.00	0.00	15
dimitrie-paciurea	0.00	0.00	0.00	6
charles-hermans	0.00	0.00	0.00	6
basil-beattie	0.00	0.00	0.00	9
lavinia-fontana	0.00	0.00	0.00	6
nicholas-roerich	0.00	0.00	0.00	317
winslow-homer	0.00	0.00	0.00	12
jean-fouquet	0.00	0.00	0.00	20
fernando-botero	0.00	0.00	0.00	12
johan-hendrik-weissenbruch	0.00	0.00	0.00	11
fernand-leger	0.00	0.00	0.00	45
ligia-macovei	0.00	0.00	0.00	21
adnan-coker	0.00	0.00	0.00	2
mily-possoz	0.00	0.00	0.00	3
balthus	0.00	0.00	0.00	12
andre-masson	0.00	0.00	0.00	6
alexander-orlowski	0.00	0.00	0.00	3
pieter-wenning	0.00	0.00	0.00	9
roberto-matta	0.00	0.00	0.00	3
jacek-malczewski	0.00	0.00	0.00	27
viktor-vasnetsov	0.00	0.00	0.00	21
tia-peltz	0.00	0.00	0.00	6
theo-van-rysselberghe	0.00	0.00	0.00	32
jean-leon-gerome	0.00	0.00	0.00	3

polychronis-lembesis	0.00	0.00	0.00	12
roni-horn	0.00	0.00	0.00	2
arkhip-kuindzhi	0.00	0.00	0.00	32
paul-cezanne	0.00	0.00	0.00	105
artemisia-gentileschi	0.00	0.00	0.00	3
arnold-ba-cklin	0.00	0.00	0.00	17
henri-catargi	0.00	0.00	0.00	4
nicolae-grigorescu	0.00	0.00	0.00	3
jacoba-van-heemskerck	0.00	0.00	0.00	3
otto-eckmann	0.00	0.00	0.00	6
constantin-daniel-rosenthal	0.00	0.00	0.00	9
constantin-artachino	0.00	0.00	0.00	6
edouard-vuillard	0.00	0.00	0.00	3
esaias-van-de-velde	0.00	0.00	0.00	18
atsuko-tanaka	0.00	0.00	0.00	2
gian-lorenzo-bernini	0.00	0.00	0.00	12
claude-monet	0.00	0.00	0.00	244
philip-guston	0.00	0.00	0.00	3
andre-pierre-arnal	0.00	0.00	0.00	3
mikhail-nesterov	0.00	0.00	0.00	17
kuzma-petrov-vodkin	0.00	0.00	0.00	33
louise-elisabeth-vigee-le-brun	0.00	0.00	0.00	12
tom-thomson	0.00	0.00	0.00	3
carlos-orozco-romero	0.00	0.00	0.00	3
jan-van-eyck	0.00	0.00	0.00	6
robert-brackman	0.00	0.00	0.00	3
mario-nuzzi	0.00	0.00	0.00	3
basuki-abdullah	0.00	0.00	0.00	3
johann-koler	0.00	0.00	0.00	3
giorgione	0.00	0.00	0.00	3
osias-beert	0.00	0.00	0.00	3
mary-cassatt	0.00	0.00	0.00	63
seraphine-louis	0.00	0.00	0.00	3
albert-pinkham-ryder	0.00	0.00	0.00	3
utagawa-sadatora	0.00	0.00	0.00	6
edouard-manet	0.00	0.00	0.00	26
janos-tornyai	0.00	0.00	0.00	3
domenico-ghirlandaio	0.00	0.00	0.00	20
mark-rothko	0.00	0.00	0.00	26
leon-bonnat	0.00	0.00	0.00	3
georges-seurat	0.00	0.00	0.00	24
gene-davis	0.00	0.00	0.00	27
konstantin-bogaevsky	0.00	0.00	0.00	6
agnes-martin	0.00	0.00	0.00	3
george-morland	0.00	0.00	0.00	20

andrea-del-sarto	0.00	0.00	0.00	9
ramon-oviedo	0.00	0.00	0.00	6
cornelis-vreedenburgh	0.00	0.00	0.00	3
albert-gleizes	0.00	0.00	0.00	12
andre-derain	0.00	0.00	0.00	6
mikalojus-ciurlionis	0.00	0.00	0.00	30
arthur-pinajian	0.00	0.00	0.00	6
william-turner	0.00	0.09	0.01	33
periklis-vyzantios	0.00	0.00	0.00	6
dosso-dossi	0.00	0.00	0.00	6
yiannis-moralis	0.00	0.00	0.00	6
william-adolphe-bouguereau	0.00	0.00	0.00	12
bernardo-bellotto	0.00	0.00	0.00	6
adolf-hitler	0.00	0.00	0.00	3
leroy-neiman	0.00	0.00	0.00	3
gabriel-metsu	0.00	0.00	0.00	6
maurice-utrillo	0.00	0.00	0.00	15
martin-barre	0.00	0.00	0.00	6
george-stubbs	0.00	0.00	0.00	18
henri-le-fauconnier	0.00	0.00	0.00	3
ilya-mashkov	0.00	0.00	0.00	30
diego-velazquez	0.00	0.00	0.00	12
jules-cheret	0.00	0.00	0.00	3
nicholas-krushenick	0.00	0.00	0.00	6
moise-kisling	0.00	0.00	0.00	20
allan-ramsay	0.00	0.00	0.00	21
francesco-solimena	0.00	0.00	0.00	9
joe-goode	0.00	0.00	0.00	6
pierre-alechinsky	0.00	0.00	0.00	12
petrus-christus	0.00	0.00	0.00	6
louis-janmot	0.00	0.00	0.00	9
richard-parkes-bonington	0.00	0.00	0.00	15
sever-burada	0.00	0.00	0.00	8
ivan-rutkovych	0.00	0.00	0.00	3
mikhail-lebedev	0.00	0.00	0.00	3
walter-sickert	0.00	0.00	0.00	16
suzanne-valadon	0.00	0.00	0.00	6
forrest-bess	0.00	0.00	0.00	9
richard-serra	0.00	0.00	0.00	3
nikolaos-gyzis	0.00	0.00	0.00	3
bartolome-bermejo	0.00	0.00	0.00	3
gerard-david	0.00	0.00	0.00	9
nikolai-ge	0.00	0.00	0.00	6
accuracy			0.00	13343

```
   macro avg       0.00      0.00      0.00     13343
weighted avg       0.00      0.00      0.00     13343
```

In [184…

```
#Counter for Most commonly predicted artists
artist_counts = Counter(predicted_artists)
print("Most Common Predicted Artists:")
for artist_idx, count in artist_counts.most_common(5):
    print(f"{artists[artist_idx]}: {count} predictions")
```

```
Most Common Predicted Artists:
keith-haring: 2386 predictions
anta-nio-de-carvalho-da-silva-porto: 1769 predictions
william-turner: 1107 predictions
francesco-hayez: 829 predictions
tano-festa: 729 predictions
```

In [133…

```
#correct v. predictions artist labels
print("Sample of True Labels:", artist_labels[:10])
print("Sample of Predictions:", predicted_artists[:10])
```

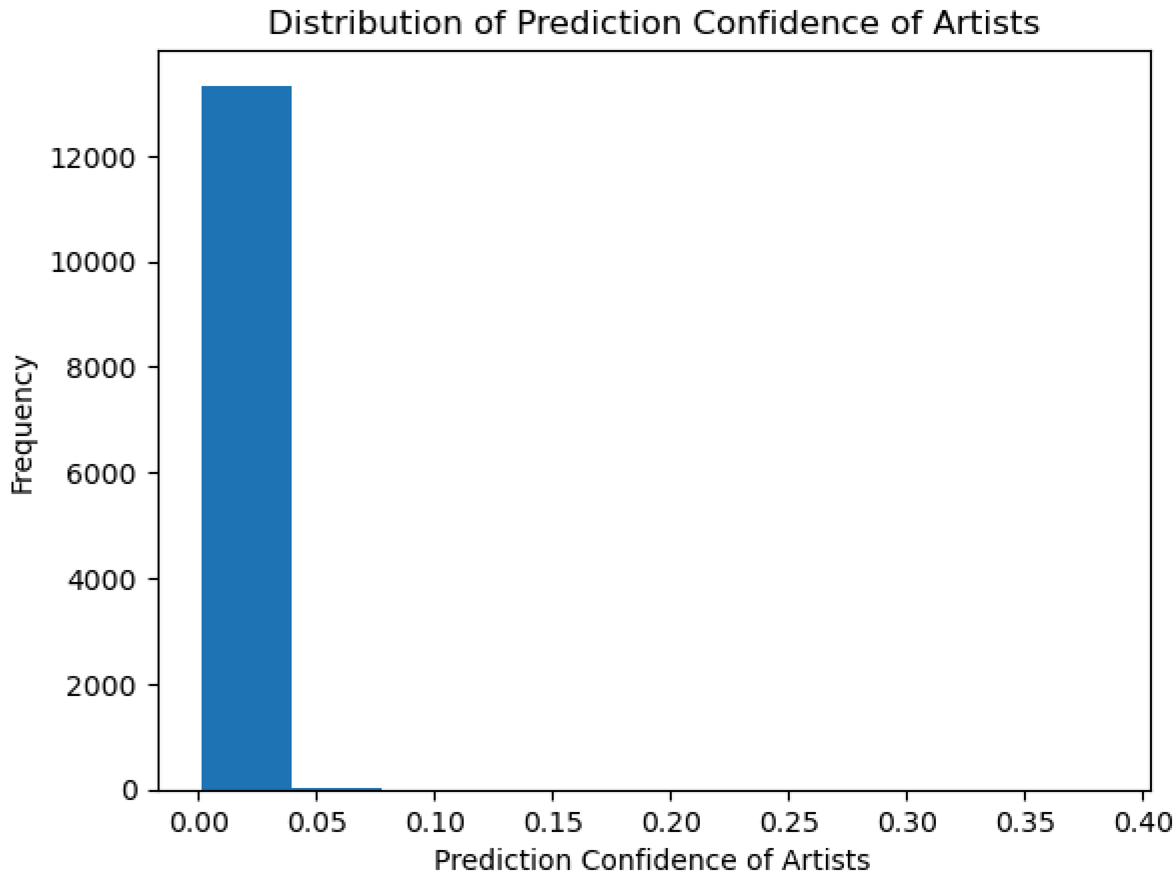
```
Sample of True Labels: ['sandro-botticelli', 'piero-della-francesca', 'carlo-crivelli', 'pietro-perugino', 'c
arlo-crivelli', 'benozzo-gozzoli', 'luca-signorelli', 'fra-angelico', 'luca-signorelli', 'carlo-crivelli']
Sample of Predictions: [251 357 815 639 633 639 492 188  47 749]
```

In [144…

```
max_probs_artist = np.max(artist_predictions, axis=1)
```

In [145…

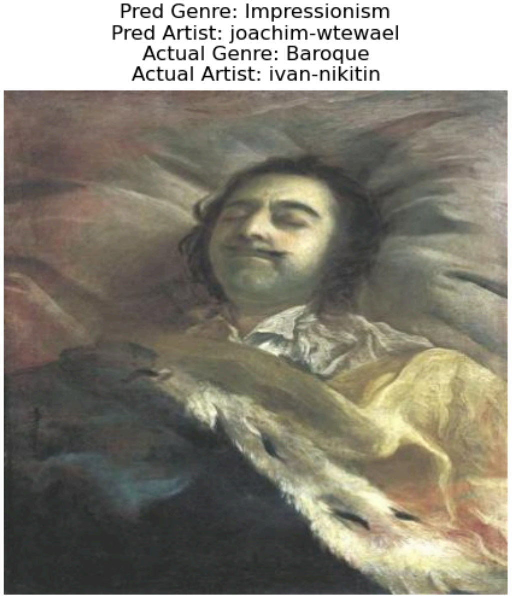
```
#plot of max probs artists confidence
plt.hist(max_probs_artist, bins = 10)
plt.xlabel("Prediction Confidence of Artists")
plt.ylabel("Frequency")
plt.title("Distribution of Prediction Confidence of Artists")
plt.show()
```

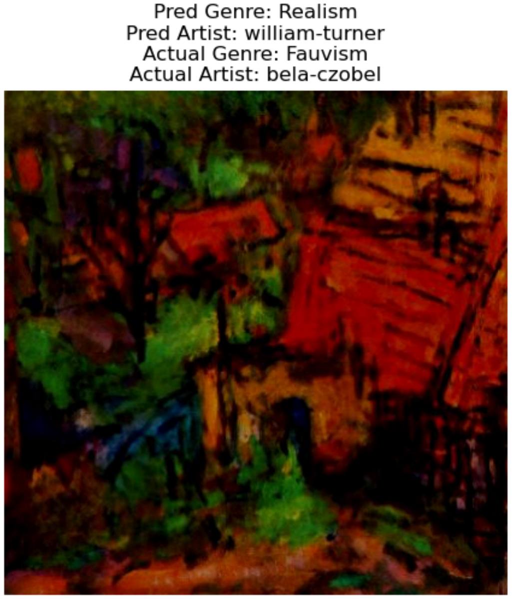


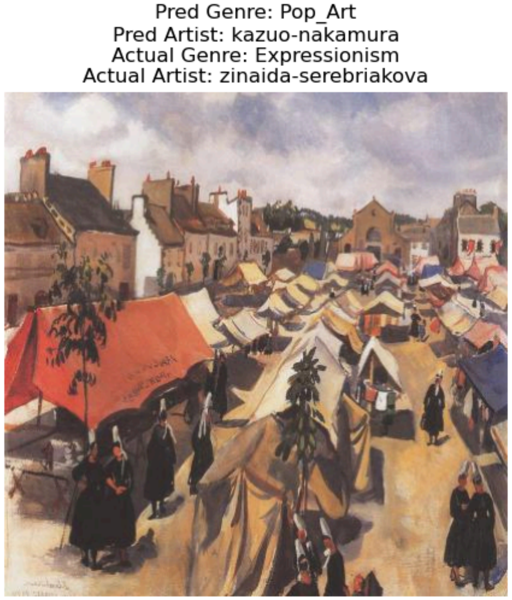
In [86]:

```
#Art work Plot with Real v. Predictions
def visualize_predictions(num_samples = 3):
    sample_indices = np.random.choice(len(image_files), num_samples, replace=False)
    fig, axes = plt.subplots(1, num_samples, figsize = (15, 5))
    for i, idx in enumerate(sample_indices):
        img_path = image_files[idx]
        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  #conver to rgb
        actual_genre = genre_classes[genre_labels[idx]] #extracting file path
        actual_artist = artist_labels[idx]
        axes[i].imshow(img) #displaying image
        axes[i].axis("off")
        axes[i].set_title(f"Pred Genre: {predicted_genre_labels[idx]}\nPred Artist: {predicted_artist_labels[
                          f"Actual Genre: {actual_genre}\nActual Artist: {actual_artist}")
    plt.tight_layout()
```

```
    plt.show()
visualize_predictions()
```







In [ ]: