Eduardo García Castrejón

Final Project Report

Problem: Image Classification of Paintings by the Greatest Artists in History

Statement:

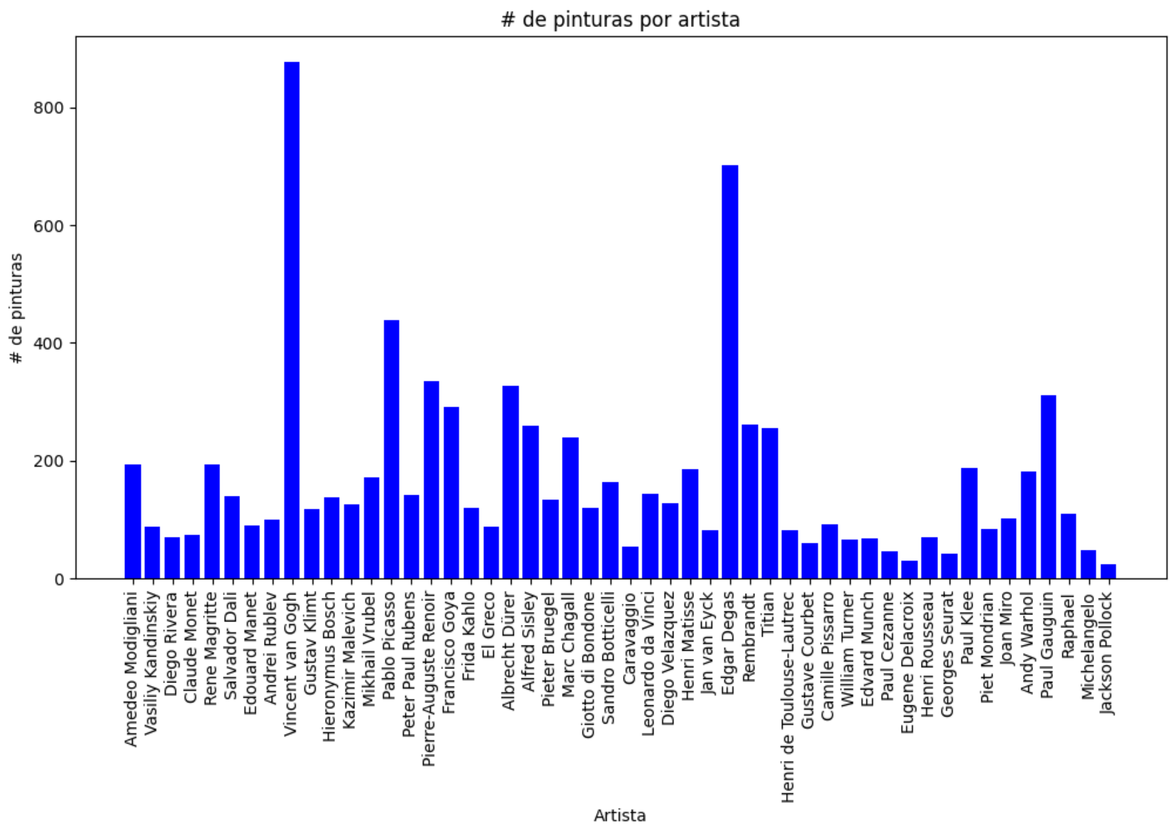
We have a dataset of a collection of paintings by 50 artists throughout history, obtained from Kaggle: https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time

The content of this dataset is as follows:

- **artists.csv:** Information about each artist, including the number of paintings in the dataset.
- **images.zip:** A collection of full-size images, divided into folders for each artist.

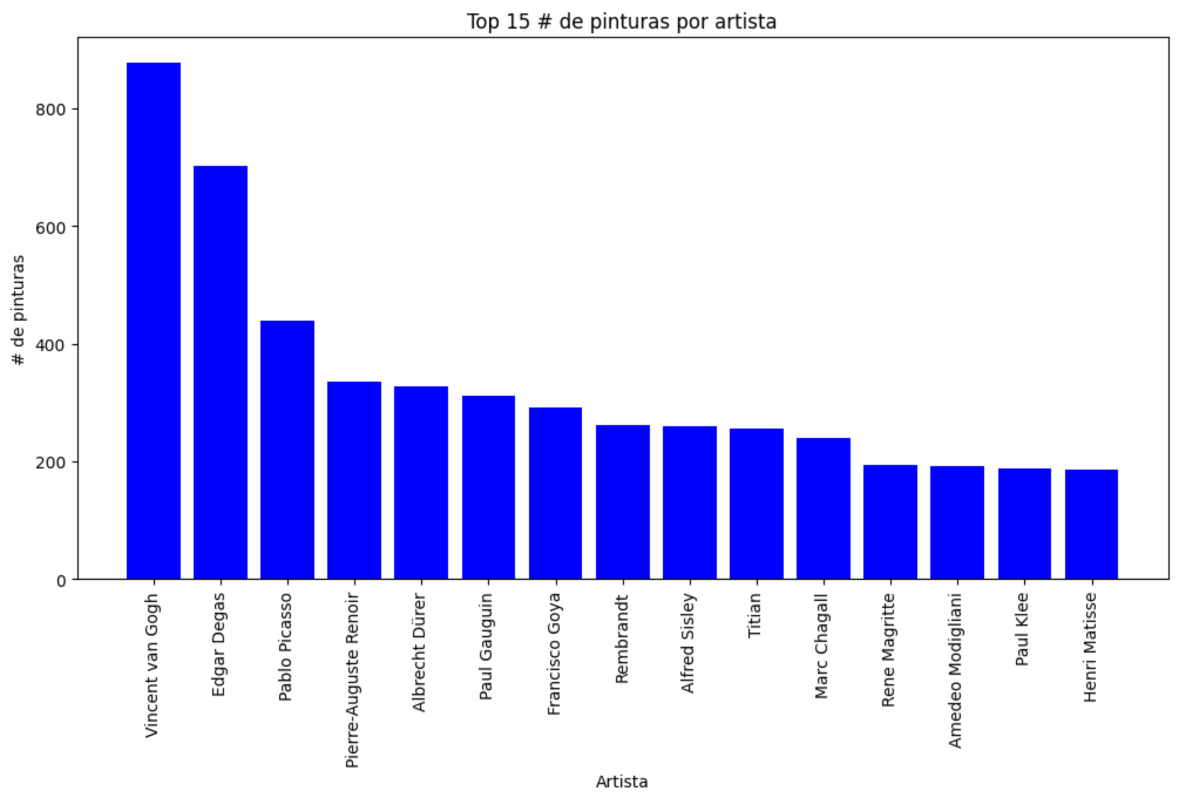
The plan is to design a Deep Learning model in Python capable of recognizing the artist by analyzing essential features in their paintings, such as geometric patterns or color palette.

The following figure shows a histogram indicating the frequency of each class in the dataset.



The main challenges of this problem are the presence of imbalance and a large number of classes. Additionally, the paintings of each artist may not be sufficiently identifiable due to a lack of uniqueness in the artist's style or an inconsistent style across all of an artist's works.

Therefore, the scope of this project will be limited by reducing the number of classes to 15 artists, specifically the 15 classes with the most data. The following histogram shows the 15 selected painters:



Objectives:

- **General:** Create a CNN model capable of identifying the paintings of 15 artists.
- **Specific:** Evaluate the model's performance.

Methodology:

The methodology is based on common Deep Learning techniques, focusing on convolutional networks, using TensorFlow and Keras tools in Python programming.

The use of Convolutional Neural Networks is justified by their ability to extract essential features from an image to perform categorical classification efficiently.

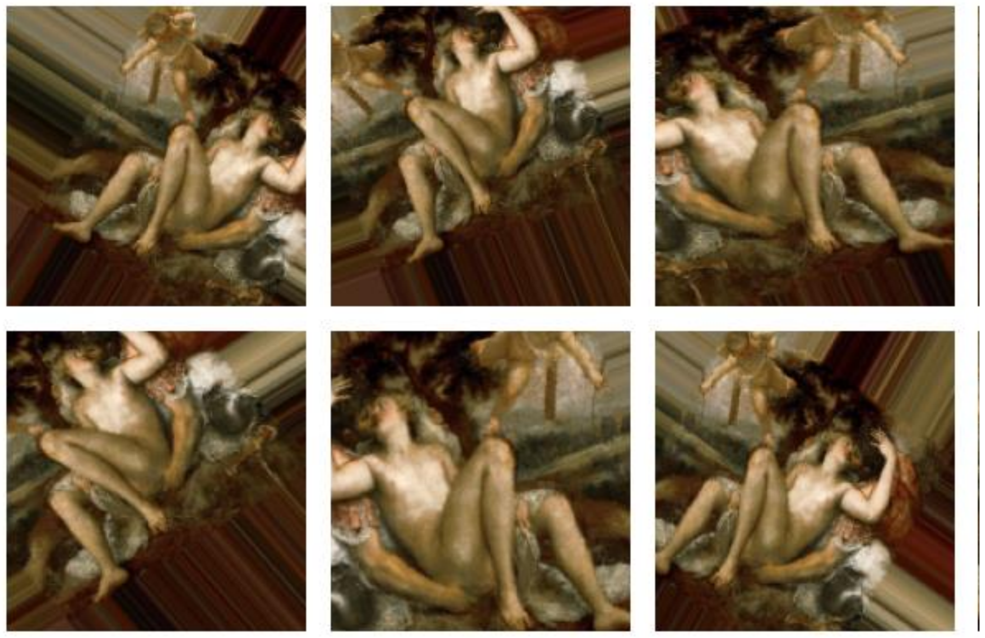
- **Convolutional and Pooling Layers**
- **Fully Connected Layers**
- **Activation Function**: ReLU
- **Output Layer**: Softmax
- **Loss Function**: Categorical Cross-Entropy, suitable for multi-category classification problems.
- **Optimizer**: RMSprop

The proposed process is carried out in the following steps:

Folder Organization: Organize the image folders to have dedicated folders for training, validation, and testing. A Python function was written to randomly split a folder into two folders based on a given ratio. The distribution was planned as follows:

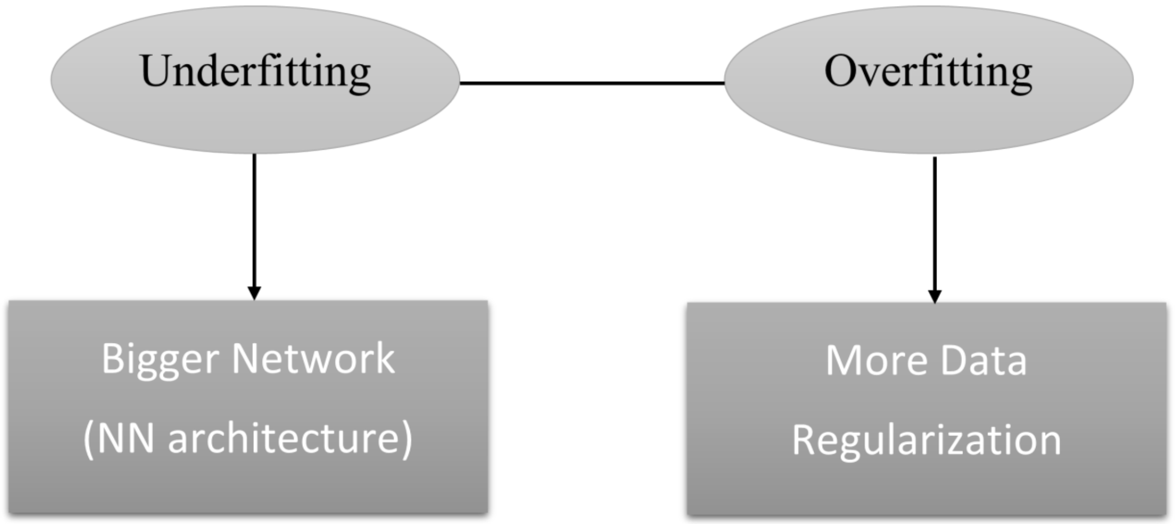
- **test: 15%**
- **train/val: 85%**
  - train: 75%
  - validation: 25%

Preprocessing: Using keras.ImageDataGenerator, the JPEG content was decoded into RGB pixel grids and converted into normalized floating-point vectors of the images with a dimension of (256, 256, 3). For the training generator, data augmentation was specified, such as rotations, translations, scaling, and mirroring (an example in the following figure) to increase the model's variability and robustness.



Design of the CNN Model Architecture: The model architecture is designed using Keras.models. It is crucial to find an architecture such that the model does not present cases where the accuracy and loss curves on the training data differ significantly from those on the validation data. If the validation loss curve does not decrease and the accuracy curve does not increase along with the training curves, it is a case of overfitting; the model has fit too closely to the training data. The opposite case is underfitting, where the model is not learning enough.

The method to achieve an optimal architecture is explained in the following figure:



When training a proposed model, if overfitting is observed, regularization must be implemented in the model, or more data must be added. In this problem, to increase the data, the parameters of data augmentation in preprocessing can be increased, but this is not recommended as it may alter important features of the artist. Therefore, to combat overfitting, the following regularization tools were implemented, experimenting with different parameters:

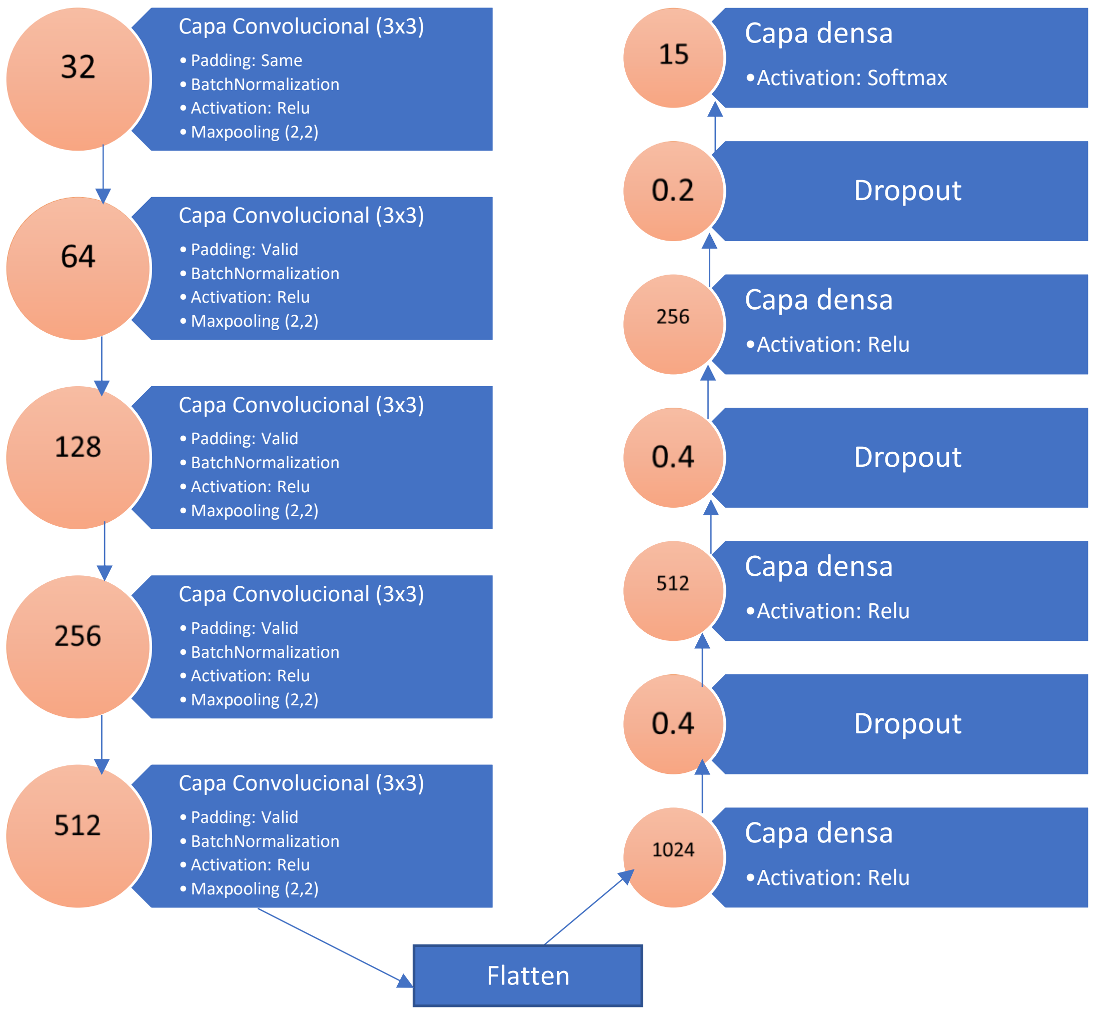
- **Dropout:** (Set the weights of a number of neurons in each fully connected dense layer to zero)
- **Learning Rate Decay:** (Implement a decay in the learning rate as epochs increase)
- **Batch Normalization:** (Normalize the activation output after each convolutional layer)
- **L2 Regularization:** (Add a penalty term to the loss function)

In the presence of underfitting, the complexity of the model is increased by adding more convolutional layers, as well as increasing the number of dense layers and neurons.

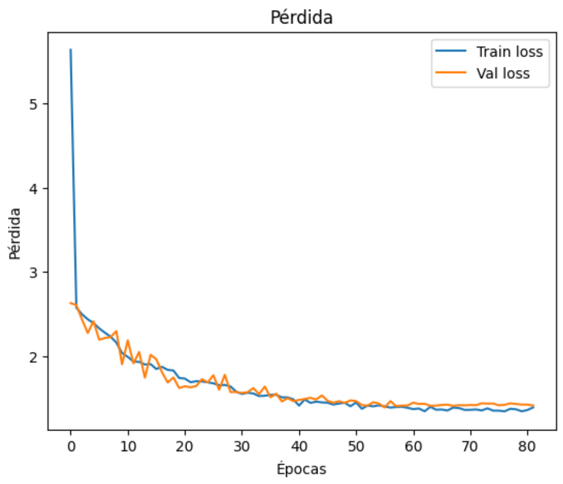
- **Evaluation:** Evaluate the model with accuracy, top-2 accuracy, training curves, and confusion matrices.

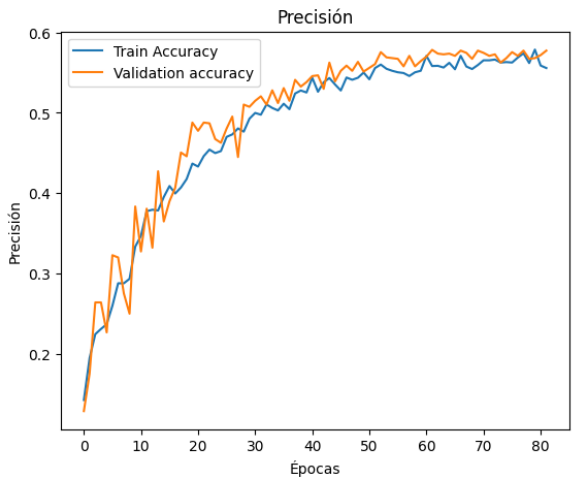
Results:

The next model has the architecture as follows:



Below are the learning curves. As can be observed, the model fits appropriately on both the training and validation data.

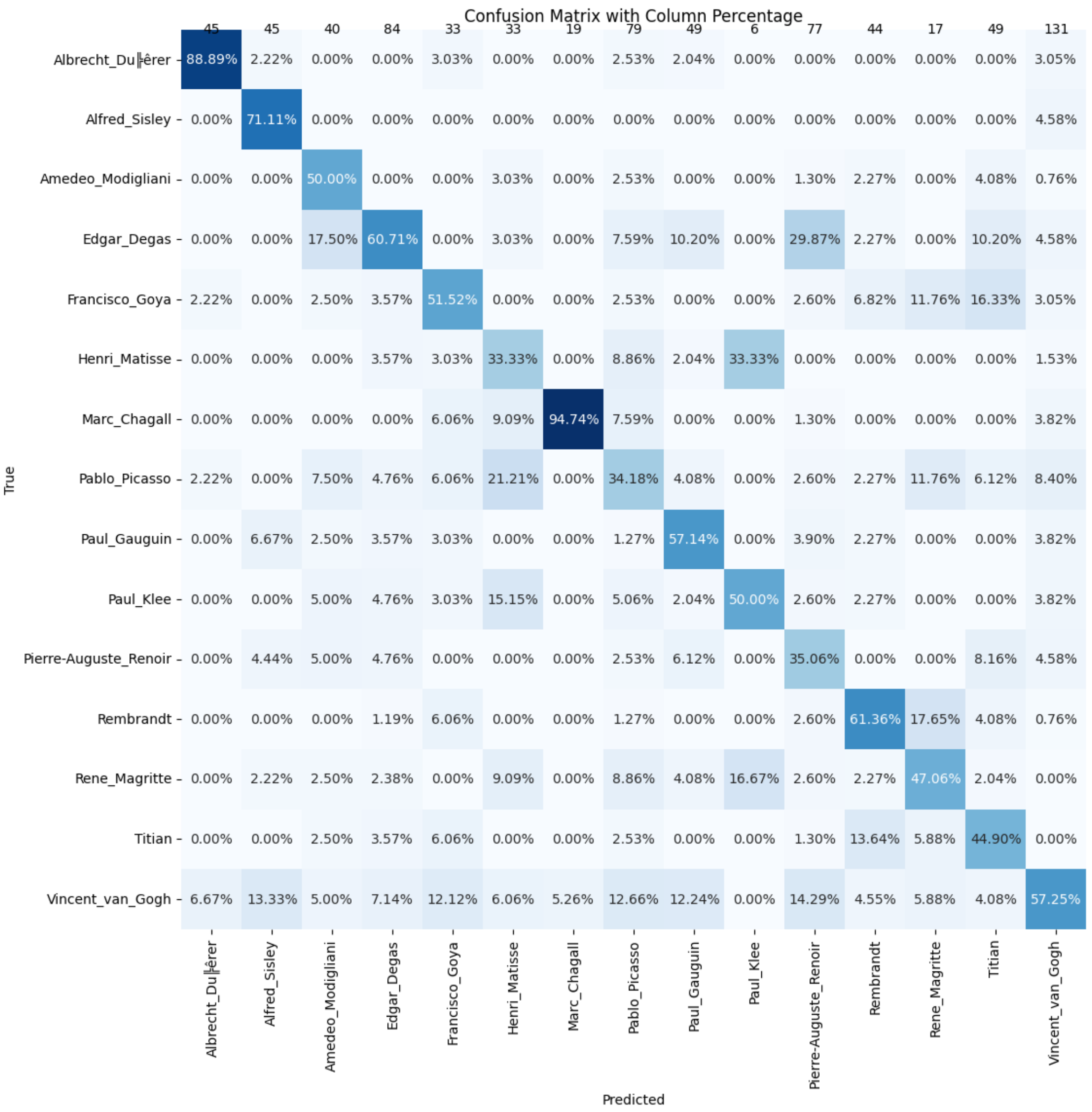




The following results were obtained:

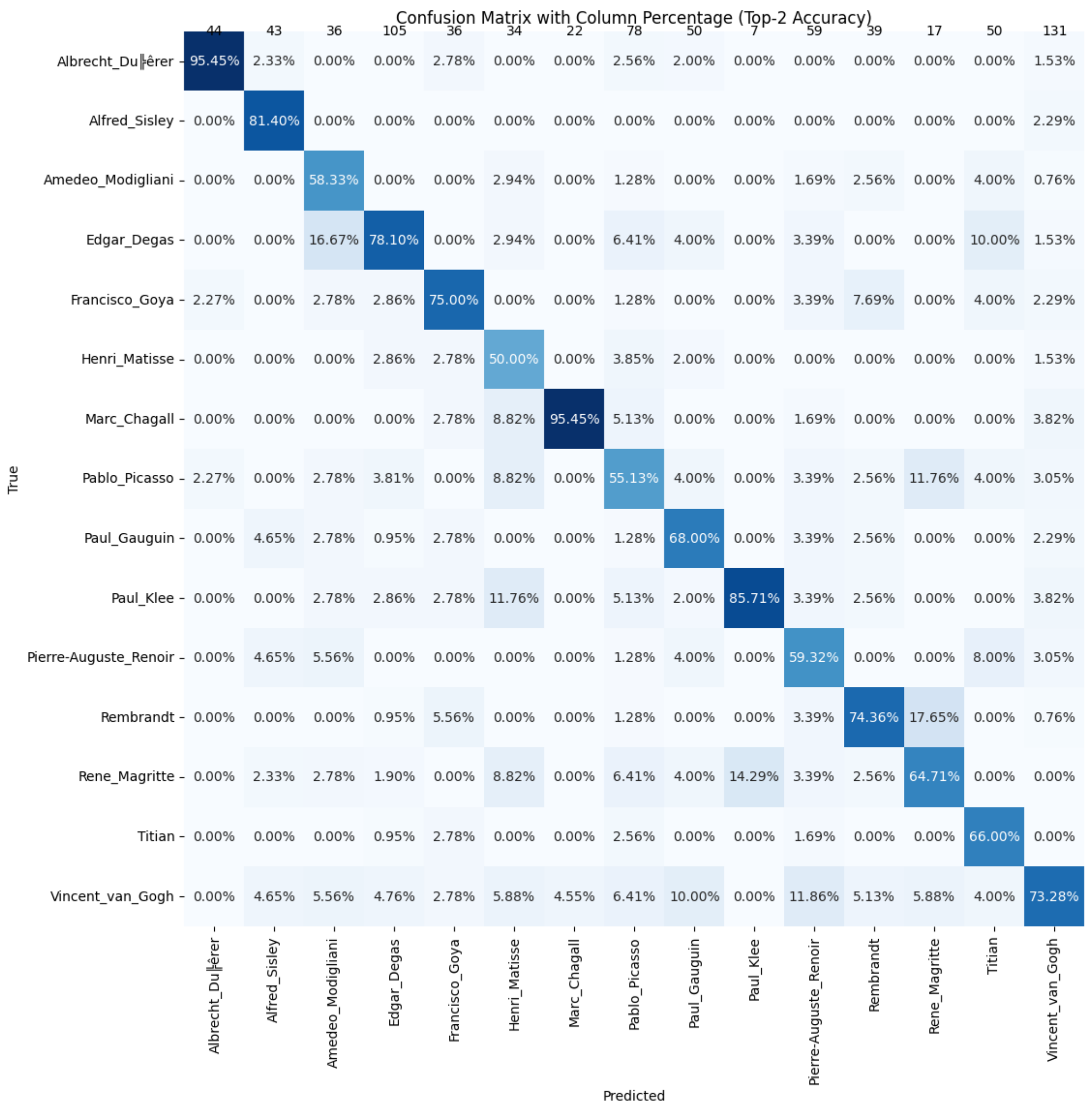
- Test accuracy: 0.55
- Test top-3 accuracy: 0.79
- Test top-2 accuracy: 0.71

However, upon examining the confusion matrix, it can be noted that there is better performance for certain artists compared to others, due to their paintings being more consistent in a unique style.


Confusion Matrix with Column Percentage
True
Predicted

	Albrecht_Du╟êrer	Alfred_Sisley	Amedeo_Modigliani	Edgar_Degas	Francisco_Goya	Henri_Matisse	Marc_Chagall	Pablo_Picasso	Paul_Gauguin	Paul_Klee	Pierre-Auguste_Renoir	Rembrandt	Rene_Magritte	Titian	Vincent_van_Gogh
	45	45	40	84	33	33	19	79	49	6	77	44	17	49	131
Albrecht_Du╟êrer	88.89%	2.22%	0.00%	0.00%	3.03%	0.00%	0.00%	2.53%	2.04%	0.00%	0.00%	0.00%	0.00%	0.00%	3.05%
Alfred_Sisley	0.00%	71.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	4.58%
Amedeo_Modigliani	0.00%	0.00%	50.00%	0.00%	0.00%	3.03%	0.00%	2.53%	0.00%	0.00%	1.30%	2.27%	0.00%	4.08%	0.76%
Edgar_Degas	0.00%	0.00%	17.50%	60.71%	0.00%	3.03%	0.00%	7.59%	10.20%	0.00%	29.87%	2.27%	0.00%	10.20%	4.58%
Francisco_Goya	2.22%	0.00%	2.50%	3.57%	51.52%	0.00%	0.00%	2.53%	0.00%	0.00%	2.60%	6.82%	11.76%	16.33%	3.05%
Henri_Matisse	0.00%	0.00%	0.00%	3.57%	3.03%	33.33%	0.00%	8.86%	2.04%	33.33%	0.00%	0.00%	0.00%	0.00%	1.53%
Marc_Chagall	0.00%	0.00%	0.00%	0.00%	6.06%	9.09%	94.74%	7.59%	0.00%	0.00%	1.30%	0.00%	0.00%	0.00%	3.82%
Pablo_Picasso	2.22%	0.00%	7.50%	4.76%	6.06%	21.21%	0.00%	34.18%	4.08%	0.00%	2.60%	2.27%	11.76%	6.12%	8.40%
Paul_Gauguin	0.00%	6.67%	2.50%	3.57%	3.03%	0.00%	0.00%	1.27%	57.14%	0.00%	3.90%	2.27%	0.00%	0.00%	3.82%
Paul_Klee	0.00%	0.00%	5.00%	4.76%	3.03%	15.15%	0.00%	5.06%	2.04%	50.00%	2.60%	2.27%	0.00%	0.00%	3.82%
Pierre-Auguste_Renoir	0.00%	4.44%	5.00%	4.76%	0.00%	0.00%	0.00%	2.53%	6.12%	0.00%	35.06%	0.00%	0.00%	8.16%	4.58%
Rembrandt	0.00%	0.00%	0.00%	1.19%	6.06%	0.00%	0.00%	1.27%	0.00%	0.00%	2.60%	61.36%	17.65%	4.08%	0.76%
Rene_Magritte	0.00%	2.22%	2.50%	2.38%	0.00%	9.09%	0.00%	8.86%	4.08%	16.67%	2.60%	2.27%	47.06%	2.04%	0.00%
Titian	0.00%	0.00%	2.50%	3.57%	6.06%	0.00%	0.00%	2.53%	0.00%	0.00%	1.30%	13.64%	5.88%	44.90%	0.00%
Vincent_van_Gogh	6.67%	13.33%	5.00%	7.14%	12.12%	6.06%	5.26%	12.66%	12.24%	0.00%	14.29%	4.55%	5.88%	4.08%	57.25%

In the confusion matrix, when considering the two most probable predictions, we observe a significantly higher performance.

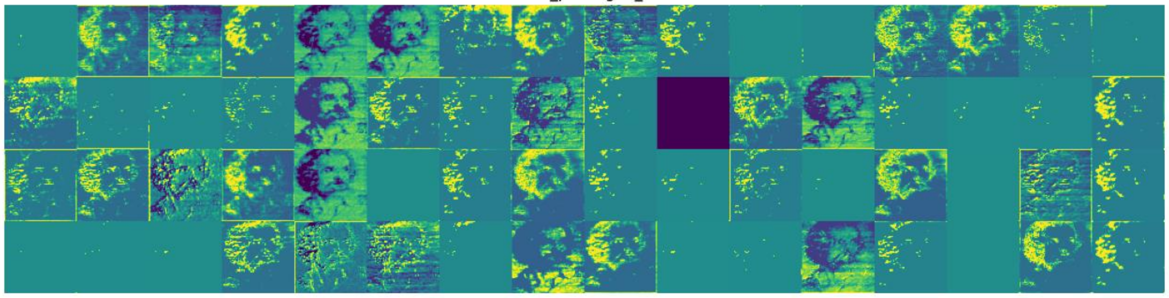


Confusion Matrix with Column Percentage (Top-2 Accuracy)

True \ Predicted	Albrecht_Du╟êrer	Alfred_Sisley	Amedeo_Modigliani	Edgar_Degas	Francisco_Goya	Henri_Matisse	Marc_Chagall	Pablo_Picasso	Paul_Gauguin	Paul_Klee	Pierre-Auguste_Renoir	Rembrandt	Rene_Magritte	Titian	Vincent_van_Gogh
	44	43	36	105	36	34	22	78	50	7	59	39	17	50	131
Albrecht_Du╟êrer	95.45%	2.33%	0.00%	0.00%	2.78%	0.00%	0.00%	2.56%	2.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.53%
Alfred_Sisley	0.00%	81.40%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.29%
Amedeo_Modigliani	0.00%	0.00%	58.33%	0.00%	0.00%	2.94%	0.00%	1.28%	0.00%	0.00%	1.69%	2.56%	0.00%	4.00%	0.76%
Edgar_Degas	0.00%	0.00%	16.67%	78.10%	0.00%	2.94%	0.00%	6.41%	4.00%	0.00%	3.39%	0.00%	0.00%	10.00%	1.53%
Francisco_Goya	2.27%	0.00%	2.78%	2.86%	75.00%	0.00%	0.00%	1.28%	0.00%	0.00%	3.39%	7.69%	0.00%	4.00%	2.29%
Henri_Matisse	0.00%	0.00%	0.00%	2.86%	2.78%	50.00%	0.00%	3.85%	2.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.53%
Marc_Chagall	0.00%	0.00%	0.00%	0.00%	2.78%	8.82%	95.45%	5.13%	0.00%	0.00%	1.69%	0.00%	0.00%	0.00%	3.82%
Pablo_Picasso	2.27%	0.00%	2.78%	3.81%	0.00%	8.82%	0.00%	55.13%	4.00%	0.00%	3.39%	2.56%	11.76%	4.00%	3.05%
Paul_Gauguin	0.00%	4.65%	2.78%	0.95%	2.78%	0.00%	0.00%	1.28%	68.00%	0.00%	3.39%	2.56%	0.00%	0.00%	2.29%
Paul_Klee	0.00%	0.00%	2.78%	2.86%	2.78%	11.76%	0.00%	5.13%	2.00%	85.71%	3.39%	2.56%	0.00%	0.00%	3.82%
Pierre-Auguste_Renoir	0.00%	4.65%	5.56%	0.00%	0.00%	0.00%	0.00%	1.28%	4.00%	0.00%	59.32%	0.00%	0.00%	8.00%	3.05%
Rembrandt	0.00%	0.00%	0.00%	0.95%	5.56%	0.00%	0.00%	1.28%	0.00%	0.00%	3.39%	74.36%	17.65%	0.00%	0.76%
Rene_Magritte	0.00%	2.33%	2.78%	1.90%	0.00%	8.82%	0.00%	6.41%	4.00%	14.29%	3.39%	2.56%	64.71%	0.00%	0.00%
Titian	0.00%	0.00%	0.00%	0.95%	2.78%	0.00%	0.00%	2.56%	0.00%	0.00%	1.69%	0.00%	0.00%	66.00%	0.00%
Vincent_van_Gogh	0.00%	4.65%	5.56%	4.76%	2.78%	5.88%	4.55%	6.41%	10.00%	0.00%	11.86%	5.13%	5.88%	4.00%	73.28%

In the confusion matrix, when considering the two most probable predictions, we observe a significantly higher performance. As a demonstration of the convolutional layer process, we have an image belonging to a painting by Albrecht Dürer and the visual representation of the result from the sixth convolutional layer:

max_pooling2d_6



Conclusion:

- The dataset is highly susceptible to overfitting due to its small size and class imbalance.
- The dataset is challenging to predict due to the limited exposure of data to the CNN model. It can also be argued that the inherent features of each painting are not sufficiently unique and are similar between artists.
- Top-2 test accuracy for each artist ranges between 50% and 95%. This difference is due to the unique features and consistency among each artist's paintings.
- More sophisticated tools will be needed to improve the model, which turned out to be very complex.