

**MSc in Business Analytics**

**Machine Learning and Content Analytics**

**Project**

**“Recommendation system development for art paintings through image classification”**

|  |  |
| --- | --- |
| Members | ID |
| Evangelos Lakkas-Pyknis | f2822306 |
| Stamatoula-Gerasimoula Mesolora | f2822308 |
| Anna Papadimitriou | f2822311 |
| Eleni Ralli | f2822312 |

Contents

[1. Project description 3](#_Toc178518848)

[Business Goal and Impact 3](#_Toc178518849)

[2. Data collection 4](#_Toc178518850)

[3. Dataset Overview 5](#_Toc178518851)

[4. Data Preprocessing 10](#_Toc178518852)

[5. Methodology 12](#_Toc178518853)

[5.1 Introduction 12](#_Toc178518854)

[5.2 Classification Models’ Description 12](#_Toc178518855)

[*5.2.1 EfficientNetB3* 12](#_Toc178518856)

[*5.2.2* *VGG16 and VGG19* 13](#_Toc178518857)

[*5.2.3* *Vision Transformer (ViT)* 13](#_Toc178518858)

[5.3 Models’ Evaluation and Comparison 15](#_Toc178518859)

[5.4 Fine-tuning ViT model 19](#_Toc178518860)

[5.5 Recommendation System 19](#_Toc178518861)

[*5.5.1 Key aspects and development of the recommendation system* 19](#_Toc178518862)

[*5.5.2* *Recommendations Examples* 20](#_Toc178518863)

[6. Members/Roles and Time Plan 23](#_Toc178518864)

[References 25](#_Toc178518865)

# Project description

Our project aims to build a model where it reads the images and the characteristics of the paintings, learn from them with multiclass classification techniques, and then develop an advanced painting recommendation system to recommend similar paintings to users. The innovation of this project on the topic of art image classification is that, despite that there are several papers and projects which explored solutions for the art image classification task (C. Sandoval et al., 2018, W. R. Tan et al., 2016, M. Oguz Kelek et al., 2019, K. Alfianto Jangtjik et al., 2016 & O. Menis – Mastromichalakis et al., 2024) the approaches that are published are based on single or double output labels (mostly based on style and artist metadata) we could not find research that responds to multiple output classification that refers to classification with 5 outputs (and with at least 20 classes per label as we had in our dataset). So, this is the challenge that we had to overcome, and after its -business value- enrichment with the development of a recommendation system.

Our system will analyze and then predict the following characteristics of artworks:

|  |  |
| --- | --- |
| **Image ID** | The image ID corresponding to the *resized.zip* file |
| **Title** | The title of the painting |
| **Artist** | The artist who created the painting |
| **Subject** | The theme or subject matter of the painting |
| **Style** | The artistic style of the painting |
| **Materials** | The materials used in the painting |
| **Period** | The time period during which the artworks were created |

## Business Goal and Impact

This recommendation system can help online art platforms, galleries, and museums increase user engagement and purchases by providing personalized art ideas. By learning about the similarities and relations between different paintings (and potentially exploiting this system in combination with historical user preferences), the platform could make tailored recommendations that not only boost user pleasure but also stimulate more interaction and potentially drive purchases. This technique will provide a more integrated experience for art enthusiasts by facilitating their search and get recommendations in a simple, quick and efficient manner.

# Data collection

For the purpose of this project, we needed to collect data for paintings. Specifically, we had to find various art paintings and gather relevant information about them in order to construct our dataset. To do this, we searched the Kaggle website for datasets containing many art paintings. After a thorough search, we found a dataset that matched our needs. This dataset, [“Best Artworks of All Time”](https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time/data), contains a collection of paintings from the 50 most influential artists of all time.

The selected dataset includes three files: *artists.csv*, *images.zip*, and *resized.zip.* The *artists.csv* file refers various information about the artists, such as their names, ages, nationalities, and genres, to name a few. The *images.zip* file contains all 8355 images (full size), divided into folders and numbered sequentially. Finally, the resized.zip file contains the same collection of images, but they have been resized and extracted from the folder structure.

For our project, we only selected the *resized.zip* file from the dataset. We first downloaded the images from this file and decided that we would select 400 images from them. Since we were a four-member group, we each selected 10 artists and chose 10 works from each, resulting in 100 rows per member. After we had selected the artists and images, we searched for information about each painting online. The information needed to be reliable, so we chose two authoritative websites on paintings: “[WikiArt.org - Visual Art Encyclopedia](https://www.wikiart.org/)”and“[Artchive.com](file:///C:\Users\Μπαμπης\Documents\MSc%20Business%20Analytics\ML%20and%20content%20analytics\artchive.com)”. Finally, after collecting the appropriate information for the paintings, we created a dataset with 400 rows and 8 columns, as shown in the table below:

|  |  |
| --- | --- |
| Variable name | Description |
| Image ID | The image ID corresponding to the *resized.zip* file |
| Title | The title of the painting |
| Artist | The artist who created the painting |
| Subject | The theme or subject matter of the painting |
| Style | The artistic style of the painting |
| Materials | The materials used in the painting |
| Start | The time period when the painting was started |
| End | The time period when the painting was completed |

# Dataset Overview

The resulting dataset was then loaded into Google Colab to proceed with the analysis. We developed code that extracted the 400 selected image files from the *resized.zip* folder based on the list of image IDs from Excel. At that point, we had updated the dataset by adding the paths to the corresponding images. The updated dataset consisted of 7 columns: *Image\_Path, Artist, Subject, Style, Materials, Start* and *End*.

The column ‘Artist’ is a categorical variable with 40 unique values. The histogram below shows the frequency of each artist in the dataset:

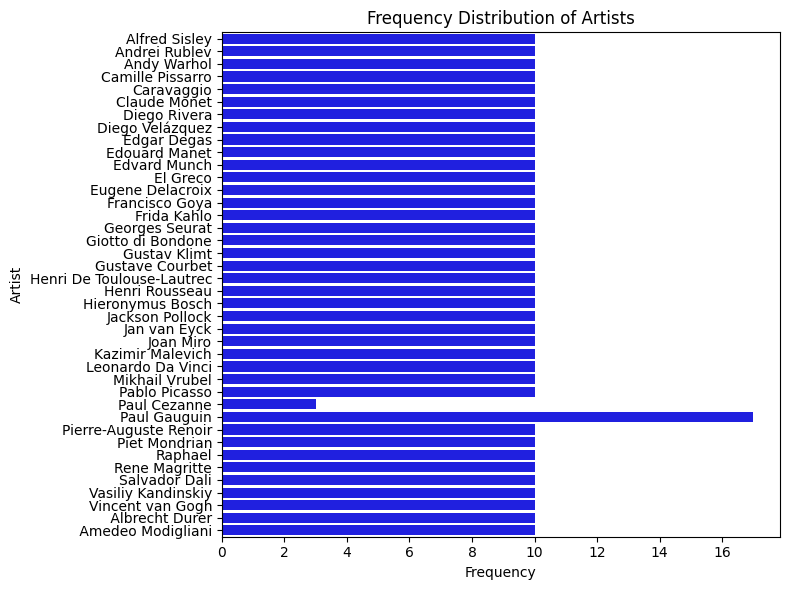


Figure 1

The column ‘Subject’ is a categorical variable with 21 unique values. The histogram below shows the frequency of each subject category in the dataset:

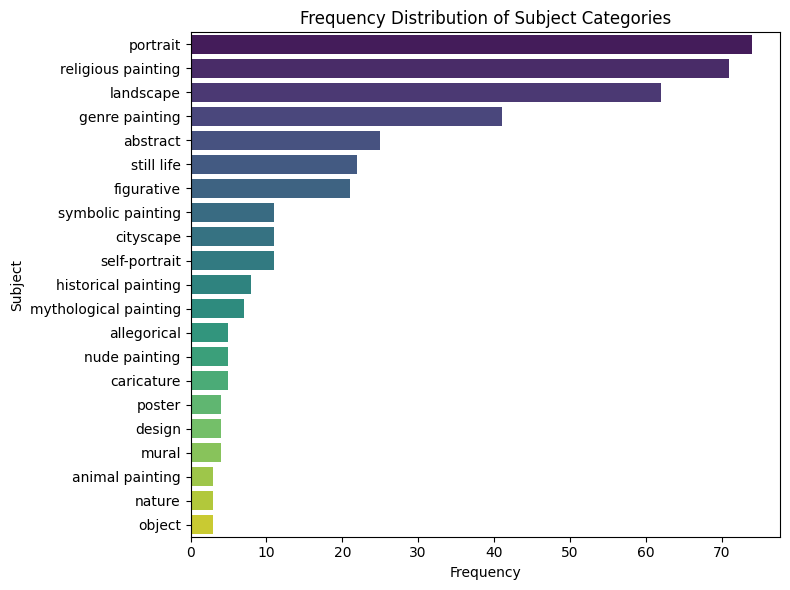


Figure 2

The most frequent subject category is “portrait” with 74 total observations, followed by the second most frequent category “religious painting” with 71 total observations. Other frequent categories include “landscape” with 62 observations, “genre painting” with 41 observations, and “abstract” with 25 observations.

The column ‘Style’ is a categorical variable with 28 unique values. The histogram below shows the frequency of each style category in the dataset:



Figure 3

The most frequent style category is “impressionism” with 69 total observations, followed by “post-impressionism” and “surrealism” each with 30 observations.

The column ‘Materials’ is a categorical variable with 21 unique values. The histogram below shows the frequency of each subject category in the dataset:

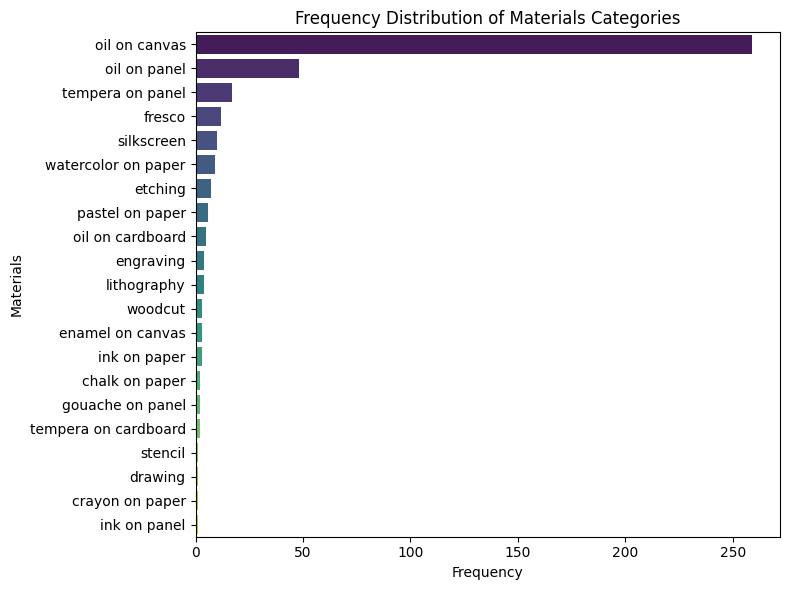


Figure 4

The most frequent materials category is "oil on canvas," with a total of 259 observations, far surpassing the second most frequent category, "oil on panel," which has 48 observations.

The column ‘Period’ is a categorical variable with 25 unique values. The histogram below shows the frequency of each subject category in the dataset:

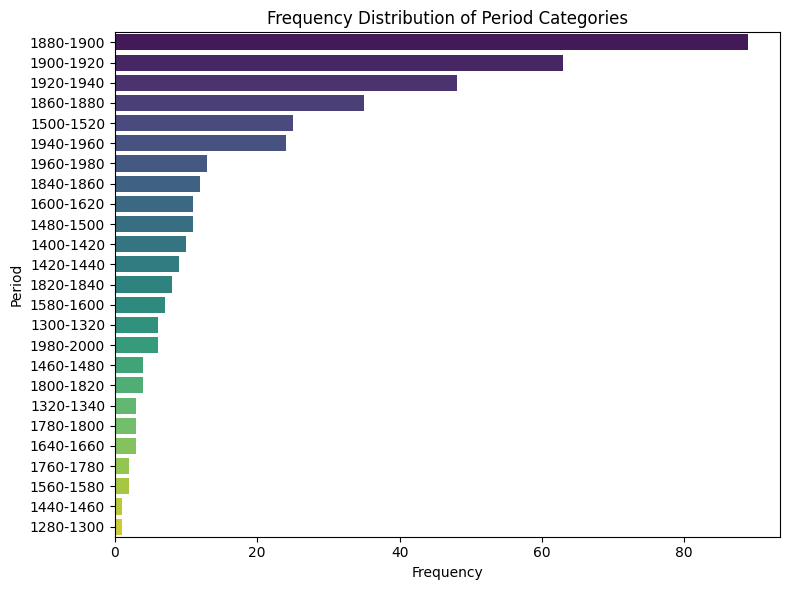


Figure 5

The most frequent period category is “1880-1900” with 89 observations, followed by “1900-1920” and “1920-1940” having 63 and 48 observations respectively.

# Data Preprocessing

Before we could feed our data into the neural network, we needed to apply some preprocessing steps. First, we had to connect the paths of the associated images to the dataset we had created in Excel. To do so, we wrote a script that matched each one of the 400 image IDs in Excel with the corresponding paths in the *resized.zip* folder and created a new column for it. After creating this dataset, the next step was to merge the ‘Start’ and ‘End’ columns into a new column named ‘Period’ and then delete the unnecessary columns.

The second step was to split our dataset into training and test sets for model evaluation. We decided that it would be best to stratify our data as well because we wanted to ensure that all the classes that were present in the test set would be represented in the training set as well also, we wanted to have in the training set as many distinct classes as possible so that the model could learn better. We encountered some difficulties in this process, as in some classes some categories didn’t have enough observations (less than 4) and the stratifying wasn’t possible. For that, we decided that those rare categories would only be included in the training set. In the end, we had a training set with 321 observations and 6 columns and a test set with 79 observations and 6 columns. After stratifying our data, it was necessary to encode our data as it was only categorical variables. Machine learning algorithms and deep learning neural networks require that input and output variables be numbers. This means that categorical data needs to be encoded into numbers before we can use it to fit and evaluate our model. Below we can see the distribution of some variables in both the training and test sets.

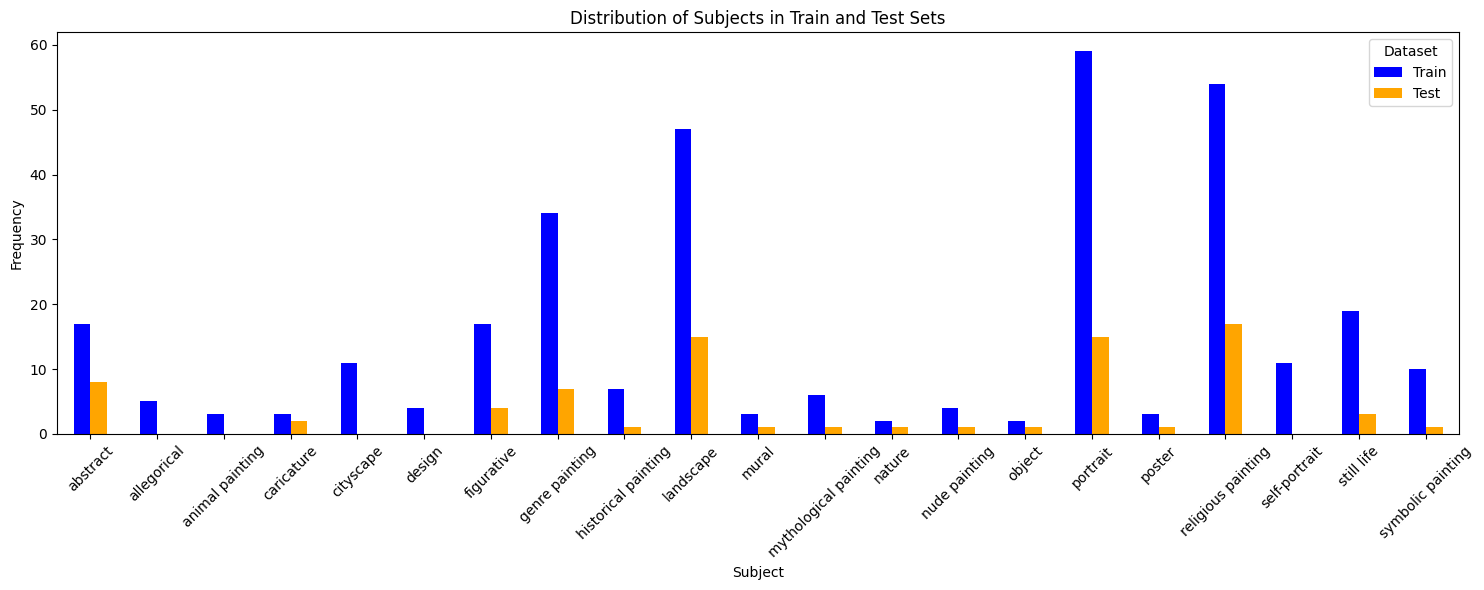


Figure 6

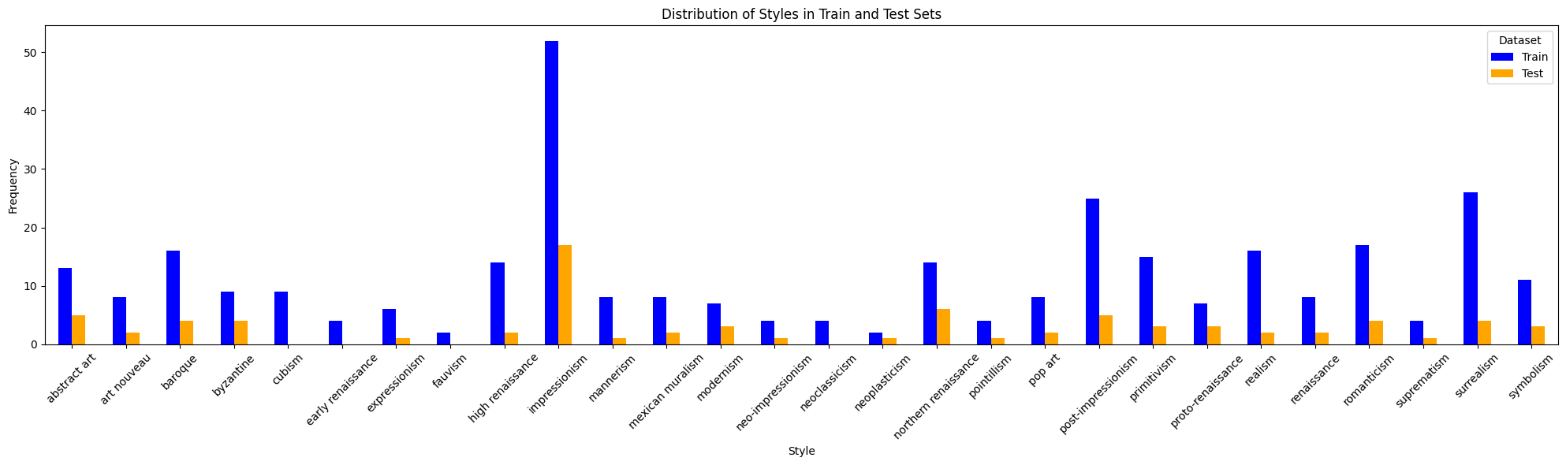


Figure 7

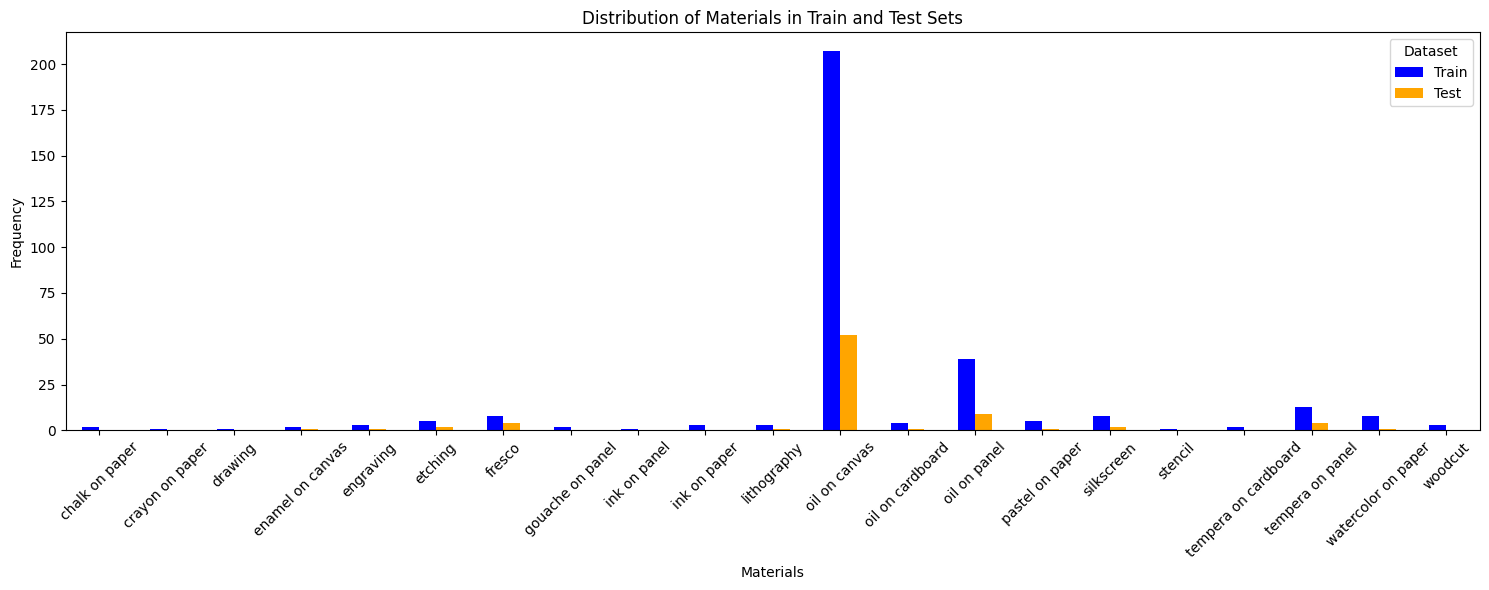


Figure 8

The third step was to preprocess the images. This step was very crucial for fitting a neural network model effectively, which required the data to be prepared and normalized. Before fitting a neural network model, we needed to ensure that the image size of the images was the size required by the ViT model and resize them. It was also very important that the input images were in RGB format to fit the transformer model, so we had to ensure that any black and white images were successfully transformed into RGB format. Finally, we had to normalize our data. Normalization of images ensures that all input features are on the same scale, preventing some features from dominating in the learning process. Normalization also improves the ability of the model to perform well on unseen datasets. For that, we divided the pixel values by 255 in order to scale the data to a [0, 1] range. In the end, the images were all in the shape (300x300x3) that the ViT model needed.

The final step of the preprocessing was the data augmentation. Data augmentation improves the model’s performance by generating variations of existing data. It also prevents overfitting because the model learns to generalize better. In this step, we applied various transformations to the images, including changes in the width and height, rotations, zooming, and horizontal flipping.

# Methodology

## 5.1 Introduction

As we stated earlier our goal was to create in the first part a deep-learning classification model that will predict the values of 5 target variables/labels which were the artist’s name, the subject and the style of the painting, the materials used for its creation and the period during which the specific painting was created. Our second and final goal was to find a recommendation system that based on the predicted label values of the classifier and the extracted image features will recommend the most similar paintings compared to a base painting that a person has already shown interest in to maximize the user’s engagement.

To construct the multi-output classifier, we used algorithms that belong to two different deep learning approaches Convolutional Neural Networks (the used algorithms were EfficientNetB3, VGG16, and VGG19) and Vision Transformer (the model that we used was based on Google’s ViT model). As we will show later in the model comparison and evaluation part the custom ViT classifier gave us better results than the CNN-based models. After selecting the ViT model as the most appropriate we used its architecture to extract both the visual and label features in an embedding format and calculate the similarity of different paintings based on these embeddings.

## Classification Models’ Description

### *5.2.1 EfficientNetB3*

The first model that we used was based on the EfficientNetB3 model. At first, we tried this model because at the beginning of our effort, we were oriented towards the CNNs architectures knowing that it is the most popular approach for Image Classification and Computer Vision tasks, and saw that the EfficientNetB3 architecture was used for an art-based image classification task in a HuggingFace project[[1]](#footnote-1) (Seo Jin Chang, sj21867, June 2024) so we thought that EfficientNetB3 was a good starting point. Also, the bibliography EfficientNet model family states that this architecture can compete with other CNNs algorithms by being significantly more efficient (Mingxing Tan & Quoc V. Le, 2020) so having limited resources we preferred to start with a lighter model.

To construct a custom model based on the EfficientNetB3 model we used the weights of the pre-trained base model on the ImageNet dataset, we set the image input size equal to 300x300 pixels which is recommended for this base model, and we removed the fully-connected layer that was on top of the Neural Network. Then, we implemented a global average pooling layer (this action was taken to reduce the output dimensions from the convolutional layers), set the layers of the base model as not trainable to avoid overfitting and excessive computational burden, introduced a dense layer for predicting the class (output) for each of the 5 labels where the number of units in each dense layer corresponding to the number of unique classes for each label[[2]](#footnote-2). Finally, we implemented the SoftMax activation function individually in each of the previously mentioned dense layers. For the training of the model (given that we wanted to construct a multi-output classification model) we used the categorical cross-entropy function for the calculation of the loss function for each one of the 5 labels and used accuracy for the model’s assessment. We used Adam optimizer, a batch size of 32 images, and a learning rate equal to 0.001 and we initially trained the model for 50 epochs.

After the training of the classifier, we saw that the model did not perform as wanted because the only target variable that was being predicted with a decent accuracy was the used materials (67.2% accuracy on the validation set) while the other outputs were being predicted poorly. Especially, the predictions for the artist’s name were particularly unreliable (the accuracy on the validation set never surpassed 5%) but also for the labels subject, style, and period (the accuracy on the validation set did not surpass 25%).

### *VGG16 and VGG19*

When we saw that the EfficientNetB3 model did not have a decent predictive ability we wanted to try some deeper CNNs architectures such as VGG16 and VGG19 that are used generally for image classification tasks but have also been used for art paintings’ classification projects in the past (C. Sandoval et al., 2019, Z. Yang, 2021 and O. Menis – Mastromichalakis et al., 2024) that aimed to classify paintings solely on their artistic style.

The architecture that we used for the custom VGG16 and VGG19 models is the same as the one we used for EfficientNetB3 because we used an input size of 300x300 pixels, frozen the pre-trained layers of the base models, and used the same dense output layers and the same loss and activation functions. We used again 50 epochs in the training of both models. The accuracy of the predictions that the VGG16 model has improved significantly for all the labels except the materials where the predictions remained as accurate as with the EfficientNetB3. Indicatively, the accuracy in the validation set during the best epoch (epoch 47) of the VGG16 model was 34.4% for the artist’s name, 68.8% for the materials, 32.8% for the period, 34.4% for style and 51.6% for subject. Compared to the previous models we saw improvement in the accuracy of the predictions of the VGG19 model related to EfficientNetB3 but its evaluation results underperformed compared to VGG16 as during its best epoch (epoch 47) the accuracies were 10.9% for the artist, 70.3% for materials, 23.4% for period, 28.1% for style and 40.6% for subject.

### *Vision Transformer (ViT)*

The last approach that we used was based on the pre-trained Google’s algorithm[[3]](#footnote-3) which has been trained on the ImageNet-21k dataset. We were inspired to use a custom classifier based on this approach by a HuggingFace project publication which refers to artwork classification (Oliver Schamp, oschamp, November 2023)[[4]](#footnote-4).

To build this custom model (after ensuring that we brought our images in a 224x224 pixels size and normalizing the values of the arrays that correspond to each image) we used the pre-trained ViTImageProcessor from HuggingFace Transformers[[5]](#footnote-5) library to extract the features from the images of the train and validation set separately. Then we applied an input layer (with dimensions 3, 224, 224) to ensure that the model’s input would be in the right dimensions and RGB format and connected the input layer with a custom layer that applied the pre-trained Google base model Vision Transformer model which processes the input as patches and extracts features using self-attention mechanisms. The output of the custom layer was used for further processing as we extracted the class token output (CLS token), which summarizes the information for the classification tasks.

The extracted features from the ViT model were then passed through a dense layer with 512 units and ReLU activation function. Finally, we connected this layer with 5 dense layers (activated with the SoftMax function) with each one corresponding to the prediction of one label (artist, materials, subject, style, and period) where the units that construct each layer are the same as mentioned before in the EfficientNetB3 description. This classifier is compiled using Adam optimizer with a learning rate equal to 0.001, using the categorical cross-entropy function to calculate the loss of information for each label and accuracy as an evaluation metric for all the outputs. Finally, we fitted the model with 50 epochs and a batch size equal to 32 images.

After training the model we saw an outstanding improvement in the predictive ability for all the target variables. More specifically, the accuracy in the validation set during the best epoch (epoch 40) of the custom ViT classifier was 53.1% for the artist’s name, 70.3% for the materials, 45.3% for the period, 65.6% for style and 60.9% for subject. So, we see that ViT performs fairly well for materials, style, and subject while even for the labels that struggle with (mainly period and less with the artist) it has a decent predictive ability compared to the previously mentioned models. In the following diagram bellow we present the above-mentioned ViT architecture:

Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, Σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 9

## 5.3 Models’ Evaluation and Comparison

As we can see from the table below the ViT custom classifier gives significantly better predictions for all the labels compared to the other three models. The accuracy that we present below is calculated based on the predictions that were made using the models’ weights that each model had during the epoch with the higher accuracies (on the test set). However, the ViT classifier outperforms the other models consistently no matter the epoch. We observe that the model has a decent predictive ability for all labels and especially for the variables subject, style, and materials performs fairly well. Also, we observe that ViT can capture information accurately about artists in contrast to the other models that struggled continuously with this label. Finally, we can see a small improvement in the predictive ability for materials. So, given all that we selected the ViT custom classifier.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Artist Accuracy (%)  (Best Epoch) | Subject Accuracy (%)  (Best Epoch) | Style Accuracy (%)  (Best Epoch) | Materials Accuracy (%)  (Best Epoch) | Period Accuracy (%)  (Best Epoch) |
| EfficientNetB3 | 4.7 | 20.3 | 20.3 | 67.2 | 20.3 |
| VGG16 | 34.4 | 31.6 | 34.4 | 68.8 | 32.8 |
| VGG19 | 10.9 | 40.6 | 28.1 | 70.3 | 23.4 |
| ViT | 53.1 | 60.9 | 65.6 | 70.3 | 45.3 |

Table 1

Below we also display 5 plots with ROC curves for evaluating the predictive ability (based on the weights that each model had during the last training epoch) of each model for each target variable. From these plots, we conclude again the superiority of the ViT classifier against the other models as the curve that corresponds to the ViT is closer to the upper left corner when compared to the other models for all the labels implying better predictive ability.

Εικόνα που περιέχει κείμενο, διάγραμμα, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 10

Εικόνα που περιέχει κείμενο, διάγραμμα, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 11

Εικόνα που περιέχει κείμενο, γραμμή, διάγραμμα, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 12

Εικόνα που περιέχει κείμενο, γραμμή, διάγραμμα, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 13

Εικόνα που περιέχει κείμενο, διάγραμμα, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 14

## 5.4 Fine-tuning ViT model

To enhance the performance of the Vision Transformer (ViT) model, which was selected as the model with the best performance, we fine-tuned it by introducing several modifications. In more detail, to further enhance generalization and prevent overfitting, we incorporated a dropout layer with a rate of 0.3, which randomly disables 30% of the neurons during each training iteration. The architecture completed with five output layers, each adjusted for a specific classification task, with a softmax activation function to produce probabilistic outputs for each label. To ensure the model converges efficiently, a learning rate scheduler was implemented. The training started with a learning rate of 1e-3 for the initial 40 epochs, which was then reduced to 1e-4 for the subsequent 40 epochs, and finally to 1e-5 for any epochs beyond 80.

As expected, the process of fine-tunning made the model more effective and more stabled. Specifically, the accuracy in the validation set during the best epoch (epoch 47) of the fine-tunned custom ViT classifier was 62.50% for the artist’s name, 78.12% for the materials, 43.75% for the period, 62.50% for style and 71.87% for subject. It is evident that the performance in the classes “Period” and “Style” has a small decrease, although is still at satisfactory levels. The fine-tuned model after the addition of the dropout layer has the following depiction.

Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, Σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα

## 5.5 Recommendation System

### *5.5.1 Key aspects and development of the recommendation system*

As we stated at the beginning of our report our final goal was to exploit the classifier model to calculate similarity (after defining a metric for it) based on the image data to find the most similar paintings and recommend them to gain the users’ interest and engagement. Some basic aspects that we took into consideration before building the recommendation system were the exclusion of the labels artist and period from the calculation of the similarity and the restriction of the number of paintings that were created by the same artist who made the base painting[[6]](#footnote-6) that the system can recommend. The exclusion of the artist and period labels was made because their inclusion would push the system to recommend more easily paintings created by the same artist, but this approach would be very simplistic and would cancel the meaning of the recommendation system because it is not useful to build an algorithm that recommends paintings of the same artist. So, we used the labels materials, style, and subject to gain information about the recommendations because we think that these metadata define a painting in a more useful way. Also, we restricted the system to be able to recommend only one painting of the same artist as the base painting, which means that if for example, we search for recommendations for one painting of Jan Van Eyck only one painting of him can be recommended again, we did this to have a larger diversity in our recommendations. We set the number of recommendations that the system would return to 5.

To build the system we used the custom ViT model that we trained before to extract visual embeddings that characterize the artworks from each one in the training and test set separately. We then combined these embeddings with one embedding that contained the values of the labels subject, style, and materials in the training set (we used a one-hot encoding approach for these labels) and one embedding that held the same values for the test set. After creating these combined embeddings that hold a unified representation of each artwork, we used them to calculate the similarities between paintings and make the corresponding recommendations. To calculate the similarities between the available embeddings/vectors we used the cosine similarity and created a similarity matrix for being able to track the 5 most similar paintings for each painting. The system was originally trained on the same training set that we used for the training of the classifier and evaluated on the test set where it makes its final recommendations. To achieve an honest evaluation the base painting came from the test set while the 5 recommendations came from the training set.

For the evaluation process, we calculated the average precision and average recall metrics which correspond to 84.3% and 19.4%. The above-mentioned metrics were calculated by averaging the values that precision[[7]](#footnote-7) and recall have for each base painting that exists in the test set so it’s the average precision and recall of 79 paintings. The high value of average precision indicates that the recommendations being made are very accurate while the low average recall means that there could be some relative paintings that are not included in the 5 recommendations. However, we concluded that a low recall may be inevitable in a recommendation system with 5 recommendations and that a big increase in the number of recommendations may lead to a less helpful function for the user because they will not have a quick and convenient search.

### *Recommendations Examples*

Below we display some examples of the recommendations being made given a base painting from the test set.

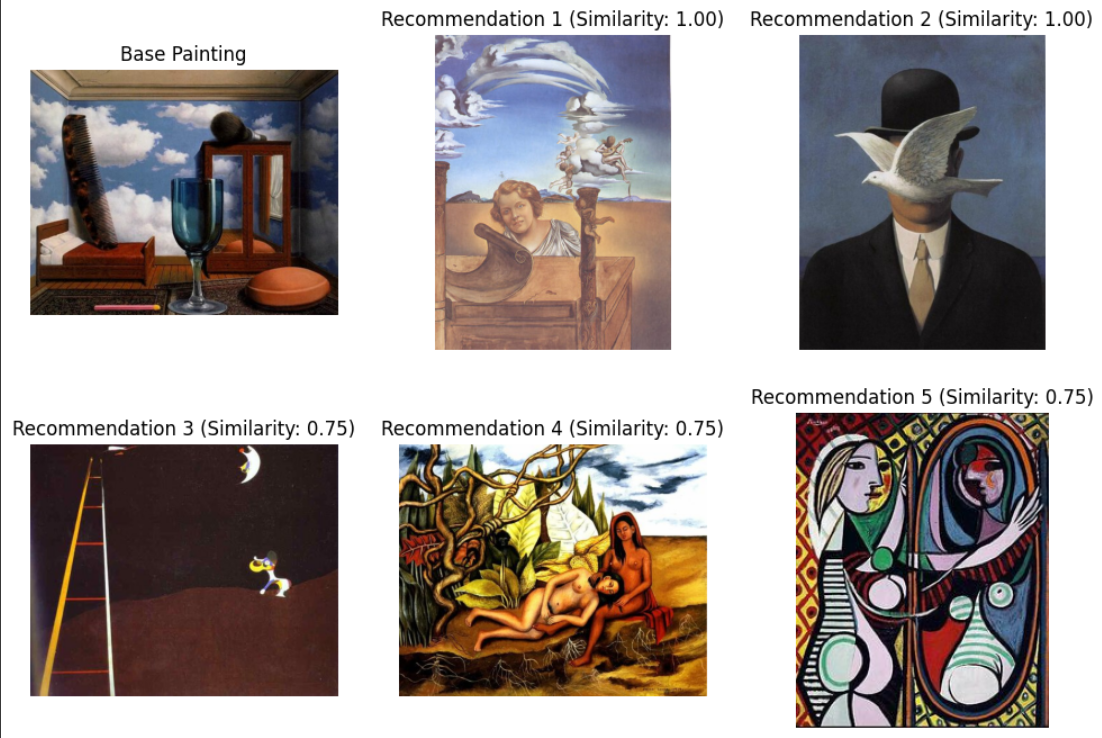


Figure 15



Figure 16

Εικόνα που περιέχει στιγμιότυπο οθόνης, ουρανός, εξωτερικός χώρος/ύπαιθρος

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 17

Εικόνα που περιέχει ανθρώπινο πρόσωπο, ρουχισμός, στιγμιότυπο οθόνης, κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 18

# Members/Roles and Time Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Phase** | **Member** | | | |
| **ANNA PAPADIMITRIOU** | **EVANGELOS LAKKAS-PYKNIS** | **STAMATOULA-GERASIMOULA MESOLORA** | **ELENI RALLI** |
| 1-2 | Data Collection and Pre-processing | The first step was collecting the necessary data to train our model.  Each team member selected 10 artists from a dataset in Kaggle, randomly choosing 10 paintings from each artist.  For each painting, we filled in the following columns: Image ID, Title, Artist, Subject, Style, Materials, Start, and End (grouped by approximately 20-year periods).  In total, we gathered 400 paintings to train our model.    In another file, we added an additional column for the image paths.    Then we filtered the images from the "resized" folder, keeping only the images we needed and recording their paths in the drive. | | | |
| Cleans the data, inconsistencies, and making sure all labels is formatted correctly. | Preparing the data splits for training and testing-stratified data split. | Label encoding on stratified train and test set. | Cleans the data, inconsistencies, and making sure all labels is formatted correctly. |
| 3-4 | Model Development | Works on image preprocessing (normalization, convert to RGB format, image resize) and implement image augmentation. | Hugging face search to choose some models to try out.  Focuses on training the EfficientNetB3,  Confusion Matrix based on the epoch with the best results, VGG16, and VGG19 models. Adjusting hyperparameters. | | Experimenting with early stopping to optimize performance.  Evaluates early model performance, monitoring metrics like accuracy and provides feedback to adjust the models accordingly. |
| 5-6-7 | Model Development (Phase 2), Recommendation System and Final Evaluation | Implementing the Vision Transformer (ViT) model for classification and compares its performance with other models. | | Develops the recommendation system after choosing our model incorporating embeddings from the best-performing model and implementing similarity algorithms like cosine similarity.  Evaluates the recommendation system, ensuring it meets the business goals. | |
| 8 | Report preparation and Presentation | All team members will focus on finalizing the project and preparing the final report.  Each member will contribute to writing different sections of the report. | | | |

# References

Icaro, 2019, Kaggle, Best Artworks of All Time, Collection of Paintings of the 50 Most Influential Artists of All Time. <https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time/data>

WikiArt - Visual Art Encyclopedia. [wikiart.org](https://www.wikiart.org/)

Artchive [artchive.com](https://www.artchive.com/)

Mehmet Oguz Kelek, Nurullah Calik & Tulay Yildirim, Painter Classification Over the Novel Art Painting Data Set via The Latest Deep Neural Networks, Procedia Computer Science, Volume 154, 2019, Pages 369-376. <https://doi.org/10.1016/j.procs.2019.06.053>

Kevin Alfianto Jangtjik, Mei-Chen Yeh, and Kai-Lung Hua. 2016. Artist-based Classification via Deep Learning with Multi-scale Weighted Pooling. In Proceedings of the 24th ACM international conference on Multimedia (MM '16). Association for Computing Machinery, New York, NY, USA, 635–639. <https://doi.org/10.1145/2964284.2967299>

W. R. Tan, C. S. Chan, H. E. Aguirre and K. Tanaka, "Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification," *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 3703-3707. <https://doi.org/10.1109/ICIP.2016.7533051>

Orfeas Menis – Mastromichalakis, Natasa Sofou and Giorgos Stamou, 2024, Deep Ensemble Art Style Recognition. <https://arxiv.org/html/2405.11675v1#bib>

Seo Jin Chang, sj21867, June 2024, HuggingFace, sj21867/ai\_art\_exp3\_efficientnetb3. <https://huggingface.co/sj21867/ai_art_exp3_efficientnetb3>

Mingxing Tan and Quoc V. Le, 2019, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Cornell University. <https://arxiv.org/abs/1905.11946>

Zihan Yang 2021 J. Phys.: Conf. Ser. 1774 012043, IOP Publishing Ltd., Classification of picture art style based on VGGNET. <https://iopscience.iop.org/article/10.1088/1742-6596/1774/1/012043/pdf>

C. S. Rodriguez, M. Lech and E. Pirogova, "Classification of Style in Fine-Art Paintings Using Transfer Learning and Weighted Image Patches," *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Cairns, QLD, Australia, 2018, pp. 1-7. <https://doi.org/10.1109/ICSPCS.2018.8631731>

C. Sandoval, E. Pirogova and M. Lech, "Two-Stage Deep Learning Approach to the Classification of Fine-Art Paintings," in *IEEE Access*, vol. 7, pp. 41770-41781, 2019. <https://doi.org/10.1109/ACCESS.2019.2907986>

Oliver Schamp, oschamp, November 2023, HuggingFace, oschamp/vit-artworkclassifier. <https://huggingface.co/oschamp/vit-artworkclassifier>

<https://huggingface.co/docs/transformers/model_doc/vit#transformers.ViTImageProcessor>

Note that for the creation of figure 9 we consulted the following references to get a better understanding of how the depiction of the ViT classifier should be.

[https://en.wikipedia.org/wiki/Vision\_transformer#](https://en.wikipedia.org/wiki/Vision_transformer)

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, 2021, Cornell University, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. <https://arxiv.org/abs/2010.11929>

1. <https://huggingface.co/sj21867/ai_art_exp3_efficientnetb3> [↑](#footnote-ref-1)
2. This means we have 40 units in the dense layer corresponding to the artist prediction because our dataset contained 40 distinct artists. Similarly, the dense layers for subject, style, period, and materials used, have 21, 28, 21, and 25 units respectively. [↑](#footnote-ref-2)
3. <https://huggingface.co/google/vit-base-patch16-224-in21k> [↑](#footnote-ref-3)
4. <https://huggingface.co/oschamp/vit-artworkclassifier> [↑](#footnote-ref-4)
5. <https://huggingface.co/docs/transformers/model_doc/vit#transformers.ViTImageProcessor> [↑](#footnote-ref-5)
6. With the term “base painting” we mean the painting based on which we want to make the recommendations [↑](#footnote-ref-6)
7. Please note that for the calculation of these metrics we defined that the paintings that should be accounted as True Positives are those with a cosine similarity>0.55 [↑](#footnote-ref-7)