AUTOMATIC CLASSIFICATION OF WORKS OF ART

IMA206
Computational photography & Patch methods
Télécom Paris

AUTHOR

Hugo CROCHET

SUPERVISORS

Henri MAITRE
Loïc LE FOLGOC

Contents

Chapter 1

Intoduction

Recent technological advancements and cost reductions have enabled large-scale digitization of visual art collections, like WikiArt. This, along with progress in deep learning and computer vision, offers new opportunities to develop automatic tools for analyzing and understanding visual arts. In this context, we aim to tackle the automatic classification of artworks using deep learning and state-of-the-art algorithms.

In this project, we have a dataset comprising works from a hundred artists and a total of 3060 paintings. Our objective is to develop a method to classify each painting into a specific artistic style and visualize these styles in a coherent and informative manner.

To achieve this, we leverage ResNet, a powerful convolutional neural network (CNN) known for its deep architecture and impressive performance on image recognition tasks. By fine-tuning ResNet with our dataset, we aim to extract meaningful features from the paintings that can help distinguish between different artistic styles.

Once features are extracted, we utilize t-distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique, to project the high-dimensional features into a two-dimensional space. This visualization allows us to intuitively explore the relationships between paintings and artistic styles.

For clustering, we employ Gaussian Mixture Models (GMM) to group paintings into distinct styles based on the extracted features. GMM is chosen for its flexibility in modeling the distribution of features and its capability to form overlapping clusters, which is suitable for the inherently subjective nature of art classification.

Through this process, we aim to create a visual representation, such as a graph or a map, that showcases the distribution of artistic styles across the paintings in our dataset. This representation will not only provide insights into the stylistic relationships between different artworks but also serve as a valuable tool for art historians, curators, and enthusiasts.

The following sections of this report will detail the methodology, implementation, and results of our project. We will begin with a review of relevant literature and background information on art classification techniques. This will be followed by a comprehensive description of our dataset, the preprocessing steps undertaken, and the specific modifications made to ResNet for our purposes.

We will then delve into the clustering process, presenting the results of our t-SNE visualizations and GMM clusters. A thorough analysis of these results will highlight the effectiveness of our approach in classifying and visualizing artistic styles. Finally, we will discuss the implications of our findings, the limitations of our current method, and potential directions for future research.

Chapter 2

Literature Review

Overview of Art Classification Techniques

Art classification has seen significant advancements with the introduction of various machine learning and deep learning techniques. Researchers have explored different methods to automatically classify and understand artistic styles, leveraging the power of convolutional neural networks (CNNs), metric learning, and unsupervised learning approaches. This section provides an overview of key methodologies and their contributions to the field of art classification.

In [1], Saleh and Elgammal worked on large-scale classification of fine-art paintings. They investigated the applicability of metric learning approaches and the performance of different visual features for learning similarity in a collection of fine-art paintings. They implemented meaningful metrics for measuring similarity between paintings, learned in a supervised manner to group paintings by Style, Genre, and Artist. Their experiments on a large dataset demonstrated that Classeme features showed superior performance for classification tasks across different metrics, including Boost metric and Information Theoretic Metric Learning. Additionally, they found that Large-Margin Nearest-Neighbor metric learning achieved the best performance when using feature fusion, significantly improving classification accuracy while reducing feature vector size.

Wynen, Schmid, and Mairal introduced in [2] an unsupervised learning approach for discovering, summarizing, and manipulating artistic styles using archetypal style analysis. This method, akin to sparse coding with a geometric interpretation, learns a dictionary of archetypal styles from deep image representations of artworks. Each new image's style is approximated by a sparse convex combination of these archetypes, allowing for interpretation and manipulation of the image's style components. This technique enables style enhancement, transfer, and interpolation between multiple archetypes, offering a flexible tool for art analysis.

In the paper [3], Folego, Gomes, and Rocha proposed a method for automatically identifying Van Gogh's paintings through a pipeline consisting of four steps: dividing images into smaller patches, extracting features using a CNN, applying a patch classifier, and using patch classification scores for a final response. Their approach leverages the detailed analysis of image patches to improve the accuracy of painting identification, demonstrating the effectiveness of patch-based feature extraction and classification.

Some provided an overview of deep learning approaches to pattern extraction and recognition in paintings and drawings. That's what Castellano and Vessio did in [4]. They discussed various visual art datasets and deep learning methods such as AlexNet, VGG, ResNet, RNNs, and GANs. Their comprehensive review highlighted current research trends and outlined high-level directions for further exploration in the field, emphasizing the potential of advanced deep learning models in art classification and pattern recognition.

In addition to these researchs about CNNs, Gatys, Ecker, and Bethge introduced in [6] a neural algorithm for artistic style transfer, demonstrating how CNN feature representations can independently process and manipulate the content and style of images. Their method, a texture transfer algorithm constrained by deep image representations, reduces style transfer to an optimization problem within a single neural network. This approach combines parametric texture modeling with pre-image search to generate new images, effectively separating content and style for advanced image manipulation.

In their study on fine-art paintings classification, Tan et al. found that fine-tuning an ImageNet pre-trained CNN yielded the best results, outperforming state-of-the-art methods. Their analysis in [7] revealed challenges in learning from paintings due to feature variations between lower and higher levels. They proposed future work to design better CNN models for painting classification and explore different visualization techniques to enhance understanding of feature extraction.

Van Noord, Hendriks, and Postma developed PigeoNET in [8], a CNN based on the AlexNet architecture, with an added visualization component for artist attribution tasks. Their system demonstrated high accuracy in using visual characteristics to assign artworks to the correct artist. Their results highlighted the effectiveness of feature learning systems in artist attribution and visual characteristic analysis.

Finally, a very intersting paper for our work is [5], by Giovanna, Castellano and Gennaro. This paper introduces DELIUS (DEep learning approach to cLustering vIsUal artS). DELIUS aims to address the challenges of clustering artworks without relying on subjective annotations. In facts, the researchers resized and normalized the images to preprocess them, in order to fed a DenseNet121 to extract visual features. Then, the global average pooled features are provided as input to a deep embedded clustering model to perform clustering. Eventually, the embedded features are projected in two dimensions with t-SNE for visualization purposes. This paper is thereby offering valuable tools for art analysis and curation.

These studies collectively showcase the diverse approaches and significant progress in the field of art classification, illustrating the potential of machine learning and deep learning techniques in advancing our understanding and analysis of artistic styles. The one that inspired us the most is the last cited article. It gave us some solid foundations to start with.

Chapter 3

Methodology

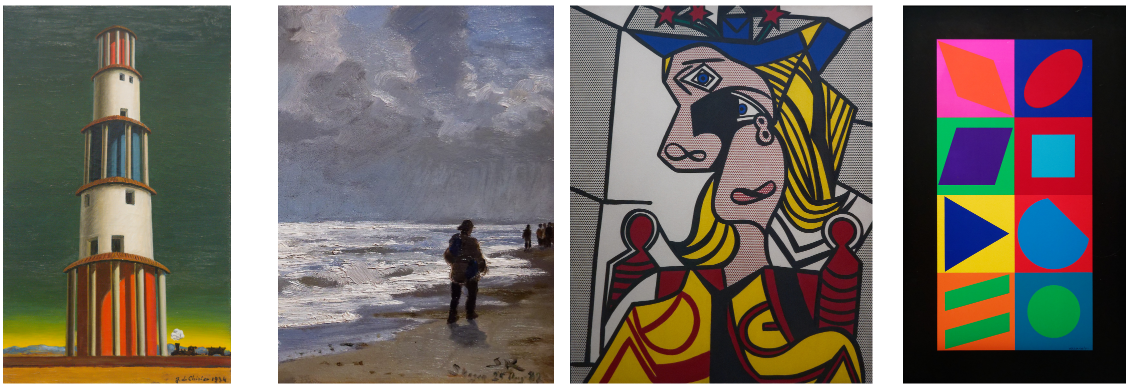
3.1 Dataset

Our dataset encompasses a broad historical range, spanning from the 1400s to the present day. This extensive timeframe includes works from the Italian primitives to contemporary art, thereby capturing a wide spectrum of artistic styles and movements. The dataset features works from 100 painters, providing a diverse and representative sample of art history.

In total, the dataset contains 3060 paintings. However, it is important to note that the distribution of paintings among the painters is not equal. Some painters are significantly more represented than others. This imbalance introduces an additional layer of complexity to the classification task, as the model must learn to recognize styles from both well-represented and less-represented painters.

The paintings within this dataset cover almost all recognized art styles, including but not limited to Renaissance, Baroque, Rococo, Neoclassicism, Romanticism, Realism, Impressionism, Post-Impressionism, Modernism, and various contemporary styles. This diversity makes the dataset particularly rich and challenging, providing an excellent opportunity to test the robustness and versatility of our classification and clustering methods.

By using this dataset, we aim to explore the capabilities of deep learning techniques in identifying and distinguishing between a wide range of artistic styles, ultimately contributing to the development of tools that can assist in art historical research and education.



(a) Chirico (b) Kroyer (c) Lichtenstein (d) Vasarely

Figure 3.1: Four paintings from the dataset.

3.2 Model Architecture

Our choices

After several research on various papers, we conclude that we wanted to challenge this task with two Neural Networks : ResNet and DenseNet.

On the one hand, ResNet was chosen for this project due to its proven effectiveness in various image classification tasks, particularly those involving complex and high-dimensional data such as paintings. Residual blocks from ResNet allows capturing the intricate details and subtle variations present in fine-art paintings. Furthermore, ResNet's architecture, specifically ResNet-18, offers a good balance between depth and computational efficiency, making it a suitable choice for the dataset size and the computational resources available.

On the other hand, DenseNets have several advantages over other state-ofthe-art architectures: they alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

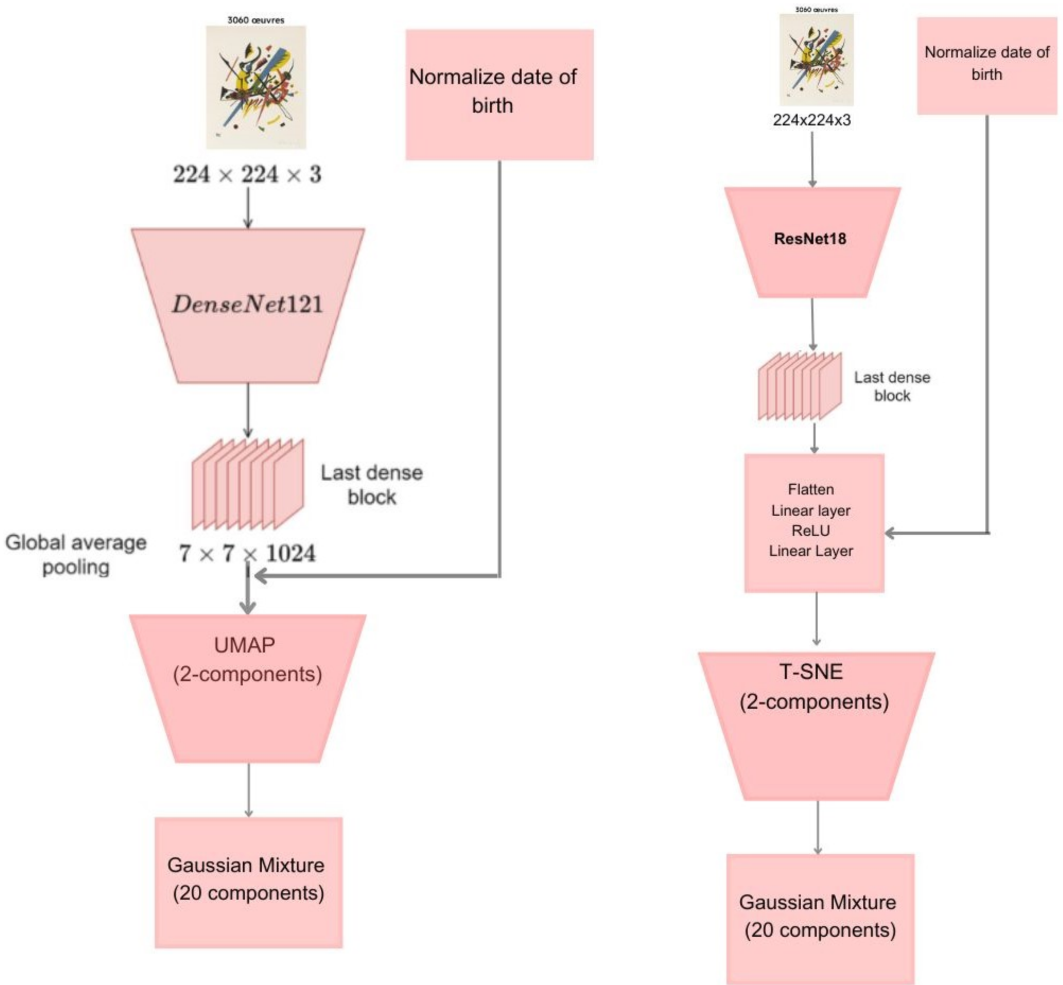


Figure 3.2: Model Architecture: Comparison of ResNet and DenseNet

Preprocessing Steps

Preprocessing is a critical step in preparing the images for input into the ResNet model. The preprocessing pipeline includes resizing, normalization, and formatting of images to ensure consistency and optimal performance of the model. Both DenseNet and ResNet take 224x224 images as network inputs. In our dataset, all our images are larger than 1500x1500 pixels. So we tried out these 3 preprocessing methods:

- Resize to 224×224 and Normalization
- Crop an area of size 224×224 centered on the image center
- Reduce the image size while maintaining the same length-to-width ratio to obtain an image of size $224 \times N$; before cropping an area of size 224×224 centered on the image center

The results obtained are very similar whatever the method chosen. As stated in some of the articles cited in the bibliography, we have therefore decided to keep the first preprocessing method, as it is the simplest.

The following transformations were applied:

Listing 3.1: Preprocessing Steps in Python

```python
normalise_resize = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])  # normalization
])
```

- **Resizing:** Each image is resized to 224×224 pixels to match the input size expected by ResNet-18.
- **Normalization:** Images are normalized using the mean and standard deviation of the ImageNet dataset. This step scales the pixel values to a range that is suitable for the pre-trained model and helps in accelerating the convergence of the training process.

Fine-tuning ResNet

Fine-tuning a pre-trained ResNet model involves several steps to adapt the model to the specific task of classifying paintings by their painters. The ResNet-18 model, pre-trained on the ImageNet dataset, was used as the starting point. This pre-training provides a strong foundation of low-level feature extraction due to the diverse nature of the ImageNet dataset.

To adapt ResNet-18 for our specific task, the following modifications were made:

- **Freezing Pre-trained Layers:** All the parameters of the ResNet-18 model were frozen to retain the learned features from ImageNet. This was done in order to prevent overfitting as one of our goal was to be able to predict the style of an unknown painting but also to gain some time and memory during training. The fact that we have a small database also motivate that choice.
- **Custom Fully Connected Layer:** The final fully connected layer of ResNet-18 was replaced to match the number of classes in our dataset (100 painters). The new fully connected layer consists of:
  - **Flatten Layer:** To convert the multi-dimensional tensor into a 1D tensor.
  - **Linear Layer:** A fully connected layer with 256 units and ReLU activation to introduce non-linearity and learn complex patterns.
  - **Output Layer:** Another linear layer with 100 units (one for each painter), which serves as the final classification layer.

This configuration allows the model to leverage the robust feature extraction capabilities of ResNet-18 while adapting to the specific task of painting classification. By freezing the pre-trained layers, the model retains the general features learned from ImageNet, while the added layers focus on learning task-specific features.

3.3 Weights adjustement

In our dataset, the number of paintings per painter is not equally distributed, leading to imbalanced classes. This imbalance can negatively affect the performance of our CNN, as the model might become biased towards painters with more paintings, thereby neglecting those with fewer representations. To address this issue, we employ weight adjustment in our training process.

When training our CNN, we assign a weight to each painter that is inversely proportional to the number of paintings they have in the dataset. These weights are then incorporated into the cross-entropy loss function

In practice, when the CNN processes a painting, it predicts the painter from 100 possible painters. The weighted cross-entropy loss ensures that the loss contribution of each painting is scaled according to the painter's weight.

By implementing this weight adjustment strategy, we aim to enhance the classification performance of our CNN, ensuring a fairer and more accurate prediction of artistic styles across all painters in our dataset.

3.4 Feature extraction

For the feature extraction phase of our art painting classification task, we employ transfer learning, leveraging the strengths of well-established convolutional neural network architectures such as DenseNet and ResNet. These models are chosen due to their proven effectiveness in various image recognition and classification tasks.

In our approach, we utilize the pre-trained DenseNet and ResNet models, but we exclude their final classification layers. Instead, we focus on the penultimate layer, which provides a rich set of high-level features. This layer outputs a feature vector that encapsulates the essential information needed for classification while discarding the model-specific classifier.

By removing the final classification layer, we ensure that the extracted features are more general and can be effectively used with our custom classifier tailored to the art painting classification task.

Instead, we could have also explored the traditional approach of manually computing features. This involved extracting various low-level and high-level features from the images, such as:

- **Color Features:** Analyzing the distribution and statistics of colors within the image.
- **Edge Detection:** Identifying the edges within the painting using Sobel edge detectors to capture the outlines and significant transitions.

Lastly, we've come into possession of the artists' dates of birth. Thus, for the two above methods, we wondered whether it would be a good idea to add them as features. Given the heavy weight of the year (between 1,400 and 2,000, as opposed to between 0 and 1 for the other features), we tried several ways of implementing it without it taking up too much importance :

- Add of date of birth
- Add of date of birth divide by 10
- Add of date of birth then normalize with StandardScaler or MinMaxScaler

The latest did not reaaly show good results as the dates distribution is nor Gaussian nor Uniform.

3.5 Dimensionality Reduction

To visualize the high-dimensional features extracted from our paintings, we employ t-distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique. Additionally, we have tried U-MAP instead of t-SNE. Initially, we use DenseNet121 or ResNet18 models with weights pre-trained on ImageNet. By removing the classifier layer from these models, we can represent each painting as a 1024-dimensional feature vector for our DenseNet Model and 100 for the fine-tuned ResNet one.

To make the analysis more manageable, we project these feature vectors into a two-dimensional space using the t-SNE or U-MAP algorithms. We also tried to implement a PCA with t-SNE or U-MAP to see if this had an impact or not. This transformation allows us to visualize the relationships between the paintings in a more intuitive and interpretable manner. However, directly analyzing all 3060 paintings in the 2D space can be overwhelming and not very insightful. To simplify this analysis, we hypothesize that each artist tends to have a relatively consistent style. Based on this assumption, we compute a single representative point for each artist in the 2D space. This point is the mean of the 2D coordinates of all the paintings by that artist. While this assumption holds well for many artists, it might be less accurate for those with highly varied styles, such as Picasso or Mondrian.

The resulting graph contains a hundred points, each representing an artist.

t-SNE is particularly suitable for small datasets due to its ability to create visually appealing representations of the data's local structure. However, it can become computationally intensive and less effective as the dataset size increases. In contrast, UMAP not only preserves both local and global structures of the data but also scales better with larger datasets and is generally faster.

Given our dataset size and the need for effective visualization, t-SNE's suitability for smaller datasets makes it an excellent choice for our initial analysis.

3.6 Clustering

To understand the distribution of artistic styles among the painters in our dataset, we apply Gaussian Mixture Models (GMM) for clustering. GMM is a probabilistic model that assumes all data points are generated from a mixture of several Gaussian distributions with unknown parameters. This method is particularly suitable for our task as it allows for the modeling of overlapping clusters, which is important given the fluid and often subjective nature of artistic styles.

After reducing the dimensionality of our feature vectors to a two-dimensional space using t-SNE, we obtain a condensed representation where each point corresponds to an artist, represented by the mean of their paintings' 2D coordinates. With these points, we employ GMM to identify clusters of painters that share similar stylistic characteristics.

GMM works by estimating the parameters of the Gaussian distributions through an iterative process known as Expectation-Maximization (EM). This process helps to determine the probability that each point belongs to a particular cluster, enabling the identification of groups of painters who exhibit similar styles. By using GMM, we can account for the inherent variability and overlap in artistic styles, providing a more nuanced classification than hard clustering methods.

The resulting clusters offer a very graphic way to visualize the painters, among their main artistic movements.

Chapter 4

Results

4.1 Difference between DenseNet and ResNet, with Umap and TSNE

In Figure 4.5, we present the clustering results obtained using different models and projection algorithms, incorporating the date of birth as a feature. It is evident that t-SNE performs significantly better than UMAP. For instance, in the first UMAP plot, Lichtenstein, Poussin, and Da Vinci are incorrectly grouped in the same cluster, which is highly inaccurate. In contrast, the t-SNE plot using ResNet demonstrates much better results, placing Caravaggio, Brueghel, Botticelli, and Veronese close to each other, even if they are not in the same cluster. Additionally, Cézanne, Sisley, and Monet are correctly clustered together under Impressionism.

The differences with DenseNet are minimal. While the UMAP with DenseNet is not as poor as with ResNet, the t-SNE projections with DenseNet do not perform as well as those with ResNet.

As shown in Figure 4.6, the manually extracted features did not yield satisfactory results. The main reason is probably because they were not sufficient to translate the complexity and richness of art paintings. Art is characterized by intricate patterns, subtle textures, and unique styles that are difficult to capture using simple hand-crafted features. With more complex features, we may have better results but we preferred to focus on deep learning instead of computing lots of hand-crafted features.

4.2 ResNet Results

After employing Gaussian Mixture Model (GMM) clustering to identify clusters of paintings and painters from the same movement, we attempted to plot a map, in Figure 4.7 based on the average position of each painter's works. This preliminary map yielded promising results, with paintings from the Rococo and Baroque movements generally clustering together. However, some points appeared slightly displaced, likely due to changes in the artistic style of painters over their careers.

To refine our analysis, we incorporated birthdates as an additional feature in the mapping process. This enhancement significantly improved the topology of the map, allowing us to trace a clear timeline through the artistic movements as demonstrated in Figure 4.8. This highlights the importance of including temporal information, as an artist's style is often influenced by their contemporaries.

By assigning a style to each painter and updating the map accordingly, we achieved even better results, with more coherent and meaningful clusters emerging among the 3060 paintings.

Recognizing the dominant influence of birthdates (see Figure 4.9, we also experimented with decimal normalization to reduce this effect as shown in the Figure 4.10. The resulting map remained satisfactory, providing a balanced view of stylistic clusters.

We've also tried to put a painting through the network to see where our pipeline would place it in the

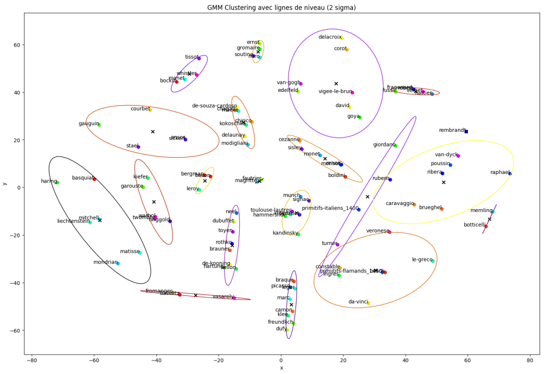


Figure 4.1: ResNet t-SNE

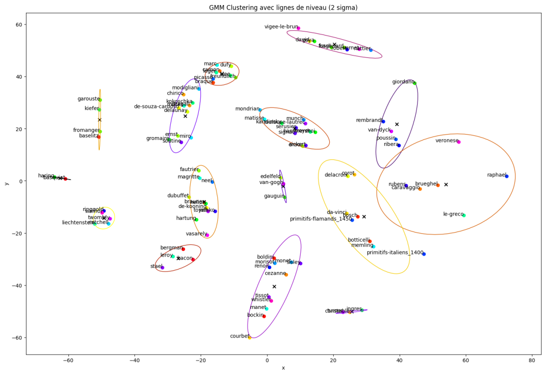


Figure 4.2: DenseNet t-SNE

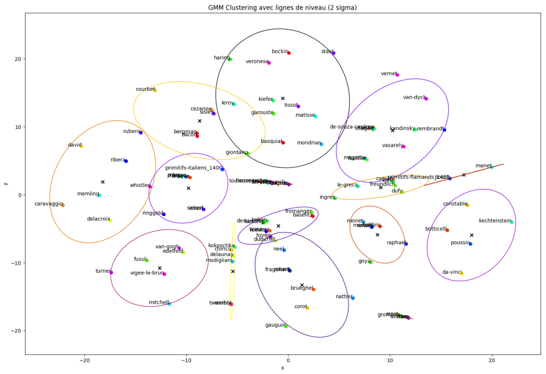


Figure 4.3: ResNet UMAP

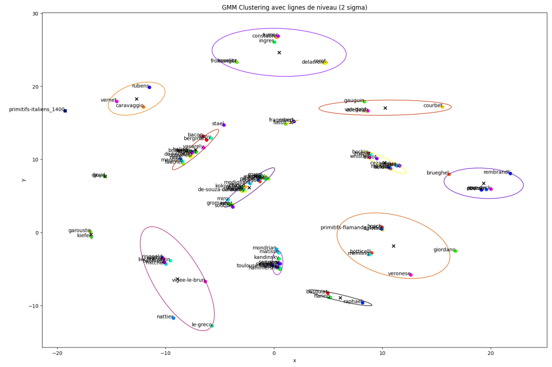


Figure 4.4: DenseNet UMAP

Figure 4.5: Comparison of t-SNE and UMAP with ResNet and DenseNet feature extraction

2D space. The main issue was the fact that t-SNE doesn't have a transform function to use on one sample. We thus had to implement a sort of transform function to do so. We annotated the map with the names of selected paintings, facilitating the identification of style clusters.Here, we put a painting named "Hollow in the snow" by Armand Guillaumin as you can see in the Figure 4.11. This artwork is a impressionism painting, and our model placed it with paintings like (see Figure 4.12) "Eugene Manet and His Daughter in the Garden" by Morisot or "Holger drachmann portrait, 1895" by Kroyer. These are other Impressionism paintings, which allow us to visually underscores the clustering of paintings by style, confirming the effectiveness of our approach.

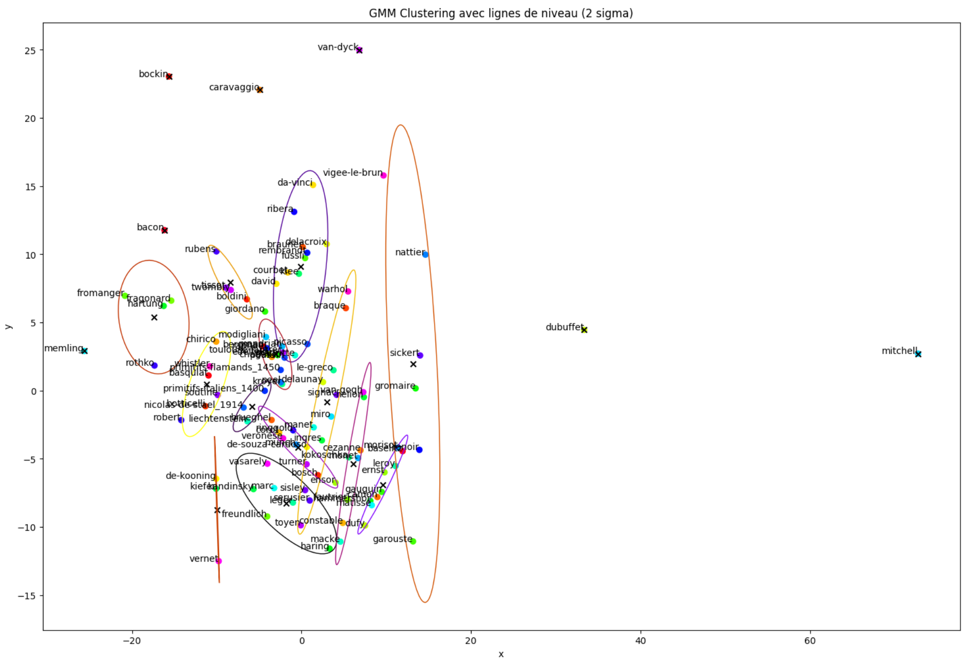


Figure 4.6: Clustering made with manually extracted features

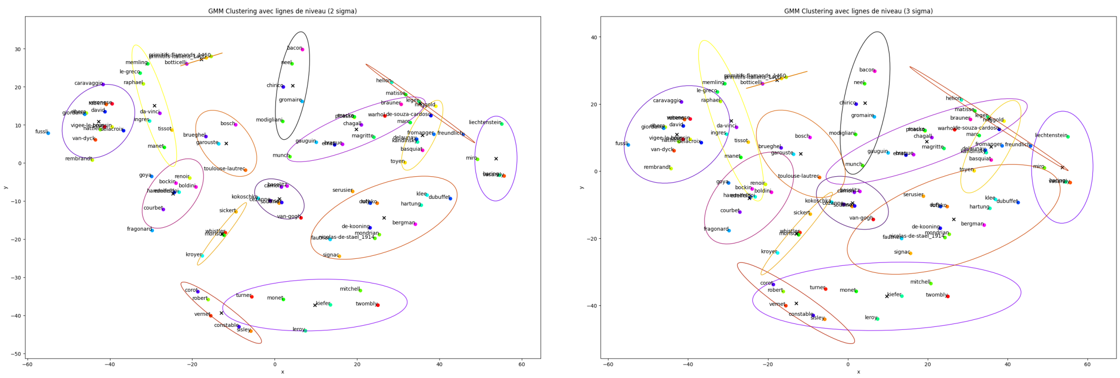


Figure 4.7: Effects of the parameter of the gaussian on clusters

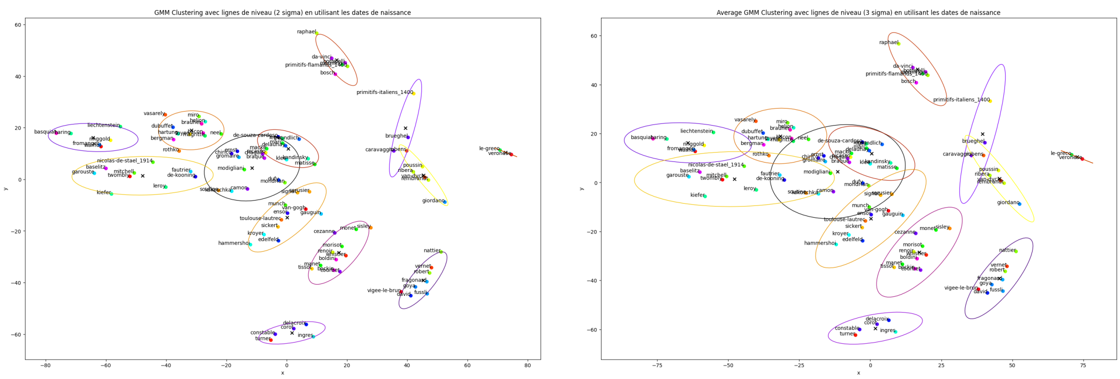


Figure 4.8: Effects of the parameter of the gaussian on clusters, with birthdate as a feature

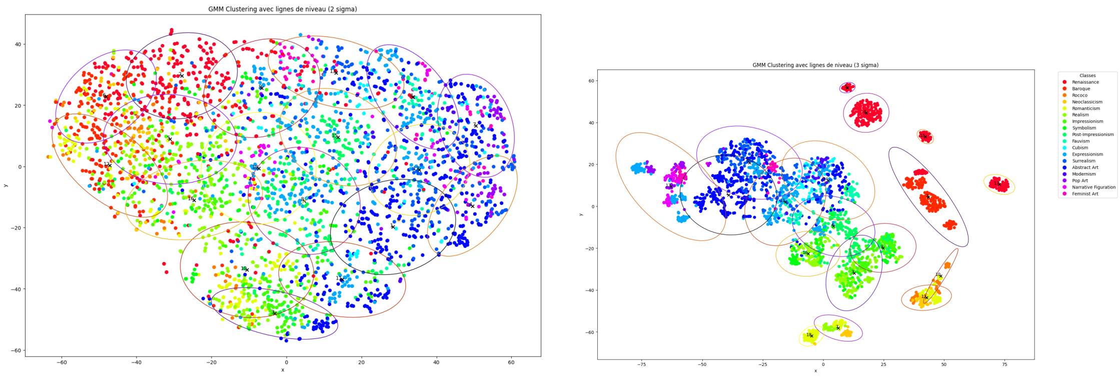


Figure 4.9: Topology difference when adding birthdate as a feature

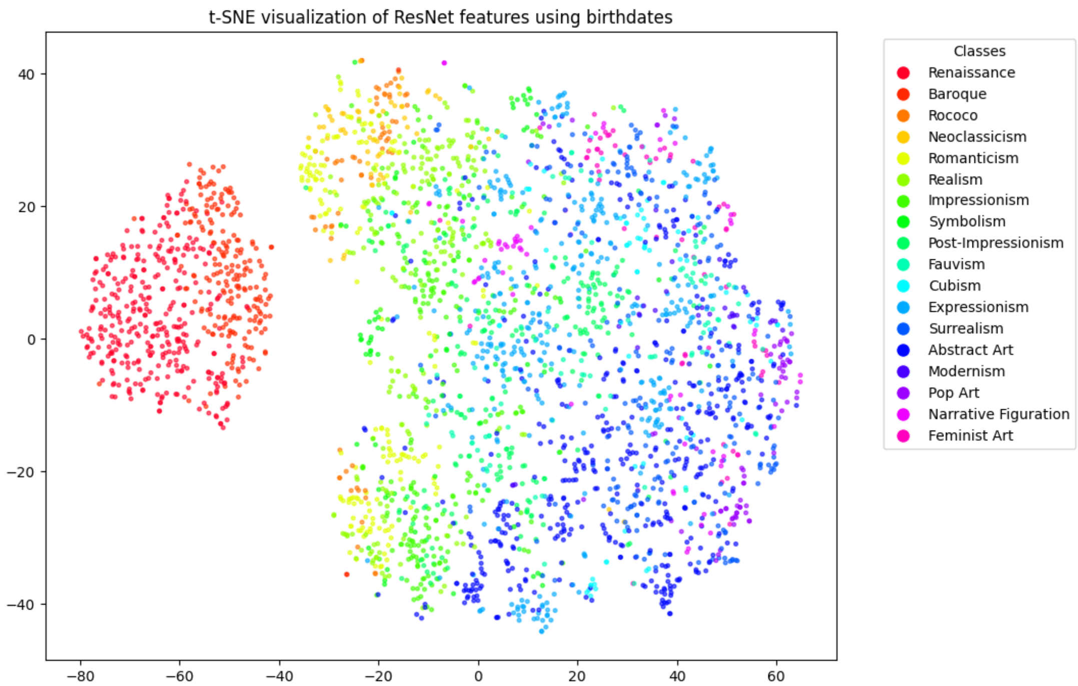


Figure 4.10: Difference due to normalization of the birthdate

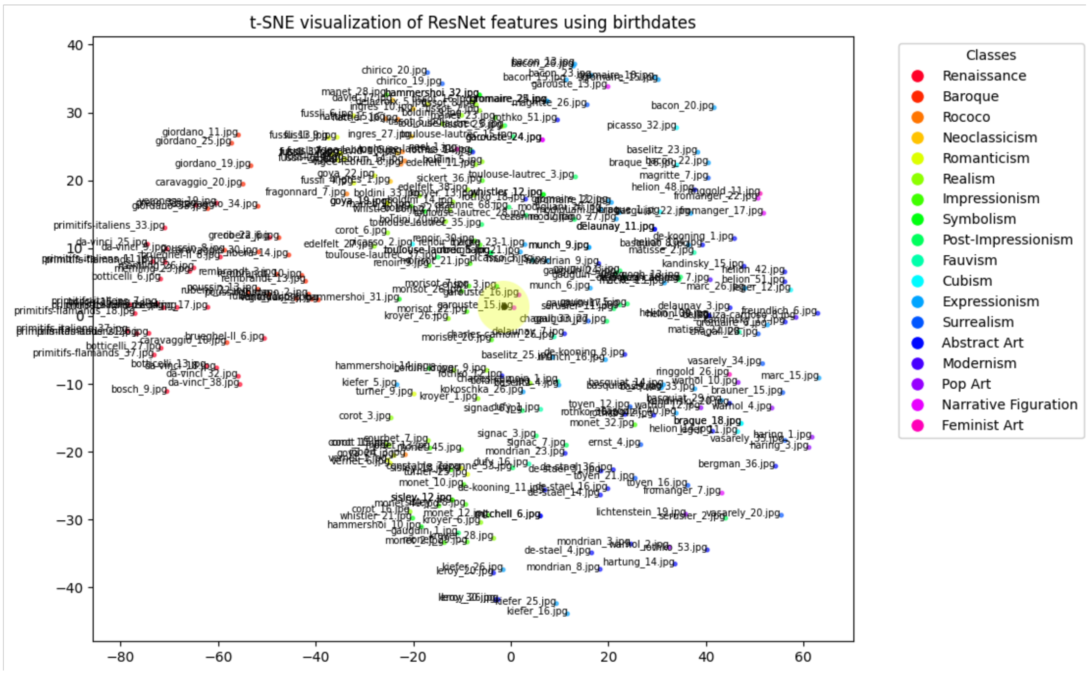


Figure 4.11: Visualization of "Hollow in the snow" by Armand Guillaumin in the clustering 2D-space

Figure 4.12: Similar paintings of Hollow in the snow

Chapter 5

Conclusion

In conclusion, we are pleased with the results obtained using the GMM clustering method to group paintings and painters by their artistic movements. Achieving greater precision without the risk of overfitting is challenging, but our approach has demonstrated that incorporating birthdate as a feature is essential. This is particularly important as some artistic styles overlap over time.

For future work, we could enhance our classification by integrating additional features such as shape, texture, and color, alongside those obtained from the model. Incorporating metadata could refine our classification, providing a more nuanced understanding of the relationships between artworks. Furthermore, exploring a larger collection of artworks from various styles would be beneficial, as some styles exhibit significant variation that our current dataset may not fully capture. This would allow for a more comprehensive analysis and better representation of the diversity within each artistic movement.

Bibliography

[1] Saleh Babak and Elgammal Ahmed. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. 2015.

[2] Wynen Daan, Schmid Cordelia, and Mairal Julien. Unsupervised learning of artistic styles with archetypal style analysis. 2018.

[3] Guilherme Folego, Otavio Gomes, and Anderson Rocha. From impressionism to expressionism: Automatically identifying van gogh's paintings. *2016 IEEE International Conference on Image Processing (ICIP)*, pages 141–145, 2016.

[4] Castellano Giovanna and Vessio Gennaro. Deep learning approaches to pattern extraction and recognition in paintings and drawings: an overview. 2021.

[5] Castellano Giovanna and Vessio Gennaro. A deep learning approach to clustering visual arts. 2022.

[6] Gatys Leon, Ecker Alexander, and Bethge Matthias. Image style transfer using convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] Wei Ren Tan, Chee Seng Chan, Hernán E. Aguirre, and Kiyoshi Tanaka. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3703–3707, 2016.

[8] Nanne van Noord, Ella Hendriks, and Eric Postma. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32(4):46–54, 2015.