UCLouvain

École polytechnique de Louvain

Preserving Cultural Heritage with AI: Addressing Data Challenges for Museums and Memory Support

Author: **Victor Rijks**
Supervisors: **Benoît Macq, Sébastien Lugan**
Readers: **Eric Piette, Nicolas Bioul, Adrien Denis**
Academic year 2024–2025
Master [120] in Computer Science

Preserving Cultural Heritage with AI: Addressing Data Challenges for Museums and Memory Support

by Victor RIJKS

This Master's thesis explores the usage of Artificial Intelligence to improve the accessibility and promotion of digital art collections, specifically for the *Royal Museums of Fine Arts of Belgium* (RMFAB) and its activities with Alzheimer's patients. The current process of selecting potential memory triggering artworks is resource-intensive and highlights an opportunity for a modern, cutting-edge solution.

Our work addresses this by developing an AI-powered search engine. We tackled the challenges of working with a limited, inconsistent and specialized dataset. The core of our approach involved fine-tuning Contrastive Language-Image Pre-training (CLIP) models by overcoming data scarcity with multilingual synthetic data generation pipelines.

Through an iterative fine-tuning process, we developed the family of *art-* models (*art-mini*, *art-base* and *art-large*) and rigorously benchmarked their performance across representative queries on the *RMFAB* artworks, proper nouns alignment, and generalization capabilities in French, English, and Dutch. Our best models demonstrated significant improvements in multi-modal retrieval, achieving high accuracy in identifying relevant artworks based on textual prompts and showing promising generalization to unseen art styles.

The resulting search engine prototype is a comprehensive tool featuring advanced hard and soft querying capabilities, relevance feedback mechanisms, and creative collection management tools. Beyond search, we also explored the potential for AI-assisted iconographic term prediction, offering a semi-automated annotation solution for curators. This research provides a valuable blueprint for cultural institutions seeking to use the power of AI to unlock the richness of their collections, foster engagement, and support therapeutic applications like Reminiscence Therapy.

Acknowledgements

I would like to thank my promoters *Benoît Macq* and *Sébastien Lugan* for their insightful guidance and their valuable feedback throughout my master's thesis.

I would also like to thank *Nicolas Bioul* and *Adrien Denis* from *Openhub* for their feedback on the software part of this project, and *Lisa Quenon* for her insights on Alzheimer's disease.

Finally, I would like to thank *Isabel Vermote*, *Karine Lasaracina*, *Marc Van Oene* and *Marie-Suzanne Gilleman* from the *Royal Museums of Fine Arts of Belgium* for the interest and the time they have accorded to this project.

Contents

List of Figures

List of Tables

This work is dedicated to the people who always supported me, my mother Grazyna, my father Philippe, my sister Tatiana, my grandparents Babcia, Sonja, Dziadek and Jean-Claude, my girlfriend Camille, my friends Aimé, Alexandra, Grégoire, Guillaume, Konstantinos, Roland and Sebastian, and also to Max and Brando.

Chapter 1

Introduction

Recent advances in object detection, natural language processing (NLP), multi-modal models, and increased computational power have led to significant progress across various industries. These technologies are reshaping how we interact with information— from smart cars detecting their surroundings to AI assistants being able to understand and interact with textual and visual prompts. In healthcare, particularly for Alzheimer's patients, these advancements hold promise for developing tools called **Serious Games** that could improve therapeutic outcomes. And in the cultural field, AI applications have also begun to make an impact.

Museums and other cultural institutions play a crucial role in our societies. Art is a link to our past, and a pillar to stand on for the future. It has the unique ability to transcend time, triggering profound emotional responses and memories, particularly for individuals suffering from cognitive conditions such as Alzheimer's disease. Using familiar objects, like paintings, can help someone to reconnect with their past. The Royal Museum of Fine Arts of Belgium (*RMFAB*), which has kindly agreed to collaborate on this project, has been exploring ways to engage Alzheimer's patients through art. They hold activities where an art historian listens to the patient and selects the most suitable art piece to trigger memory recall. However, this process is resource-intensive and difficult to scale, highlighting the need for an automated solution that can efficiently assist in this context.

In this thesis we will explore how *AI* and its surrounding technologies can be used to help a museum to promote its digital gallery, particularly to groups composed of Alzheimer's patients. This work presents a workflow allowing a museum (or any other cultural institution) to accessibly train and deploy a search engine powered by *AI*. The objectives of this thesis are to :

1. Investigate ways to better promote art through *AI*
2. Create a new and smart dataset from the data given by the *RMFAB*
3. Develop *AI*-powered tools that helps Alzheimer's patients trigger memories by identifying the most relevant art pieces from the museum's collection.

Chapter 2

Context of the project

This project is done in partnership with the *Royal Museums of Fine Arts of Belgium* (*RMFAB*) with the final goal of producing a viable prototype that could be tested or even implemented in their activities. Therefore, it is necessary to correctly identify the needs of the museum and the requirements that a tool made for Alzheimer's patients must respect. This section explains the current activity done at the museum, the nature of our partnership and the data available.

2.1 The activity currently done at the museum

Currently, the *Musée sur Mesure* [32] team at the *RMFAB* offers tailored activities for specific groups. They organize activities in two stages. The first stage consists of meeting the patients often at their caregiving facilities to discuss with them. This first step allows the curators of the museum to find which artwork they could show the patients when they come to the museum. From the discussions we had with the museum staff, we could not identify the exact criteria used to select the artworks. From our understanding, there isn't a specific rule-set that guides this selection, rather, the selection is made to establish links with the discussion with the patients.

The second stage happens directly as the museum, the group of patients are guided through the museum by the guides. They will be presented relevant artworks that will hopefully trigger discussions among the group. These activities spark interest and offer an invaluable bridge to the artworks owned by the museum.

2.2 The *RMFAB* data

The core data of this project resides in the online library of the *RMFAB* named *Fabritius*. This tool offers the possibility to browse the museum's catalogue, with advanced searching filters like filtering by period, by style, by author, etc. *Fabritius* is available in French, Dutch and English. From our analysis, the French dataset is the most complete, followed by the Dutch and finally the English dataset. The figure 2.1 presents two screenshots of the *Fabritius* search engine.

(A) Advanced searching form

(B) Results page for the query *"Une maison"*

FIGURE 2.1: Screenshots of the *Fabritius* search engine

2.2.1 What is the data

This section will cover the nature of the data given to us by the *RMFAB*. We had access to 6869 artworks metadata in French, for most artworks we had access to a low resolution and a high resolution digital image of the artwork. This raw dataset was filtered and some entries were removed due to copyright issues or missing images. The resulting filtered dataset contains:

1. **5301 artworks**
   With:
   (a) Unique identifier (recordID)
   (b) Artist
   (c) Title
   (d) Type of work
   (e) Materials used
   (f) Signature information
   (g) Date of creation information
   (h) Dimensions
   ...
2. **870 artists**
   With:

(a) Unique identifier (creatorID)
(b) First name
(c) Last name
(d) Nationality
(e) Birth date
(f) Death date
...

3. **516 (**9.73%**) Textual interpretation of the iconography**
   For example: *C'est avec émotion et une rare force expressive, que Redon a représenté le Christ en homme de douleur. (d'après Brita Velghe, in 'Musée d'Art Moderne. Oeuvres choisies')*

4. **123 (**2.32%**) Textual description of the image (caption)**
   For example: *Sept femmes portant chacune une raquette de tennis, sur un terrain couvert d'herbe.*

5. **125 (**2.36%**) Textual identification of the subjects present in the image**
   For example: *Sept fois la soeur du peintre, en pied, dans sept tenues différentes.*

6. **3610 (**68.10%**) Subject matter objects present in the image**
   For example: *figure : femme ; arbre ; table ; chaise*

7. **2817 (**53.14%**) Subject matter of the iconographies present in the image**
   For example: *scène ; bourgeoisie ; méditation ; musique*

8. **670 (**12.64%**) Subject matter of the concepts present in the image**
   For example: *hybride fabuleux : sphinx*

The latter three categories are available in two formats. In the examples given above, the format used is the flattened format, this format is a list of strings. The other one is a tree-like format that respects the structure defined by the *Thesaurus Garnier*.

Thesaurus Garnier

Many museums (and similar institutions) use a hierarchical and structured iconography format with limited classes to describe their artworks. The most used iconography format is named *Iconclass*. Interestingly, the *RMFAB* chose to use another iconography format in the 2000's. The format that they use is called the *Thesaurus Garnier*.

We will explain the basic format of the Thesaurus without listing all its classes and caveats as this is not required to understand the work we have done. If the reader is interested, the Thesaurus Garnier is described at length in the book *THESAURUS ICONOGRAPHIQUE système descriptif des représentations* by *François Garnier* available on the *culture.gouv.fr* website [15].

This format does not only store the objects, the people or the concepts present in the artwork, it also stores hierarchical relations between them. The relations are simple, they only describe a parent to child relation. A relation does not represent an action like ***man*** WORKING IN ***workshop***

would for example. The Thesaurus Garnier would only represent that previous example as: ***man*** → ***workshop***.

A *Subject Matter* entry is stored as a string with *iconographic terms* separated by delimiters. The rules are as follow:

- **;** Neighboring object
- **:** Child of the previous object
- **(** Start of a group
- **)** End of a group

As described in 2.2.1, there are **3** fields of the type *Subject Matter* that are using the *Thesaurus Garnier* format. The first *Subject Matter* field is called **Subject Terms**, it describes the objects present in the artwork. The second *Subject Matter* field is called **Iconographic Terms**, it describes the iconographies present in the artwork. The third and last *Subject Matter* field is called **Conceptual Terms**, it stores the concepts represented in the artwork.

Le port
Paul Bril

The painting *Le Port* 2.2.1 by *Paul Bril* has the following **Subject Terms**:

scène (bateau : caravelle ; pavillon ; homme ; travail) ; fond de paysage (baie ; rocher ; bord de mer ; port ; nuage ; montagne ; soleil ; effet de soleil ; mer)

This is very valuable to us since it offers a very structured and qualitative description of an artwork made by curators from the museum

Representing the iconography as a tree

By following the rules of the *Thesaurus Garnier*, we can represent a *Subject Matter* as a tree. If we transcribe the *Subject Matter* presented in 2.2.1, we get the tree presented in figure 2.2.

This step can easily be done by a parser. Establishing the rules of said parser is trivial but since the museum data contains some imperfections, we have to manually deal with these edge cases. This caveat is explained in the subsection 2.2.2.

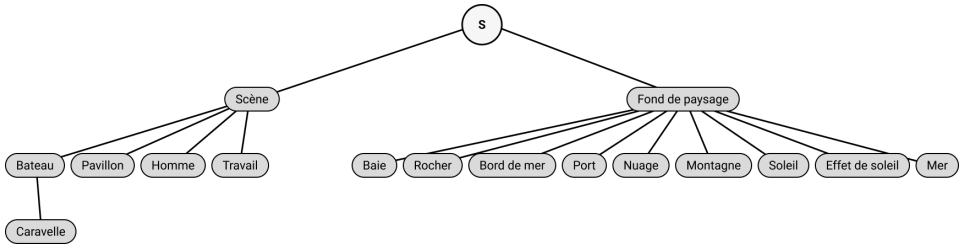


FIGURE 2.2: Subject matter of *Le Port* by **Paul Bril** represented as a tree

2.2.2 Issues

The *RMFAB* dataset is very useful as it is of the highest quality possible. The fields have been entered by museum's curators and we can have a high confidence in their content. Yet, we have noticed some issues with them.

Lack of data consistency

After the initial filtering, we managed to get **5301** artwork descriptions that can be used reliably for the next steps of this project. However, there is variability in the completeness of these descriptions. Indeed, the museum's curators added artworks throughout the years with varying amounts of fields filled. If we look at the percentage of columns filled (Figure 2.3), on average only **67.53%** of them contain a value. This is an important observation. *AI* models often require large amounts of data to train or finetune, we will have to take that lack of consistency into account.

This inconsistency varies depending on the sub-datasets that we have (*Artist information*, *Artwork information* and *Subject Matter information*). The *Artist information* sub-dataset is the most complete, we exactly know the artist's identity 83.31% of the time. Notably, we **always** have a description of the artist(s) denoted by the column *Description of the Artist*. One could potentially infer the exact artist identity based on this column. We have less information about the artwork itself, on average 77.38% of an artwork's entries are filled. The most sparsely annotated category by far is the *Subject Matter* category with only on average 24.72% of its columns filled. This is a disappointing observation as this sub-dataset contains entries that can be easily used for training a captioning model (particularly the column *General Subject Description* that is a high quality caption of the artwork).

Incorrect subject matter

The parser we have made to build the tree from the raw subject matter strings corrects two types of mistakes:

1. **Mistake 1:** Missing ; between two groups defined using parenthesis
   Example: *(paysage : arbre) (femme)* should be *(paysage : arbre) ; (femme)*
2. **Mistake 2:** Additional)
   Example= *homme)* should be *homme*

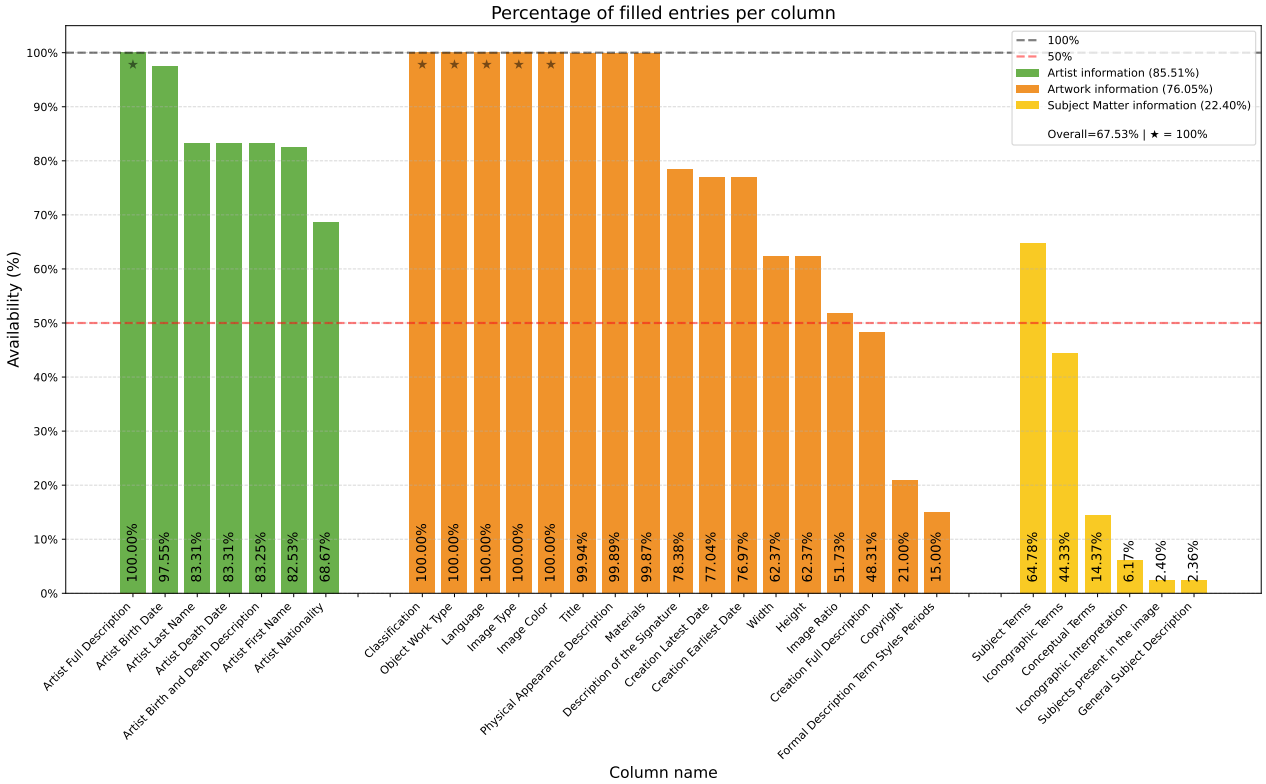


FIGURE 2.3: Availability per column

These two mistakes happen **very rarely** in the dataset. Out of the 7097 subject matter present in the dataset, only 10 contained a mistake. Yet we would argue that it is important to keep and fix these 10 rows as our dataset is already sparse.

Missing images

As explained in the beginning of this section, the dataset that was given to us contains 6869 descriptions of artworks. Out of those, 1551 did not contain either a low or either a high quality image. This left us with 5318 artworks. A final filter was applied on those image to remove the ones that were corrupted, this removed 17 images that would have been unusable.

High number of classes

The *Subject Matter* fields are very valuable. They describe the objects, the iconographies and the concepts present in a considerable portion of the artworks at our disposal. A *Subject Matter* tree can be flattened to get the list of the *iconographic terms* contained in an artwork. This is a first transformation of the data that could be easily used to browse the artworks of the digital collection. We could collect all the artworks containing a specific class or a specific set of classes.

By counting the number of unique classes (see table 2.1) we notice that there is a **substantial** amount of unique classes. In total, there are 3860 unique classes appearing in the *subject matter* fields. For the three *Subject Matter* variants, some classes appear more than once, this is particularly noticeable for the *Subject Terms* where the ratio between the number of unique classes and the number of entries is equal to 0.394. The other two ratios are to 1 with 0.913 for the *Iconographic Terms* and 0.949 for the *Conceptual Terms*. This difference in ratios is mainly explained by the specificity of these three variants. The *Subject Terms* variant focuses on objects, this variant therefore does not contain many proper nouns as we can see in the table 2.2 (less than 2% compared to $> 50\%$ for the other two variants).

Field	Number of unique values	Number of entries	Unique value ratio
Subject Terms	1422	3610	0.394
Iconographic Terms	2573	2817	0.913
Conceptual Terms	636	670	0.949
All subject matter fields	3860	7097	0.544

TABLE 2.1: Table presenting the number of unique classes per *subject matter* variants

Field	Number of capital letter words	Number of unique classes	Percentage
Subject Terms	26	1422	1.83%
Iconographic Terms	1713	2573	66.58%
Conceptual Terms	322	636	50.63%

TABLE 2.2: Table presenting the number of classes with at least one capital letter per *subject matter* field

The class distribution is heavily skewed. Most classes (86.5%) do not appear more than 10 times in the *Subject Matter* entries, *only* 521 classes appear frequently (more than 10 times). Proportionally, only 25 classes appearing in more than 5% of the *Subject Matter* entries ($= 266$ entries). These top 25 entries are detailed in the table 2.3. This uneven distribution allows to reduce the number of classes for techniques sensitive to contextual sparsity such as YOLO [44]. However, such an aggressive reduction might lead to the loss of crucial distinctions between artworks. If we reduce the number of classes too much, we might end up with the same set of classes for many or even most artworks rendering these keywords useless.

Nature of the images

By looking at the images extracted from *Fabritius*, we quickly noticed the high variance in the way they look. This is an issues often arising when working with artworks. A picture of a sculpture is obviously very different from a picture of a painting. And two paintings of the same subject can have very different styles; for example one could be a realistic painting while the other could be an impressionist painting.

Rank	Class	Count	Percentage		Rank	Class	Count	Percentage
1	homme	1861	35.11%		14	en buste	334	6.30%
2	figure	1436	27.09%		15	eau	327	6.17%
3	femme	1367	25.79%		16	en pied	324	6.11%
4	groupe de figures	972	18.34%		17	nu	312	5.89%
5	animal	822	15.51%		18	chien	309	5.83%
6	paysage	663	12.51%		19	architecture	287	5.41%
7	portrait	660	12.45%		20	intérieur	285	5.38%
8	scène	641	12.09%		21	maison	282	5.32%
9	arbre	633	11.94%		22	couvre-chef	280	5.28%
10	enfant	543	10.24%		23	[SO]	279	5.26%
11	vêtement	428	8.07%		24	ville	277	5.23%
12	assis	395	7.45%		25	cheval	267	5.04%
13	chapeau	372	7.02%					

TABLE 2.3: Table presenting the 25 classes appearing in more than 5% of the *subject matter* entries (more than 266 entries)

The column *Object Work Type* in the *RMFAB* dataset gives us a hint as to the nature of the image. We **always** have a value for this column (see figure 2.3). In a subset of the total dataset [1], there are 34 unique values for this column (see table A.1). This high number of unique values makes it difficult to create a smaller representative set as some unique classes would get only few images.

The solution we used to reduce this high number of unique classes is to use the explicit *Object Work Type* values (24 out of the 34 unique values) to cluster the artworks in three subgroups. For example, the *Object Work Type tableau (toile)* is obviously a *Tableau*. The object work type *statuette* is obviously a *Sculpture*. The explicit *Object Work Type* defined for the three subgroups can be found in the table D.1.

To cluster the 3012 artworks into our three subgroups, we first one-hot encode them. This gave us 3012 vectors of dimension 34. Then we can compute the cluster's centroid by applying the algorithm described in the pseudo-code 23. By running this algorithm on the subset, we get the centroids plotted on the figure B.1. We can cluster the artworks by assigning them to the closest centroid. A sample of the resulting clustering ordered by the confidence from lowest to highest (using a softmax) can be seen in figure 2.4. Of course, a generic centroid finder algorithm could have been used.

[1] We did not use the full dataset for this analysis because we initially chose to restrict our dataset more. Following the initial results that will be discussed later, we decided to extend it to have more artworks available. This subset still contains more than 3012 artworks and its results are representative and used when creating the validation and testing sets.

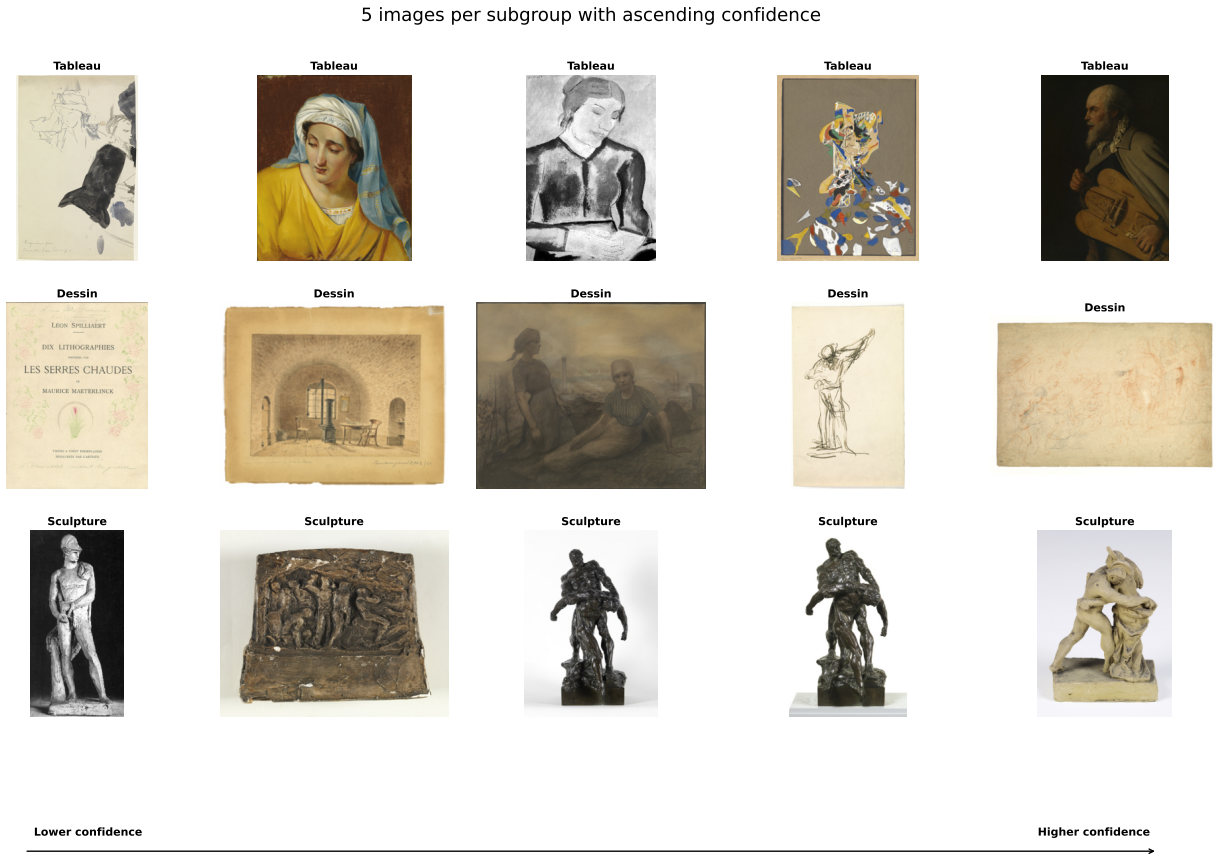


FIGURE 2.4: Sample of the assignements made by the algorithm 23 on the February subset of the *RMFAB* dataset ordered by lowest to highest confidence using a softmax

Accessibility to the data

This final point, though specific to this project, highlights a common challenge when working with data from cultural institutions. Late access to datasets, often due to outdated data management infrastructure, necessitates significant initial effort in processing, cleaning, and restructuring the information into a modern and practical format. We would advice anyone taking this work as an inspiration to take the time to properly do this step as it greatly facilitates the training and benchmarking of the models.

Chapter 3

Theoretical background

3.1 Usage of computer science to make therapeutic games (Serious Games)

A few studies have been made on the utilization of computer games as a therapeutic tool for Alzheimer's disease (AD) patients. These tools have garnered interest as a low-cost and non invasive option with the potential to enhance cognitive function, emotional well-being and social engagement. It has been shown that well-designed serious games (SG) can create an enjoyable and supportive environment for individuals with reduced cognitive functions.

For instance, Weismen (1983) [54] highlighted that video games can provide a way for frail elderly individuals to improve self-esteem and facilitate social interactions. Another example of a well-designed tool would be the Butler system, a simple operation system tailored for elderly users that has demonstrated a potential to improve their mood and reduce the feelings of sadness and anxiety (Castilla et al., 2013) [6]. Serious game and other computer-related applications, when well-designed, can provide significant emotional benefits.

Besides emotional well-being, SG have also been used as an accessible tool to train a dementia patient's cognitive capabilities. Tong, Chan and Chignell (2017) [50] presented several games, each aiming at training a specific capability like *short-term memory* for example. This ability of remembering newly acquired information can be trained using a simple card Memory game for example. Another ability, *long-term memory* can be trained using reminiscence games, which uses a variant of trivia to train the retention of old information (Tong et al., 2017) [50].

These approaches align with studies that have discovered correlations between SG performance and commonly used cognitive assessments like the MMSE and the MoCA. This offers the potential to use SG as a way to assess a patient's cognitive decline (Imbeault et al., 2011) [20]. Moreover, an interactive multimedia tools called Smartbrain ([40]) improved the results of classic cognitive training programs like IPP (integrated psychostimulation program) for example (Tárraga et al., 2006) [48].

But to fully take advantage of the potential of serious games, one must understand the key designing choices needed. Indeed, when making a SG aimed for AD patients (and dementia patients in a broader view), we must carefully consider the usability, the engagement and the accessibility offered. Studies have emphasized the importance of simple interfaces, touch-screen technologies (or other very accessible input methods) and adjustable difficulty (Pyae et al., 2016 [41]; Bouchard

et al., 2012 [3]). This prevents making the patient frustrated because he does not understand what's being asked of him, or because he finds the game too easy. For the latter insight, a study by Tremblay et al (2012) [21] explored the idea of using dynamic difficulty adjustment (DDA) to automatically decrease or increase the difficulty without asking a patient that has no idea as to which difficulty suits him most to select it.

3.2 Reminiscence Therapy (RT)

Reminiscence Therapy or RT for short is a non-pharmacological therapy for dementia that boils down to recalling memories from the past. The idea is to try to unleash these forgotten memories by presenting old pictures, familiar sounds/music or by discussing with the patient. This type of therapy presents many advantages: it is non-pharmacological, it can be used with varying levels of cognition [14] and it reduces social isolation notably by providing a way to keep in touch with loved ones [53].

Although it should be noted that even if several reviews have been published praising this approach, the area lacks a large study with a sizeable sample size [27]. RT is a promising type of therapy that can easily be offered to the patients. It offers an enjoyable experience for the patient and the caregivers.

The wide availability of computers and phones and the digitization effort undertaken by the *RMFAB* could be catalysts for such a therapy. RT has been provided remotely [17] [18] [26]. But we must still keep in mind that remotely delivering the artworks from the museum in the context of a remote RT session requires that we design the application to be easy to use and intuitive. The remote RT systems discussed in [27] used simple interfaces where the patient was passively receiving the therapy. The therapist was remotely updating the content shown to the patient.

A very recent tool made by *Polaroid* in partnership with the *Fondation Recherche Alzheimer*, *Alzheimer Belgique* and *Baluchon Alzheimer* called *Memory Shots* [25] uses image generation models to generate realistic pictures (in the style of *Kodak* cameras) of memories written by the user. This is a unique and impressive approach that tries to **create** visual memory triggers that can be more evocative than a simpler textual description.

3.3 Usage of computer science to promote culture

The current tools used by the *RMFAB* are already computerised. This is consistent with the times in which we live in. Their current platform *Fabritius* allows the user to navigate through their digital gallery, querying by title or period, for instance. It offers an online access to culture. The RMFAB is not the only museum to use AI or other technologies to provide and analyse its digital gallery. For example, the Harvard Art Museums offers an AI Explorer. [31] allowing the user to explore their digital gallery using objects detections performed by various SOTA detection models. A similar tool called *SMKExplore*, powered by the zero-shot detection model *GLIP*, has been developed by a team from *IT at the University of Copenhagen*. It allows users to explore the digital gallery at the *National Gallery of Denmark (SMK)* by browsing thematic groups of artworks [30]. This approach of using vision models to detect objects on artworks has become possible with the recent improvements in vision models, it offers the possibility to define keywords for the thousands of artworks lacking them [10].

Large Language Models (LLMs) could also use their extensive knowledge to add context around an artwork, a very recent study, Reusens et al (2025b) [45] explored this idea by using LLMs to find relevant keywords and contextual information that people could use to search for historical objects. This approach enriches the keywords associated with a document, thus making it easier to find.

But AI could also help museums in the physical world. For example, *Pei et al. (2024)* [39] presented a case study in which, in partnership with a Beijing-based museum design company, they used an LLM and an image generation model to refine ideas for future exhibitions by generating visual mock-ups for them

However, it is important to recognize the limitations of using large pre-trained models. Vision models suffer from the *cross-depiction problem* as they are almost exclusively trained on photographs, which makes their performance on images with high style variance such as artworks less predictable [4]. In addition, initial studies of these detection models encountered gender and Western cultural biases [10]. On top of this, the limited contrast of the models [10] and the lower resolution at which they operate can miss small but important details.

3.4 Exploratory Search

Exploratory search (ES) is a specific vision of searching. It is characterized by its open-ended and iterative nature. Unlike traditional lookup search, where a user knows what he wants to see and he expects a precise answer, ES is performed whenever a user wants to discover a domain, increase his knowledge or learn about new topics [38].

In Bates (1989) [1], the author describes the metaphor of *Berrypicking*, the user's information need and search query evolve and shift as they interact with and discover new information. Their idea of a right answer is therefore not static. It is an iterative and dynamic process, akin to picking berries in a forest where one constantly adjusts their path based on what they find. This contrasts sharply with traditional "one-shot" retrieval models and highlights the continuous reformulation of

queries and information needs during ES. This vision of searching requires that the searching tool allows for such an exploration [33]. This can be done using recommendations based on the result the user clicked on for example.

Evaluating these exploratory search systems presents challenges, as it requires a qualitative and quantitative analysis of both the search results and the user behaviours [38]. Palagi et al. (2015) [37] proposed to take into account the concept of *surprise* beyond the classical retrieval metrics. A result may be considered *interesting* if the user perceives it as similar to the topic he is trying to explore or/and it may be considered *surprising* if its relation to the explored topic is unexpected [37]. In any case, evaluating this kind of systems is by definition difficult as defining the *best* results if often subjective to a certain degree.

3.4.1 Image-Text vectorization: *CLIP*

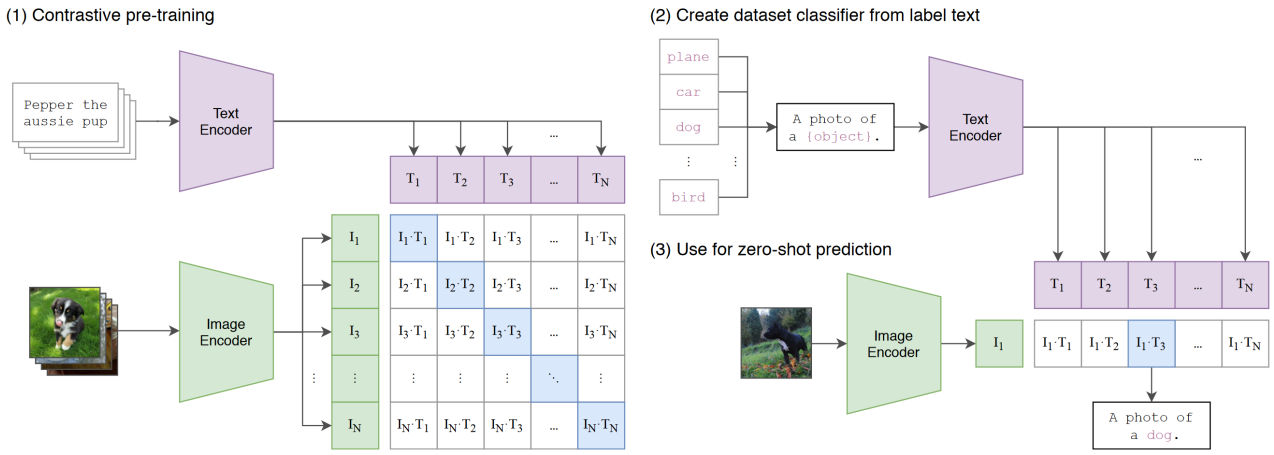


FIGURE 3.1: Summary of the *CLIP* approach taken from [42] (fig. 1, page 2)

Unlike classical visual models that extract features from images to predict specific labels, *CLIP* operates by jointly training an **image encoder** and a **textual encoder** [42]. The core idea is to learn a shared embedding space where the embedding of an image and the embedding of its respective captions are close while incorrect pairings are not.

During training, CLIP is presented with batches of image-caption pairs. For each batch, the objective is to predict which caption correctly corresponds to which image. This is achieved by maximizing the cosine similarity between the embeddings of the N correct image-captions pairs while minimizing the cosine similarity for the $N^2 - N$ incorrect pairs [42]. This **contrastive learning** approach allows the model to learn a robust representation of image features and linguistic concepts.

A key observed advantage of this approach is its ability to perform **zero-shot transfer** [42]. The classes we predict for can be chosen at test time with impressive performance [42]. This is done by

creating textual embeddings for descriptions like "*A photo of a **cat***" and "*A photo of a **dog***" and then picking the textual embeddings that maximize the cosine similarity with the image embedding we want to predict for. This flexibility comes from the generality of natural language, which can express a vast range of concepts [42].

Yet, there are some observed limitations, one common issue is **polysemy**, where a word can have multiple meanings [42]. In the original *CLIP* paper by Radford et al. (2021), an example was given with the word "*boxer*" that could refer to a dog breed or an athlete. Without additional context, *CLIP*'s textual encoder may struggle to differentiate the intended meaning [42]. To mitigate this, the authors proposed prompt engineering methods like using a prompt like "A photo of a *label*". Furthermore, while CLIP demonstrates greater robustness to distribution shifts compared to standard ImageNet models which is very interesting for artworks that are by nature often out of the distribution of photographs, it can still struggle with truly out-of-distribution data [42]. This is bad news for very out of distribution artworks like abstract art for example. Its performance can be weak on fine-grained classification tasks, abstract tasks like counting objects, and as expected novel tasks not represented in its pre-training data [42].

CLIP has already been explored in the artistic world, for example, *Conde and Turgutlu (2021)* [11] adapted CLIP for fine-grained art recognition, creating a model called *CLIPart*. Their approach involved generating free-form text descriptions from categorical annotations from the *iMet dataset* [7] then fine-tuning a *CLIP* model on it. This domain adaptation demonstrated better results than few-shot supervised state-of-the-art models for fine-grained art classification and image-text retrieval [11].

3.5 Vision Language Models (VLMs)

Vision Language Models or VLMs for short are a type of models capable of processing both image and textual inputs to produce textual outputs [52]. These large-scale models exhibit strong zero-shot capabilities [52]. The development of models like *CLIP* has notably contributed substantially to the recent progresses within VLM studies [57]. VLMs can be used in several ways. They can be used to chat about the provided image, to generate captions or to answer questions about the image [52].

VLMs can also be used to distill their general knowledge into models that are better suited for specific tasks [57]. For instance, this technique allows the transfer of the general detection capabilities of a large VLM model into a detection model having an architecture better suited for object detection by generating a detection training set [57].

Chapter 4

Approach and Methodology

4.1 Choosing a model

The choice of the model or the models that would be used in this thesis was very influenced mainly by the size and the content of the *RMFAB* dataset and by time and the resources available to me. We decided to go with a single monolithic model that would link the visual and the textual space. The model chosen was *CLIP*. Using a single model avoids the difficulty of weighting the various models' outputs to combine them into a single retrieval output. It also reduces the training resources necessary for its finetuning and the complexity of its implementation. Using a single model makes this work more easily reproducible by other institutions and more practically maintainable by the *RMFAB* staff.

4.2 CLIP performance out of the box

As seen in figure 4.1, *CLIP* is already a fairly powerful model out of the box on artworks. The very large training dataset (400 millions of pairs) that *CLIP* was trained on allows it to perform remarkably well in zero-shot environment. The English top 1 retrieval are particularly good, this is certainly due to the fact that *CLIP*'s training set is mostly in English. In the few samples shown, French seems to perform better than Dutch. The model used for this experiment is *CLIP L/14*.

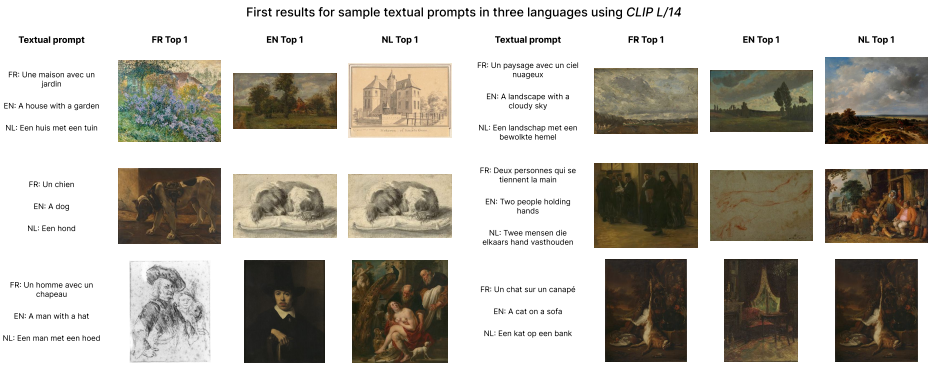


FIGURE 4.1: A few examples of what *CLIP L/14* retrieves on the *RMFAB* dataset

4.3 Finding the right format

As explained in the previous sections. With a very limited dataset in French, we would like to make *AI*-powered tools or games to assist the museum curators in making their activities, notably their activities for Alzheimer's patients.

The choices guiding this project will take this context into account. Our objective is to provide the *RMFAB* with tools or games that could assist them during their activities. The nature of these tools and games is not evident. As supported by the findings of the *Theoretical Background* 3 chapter, we could develop a *Serious Game* using the artworks from the museums or we could develop a exploration engine similar to *SMKExplorer* that aids the curators from the museum to prepare the activities at the *Musée sur Mesure*. To decide which path would help the *RMFAB* the most, we decided to develop a few prototypes in concertation with the *RMFAB*.

4.3.1 Finetuning the first model

Before explaining the prototypes that were presented in early February 2025 to the *RMFAB* staff, it is important to at least glance over the finetuning process that was used to obtain the first *CLIP* model. This naive pipeline lead to the more intricate pipelines that will be described later in this paper.

As explained in the analysis of the *RMFAB* dataset that was provided to us, the nature and the size of the dataset we were to work with was very limited. Making our own dataset by captioning the artworks manually was not feasible because of the limited time and resources that we had. Moreover, initial finetuning metrics using this approach resulted in a model that would overfit quite fast, most certainly due to the limited number of captions[1]. Our task became therefore to find a technique allowing us to generate a coherent dataset that could improve significantly the performance of *CLIP* on linking the textual world and the world of digitalized artworks, specifically artworks from the *RMFAB*'s gallery. The full details surrounding the training of the models used in the prototypes can be found in section 4.5.2.

4.3.2 Initial prototypes

We developed 3 prototypes to get the feedback from the *RMFAB* staff, especially from the people preparing and providing the activities for the Alzheimer's patients. This is a necessary step to ensure that this project can be used in practice.

Guess the Artwork !

This prototype is heavily based on the famous board game **Guess Who** [12]. This deduction board game has two participants playing against each other where both aim to identify their opponent's randomly selected character from a set of 24 distinct faces by posing a series of yes-or-no questions, usually regarding physical attributes. A player iteratively eliminates wrong candidates until he is ready to guess the other's card.

[1] We can only guess the reasons behind this observation, we believe that this approach could work only if **many** captions were manually made which is not practical.

We adapted this board game to use artworks taken from the *RMFAB* digital gallery instead of faces. The player is presented with 10 artworks from which he chooses his secret card. The *AI* playing against the player will also choose a secret card from the same set of artworks. As the game progresses, the player receives hints from the *AI* and the *AI* asks questions that the player can answer with 5 degrees of certitude: from a strong **NO** to a confident **YES**[2]. The figure 4.2 presents some screenshots from this prototype.

This game focuses on stimulating the *short term memory* of the players. Indeed, as hints pile on, the player has to memorize not only the cards he eliminated but the hints he received. As previous studies have found, *Serious Games* should implement adjustable difficulties to tailor the experience to the player (Pyae et al., 2016 [41]; Bouchard et al., 2012 [3]). This idea has been implemented in *Guess The Artwork !* by offering three levels of difficulty to the player:

1. **Easy:** The player receives 10 hints from the *AI* whereas the *AI* can only ask 5 questions
2. **Medium:** The player receives 7 hints from the *AI* whereas the *AI* can ask 7 questions
3. **Hard:** The player receives only 4 hints from the *AI* whereas the *AI* can ask 10 questions

As the difficulty progresses and becomes more challenging, the player receives fewer hints, while the *AI* asks more questions. Furthermore, the hints provided will be less helpful, and the *AI*'s questions will become more insightful. The quality of information is determined by a heuristic designed to best distinguish the 10 artworks.

A hint given to a player is a **verified** information. When first developing this part of the game, we implemented a version that **infers** hints from the image (i.e. an *AI* model looks at the image and finds some hints). This proved to be a bad idea as giving only a single wrong hint to the player would totally mess his perception of the secret card that the *AI* chose. For example, if the *AI* confuses a lamppost for a man, giving it as a hint to the player could eliminate the *AI*'s secret card, making the game frustrating. This is a key insight of this prototype, **the uncertain nature of artificial intelligence responses makes it difficult to use them in a SG scenario**. To fix this issue, we decided on using the *Subject Matter* fields from the *RMFAB* dataset. This field has been written by museum's curators and we can be highly confident in its veracity. The *AI* will use objects from this list as hints.

A question asked by the *AI* takes the form of *Does your secret card contain this* ***OBJECT*** *?*. Most often, the player has to look at his selected artwork in detail, this is a desirable indirect effect of the game's design as it increases the time the patient spends looking at artworks, this is one of the objectives of the *RMFAB*.

[2]The exact degrees are:

1. Non (No)
2. Pas Vraiment (Not really)
3. Je ne sais pas (I don't know)
4. Plus au moins (More or less)
5. Oui (Yes)

The heuristic ordering the hints given and the questions asked works as follows. Let L be the list of all unique *iconographic terms* present in the 10 artworks. This list is composed of the objects, subjects or iconographies identified by the museum's curators on the artworks. In other words, it is composed of the concatenation of the flattened *Subject Matter* fields. The length of the list L is defined as N. Let L' be the CLIP embeddings of the N *iconographic term* in list L. Let A_i with $i \in [1:10]$ be the *CLIP* embeddings of the 10 artworks. We can compute a matrix of size $(N, 10)$ where each entry (n, i) is the cosine similarity between the *iconographic term* n (embedding L'_n) and the artwork i (embedding A_i). Using this matrix, the heuristics sorts the N keywords by the variance of its row in a descending order. This heuristic takes advantage of the supposition that a good keyword is a keyword that separates the artworks a lot. This approach also presents an easy way to advantage or disadvantage the player. Indeed, by adding noise to the cosine similarities, we can blur the ordering of the keywords. The stronger the noise, the more likely a bad *iconographic term* is sorted first (opposite: a good *iconographic term* is sorted last).

When presenting this prototype to the *RMFAB*'s staff, the overall feedback was positive. The game is stimulating and presents potential. Sadly, even with a reduced number of cards, a very simple UI and adjustable difficulty, the game was still too difficult. The *RMFAB*'s staff feared that it would be too hard to explain to some patients.

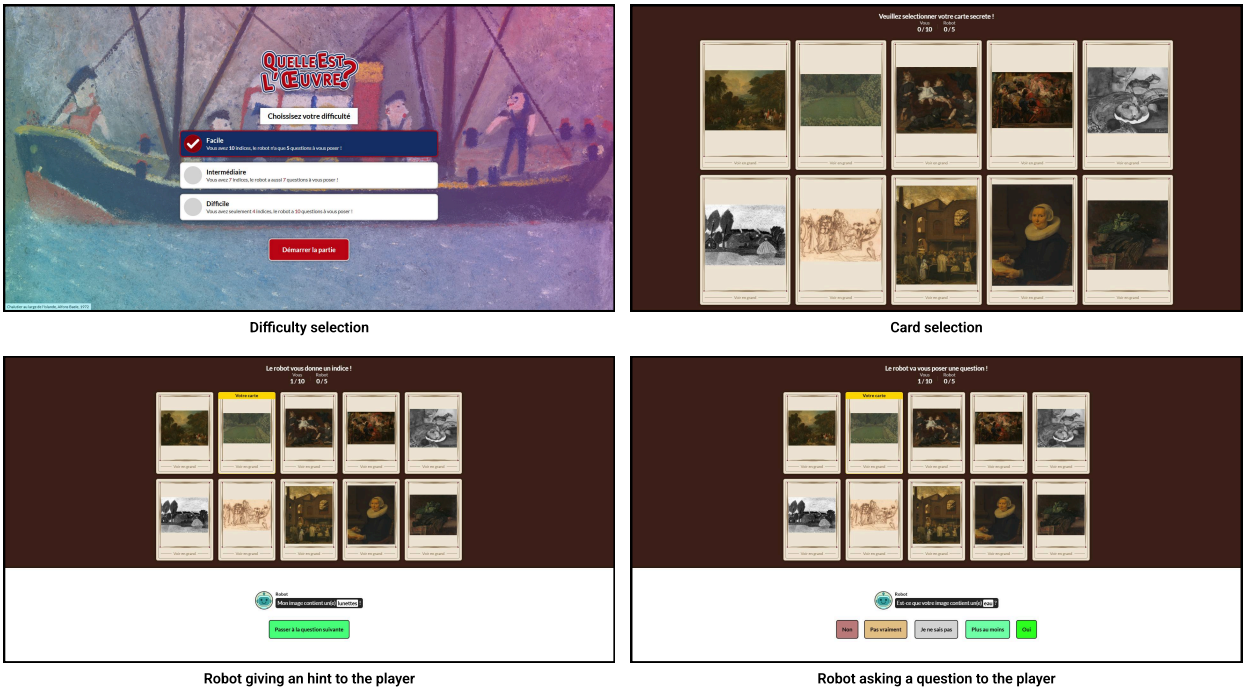


FIGURE 4.2: Screenshots of the *Guess the Artwork !* prototype

Journey generator

This prototype was meant as an exploration tool. The user is presented with a very simple interface containing 5 artworks on a black background and a simple prompt: "Please choose one artwork". Once the player makes this first choice, he is prompted with 5 different artworks from which he needs to choose. After choosing a starting and an ending artwork, a journey is generated.

The generation works by first embedding the two artworks chosen using the *CLIP* model. This representation allows us to generate a line connecting the two in the *CLIP* embedding space. We can place N points along this line at an equal interval. Each interval point will give us an *interval artwork* by finding the closest artwork (by cosine similarity) to this interval point. Once the generation has finished, the interface presents a sideshow of the artworks found to the user. This presentation has three objectives:

1. Make the user explore the artworks from the collection
2. Trigger memory retrieval
3. Tell a story using the artworks

The efficacy of this method was mixed. We observed instances where the results were illustrative and really told a story, we also observed instances that were hard to interpret. When presenting this prototype to the museum's curators, they showed interest in the journey generation possibly but the game was too abstract and raw.

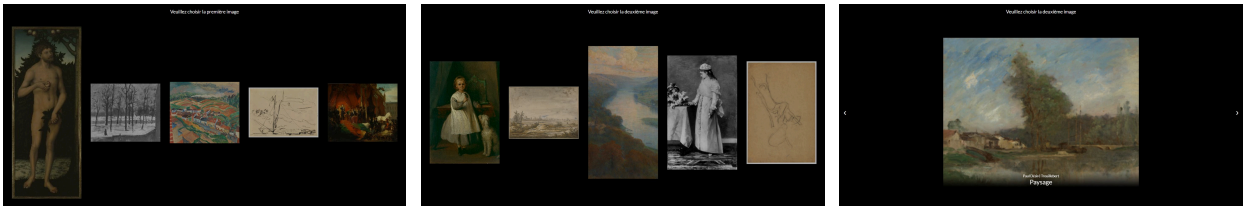


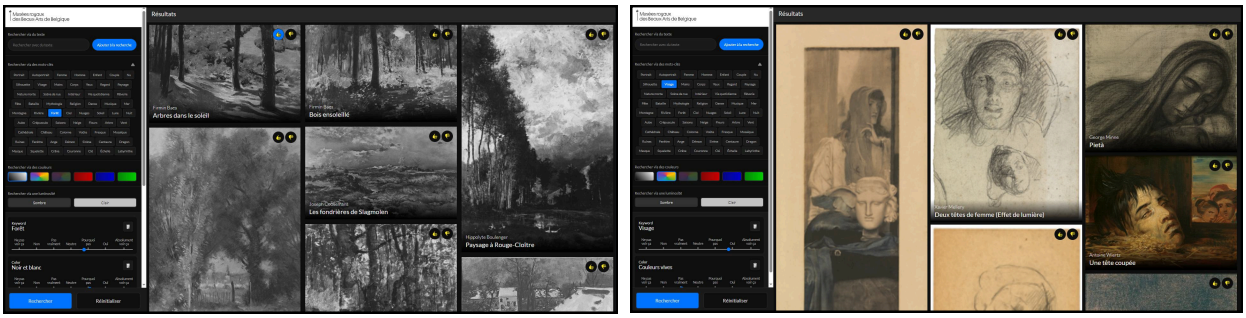
FIGURE 4.3: Screenshots of the *Journey generator* prototype

Search engine

The final prototype that we presented to the *RMFAB* in early February 2025 was a search engine that would allow the user to search with a textual prompt the database of artworks digitalized by the museum. This task fits totally in the scope of *CLIP*. By embedding both the artworks and the textual prompts submitted by the user, the back-end of this prototype could sort the results by returning the ones with the smallest cosine similarity with the prompt. The figure 4.4 presents some screenshots of this prototype.

An issue that quickly arose with the usage of *CLIP* was the limited number of tokens that the textual encoder could process at once. A textual query could contain a maximum of 76 tokens (with the $[SOS]$ and $[EOS]$ tokens) [42]. To circumvent that limitation, I decided on allowing the user to mix various textual prompts together. This approach allows the user to add new information to a previous query by adding a new term, it also allows the user to modify the importance of added queries by modifying their weights. It also allows to search for the opposite of a certain term by using a negative weight. The reader can find a few examples comparing prompts made in a single query versus prompts made using multiple queries. Another interesting feature that was made available in this prototype was the ability to *like* or *dislike* an artwork. Liking an artwork would mix the query with the *CLIP* embedding of that artwork, disliking an artwork would mix the inverse of the *CLIP* embedding of that artwork.

The composition of the features made the search engine very powerful and the museum staff was clearly interested in it. This prototype was a game-changer compared to their current tool (*Fabritius*) and they directly envision using it internally and potentially in their activities. During the demonstration, one of the curator of the *RMFAB* asked us to query with *Toison D'or*, this yielded incoherent results. We should take the proper nouns into account when finetuning the *CLIP* model. This behaviour is expected as the model used for this prototype **did not use** any of the *subject matter* fields.



Querying for: *forest* AND *black and white*

Querying for: *face* AND disliking the term *bright colors*

FIGURE 4.4: Screenshots of the *Search Engine* prototype

4.3.3 Focusing on the Search Engine

Although the *RMFAB* liked all prototypes and found them of interest, we decided to fully focus on the **Search Engine**. We chose to focus on this tool because it showed the most potential and because the *RMFAB* would directly benefit from it. The remaining of this thesis will therefore explore ways to improve the *CLIP* model on the artworks of the museum and it will also explain the development of the Search Engine interface.

4.4 Finetuning context

4.4.1 Performance measurements

Before improving the performance of the backbone *CLIP* model, it is necessary to define rigorous comparison benchmark to gauge the performance of the candidate models in specific areas. In the prototypes that have presented in earlier sections of this paper, the user could interact with the painting by using a textual prompt or a proxy to a textual prompt.

As discussed previously, the *RMFAB* staff envisioned two use cases for the search engine. The first use case would be an internal use of the engine, the curators and researchers from the museum could replace *Fabritius* with this new search engine. It is therefore important that we at least offer the same set of features that *Fabritius* currently offers. Our search engine should allow to search artworks using *hard* filters, for example: "*Get all the artworks created between* 1898 *and* 1910". The second use case would be an external use of the engine, visitors could one day use this search engine to explore the digital gallery of the *RMFAB*. We should provide a way to browse and search through the artworks with a simple an intuitive textual prompt. For example, the user could search "*painting with two people sitting on a bench*" and he should receive artworks representing that concept. The latter use case is very beneficial to the first one. Indeed, when discussing with the staff from the *RMFAB*, a common issue when using *Fabritius* was the difficulty to find a relevant artwork when not knowing **exactly** the title or specific keywords. Interestingly, a user could already use the flattened *subject matter* fields by entering an term he wanted to see in *Fabritius*.

Choosing the areas of interest

To construct our benchmarks, we should first answer the question *"What should a good search engine for artworks focus on ?"*. This is not a trivial question as it is somewhat subjective. The research around *CLIP* lacks a common benchmarking procedure but some propositions have been made to fill that void [8] [29] [13]. An issue with retrieval metrics is that they can vary significantly depending on the size of the candidate pool. Ideally, a model should produce the same metrics even when increasing the number of candidates, this would mean that the model clearly spots the best candidate and separates it well from the others in its representation space. This is not realistic as the *best candidate* is often subjective. Therefore, this thesis will use the metrics obtained **only** as a comparison tool between the models.

We decided to build our own benchmarks for this project because our model has the aim to work well on the *RMFAB* data primarily. Our goal is to improve a *CLIP* model so that it allows the

search engine to align better the textual prompts to the artworks of the *RMFAB* digital gallery. We also have two secondary goals. First, we would like to align the artworks with the proper nouns they are linked to as asked by the *RMFAB*'s curators. Secondly, we would be to make the finetuned models work well on Art images in general, this is desirable as the *RMFAB* often add new artworks in their digital gallery.

Quantitative analysis The baseline models finetuned by *OpenAI* [35] and the models that we have finetune during the course of this thesis will be tested on three benchmarks detailed in the next sections. The chosen metrics used to gauge the performance of the various models are:

1. **Average position**:
   The average position of the artwork for which the manual caption was written
2. **MRR**:
   The reciprocal rank of a query response is 1 divided by the rank of the first correct answer
3. **Recall@1, @3, @5 and @10**:
   A measure of the proportion of relevant items that are successfully retrieved within the top k results.
4. **nDCG@3, @5 and @10**:
   A stricter metric than Recall@k as it also takes into account the graded relevance of items by penalizing relevant items appearing lower in the results

Qualitative analysis In the chapter *Qualitative Analysis* 5, we have provided the reader with retrieval comparisons of the various models finetuned during the course of this thesis. This qualitative analysis is needed as retrieving the best artwork when given a prompt is by nature partly subjective,.

The *CLIP* models will be compared using three types of benchmarks: measuring how well sample queries on the *RMFAB* artworks perform, measuring how well proper nouns align with the artworks representing them and measuring how well the model generalize to artworks it has never seen.

Queries on the *RMFAB* dataset This benchmark measures how well a model performs on representative queries that a user could make on the search engine. We manually made 454 captions on 454 artworks from the *RMFAB* digital gallery. The artworks are selected randomly while still preserving the same *subgroup* distribution[3] (*subgroups* are defined in the section 2.2.2).

[3]The percentage of the subgroup *Sculpture* was slightly increased so that a representative set would not get a single sculpture image

Each caption has three additional information:

1. **Colour:** This field can be empty or it could contain one or more colors
2. **Luminosity:** This field can have the value *Dark*, *Neutral* or *Bright*
3. **Emotion:** This field can be empty or it could contain one or more emotions

A screenshot of the custom made captioning tool is presented in figure 4.6. The additional fields allows us to measure the performance of the model in specific areas.

Using this method, 1816 *tasks* where obtained. A *task* is simply an artwork, the area this tasks focuses on, a textual caption of this artwork and the additional information on this artworks required by the provided focus. This benchmark then splits into two variants. The first variant that is identifiable with the suffix *-PROMPT* in the tables of this thesis merges the caption and the additional information naively by joining them into a single string using commas. The second variant identifiable with the suffix *-MIXED* mixes the caption and the additional information by taking the mean of their respective *CLIP* embedding[4]. Using these two variants helped us to study the impact of prompting the model naively compared to prompting it by mixing several embeddings. An example *task* can be found in the figure 4.5.

Lucrèce

Guido Reni

Captions

Une femme avec un sein visible et une longue robe regarde vers le haut avec un dague en main, défense, agression

A woman with a visible breast and a long dress looks up with a dagger in her hand, defense, aggression

NL Een vrouw met een zichtbare borst en een lange jurk kijkt omhoog met een dolk in haar hand, verdediging, agressie

Additional information

Color	Luminosity	Emotions
Couleur noir et blanc	Luminosité lumineuse	Émotion peur, Émotion colère, Émotion surprise
Black and white color	Bright luminosity	Fear emotion, Anger emotion, Surprise emotion
NL Zwart-wit kleur	NL Heldere helderheid	NL Angst emotie, Boosheid emotie, Verrast emotie

FIGURE 4.5: An example *benchmark task* using an artwork from the *RMFAB* dataset

[4]If there are N additional information, the resulting vector for that query will be the mean of the $N+1$ ($+1$ to add the caption embedding) embeddings of the task.

Proper nouns alignment This benchmark measures how well the model aligns the proper nouns to the artworks containing them. A proper noun can be a person's name, the name of a city or an important historical event for example. Using *CLIP* as a face recognition model has been studied. Even in the original *CLIP* paper Radford et al., 2021 [42], the authors benchmarked *CLIP* on the *CelebA* dataset (Liu et al., 2018) [28], they measured that *CLIP* had a top-1 accuracy of 59.2% for "in the wild" celebrity image classification when choosing from 100 candidates and a top-1 accuracy of 43.3% when choosing from 1000 possible candidates. Another study by *Aaditya Bhat* and *Shrey Jain* from 2023 [2] compared the performance of different *CLIP* models on cropped images of faces and got better top-1 and top-5 accuracies but still fell short behind the SOTA face recognition models. Yet it is still interesting to fine-tune our *CLIP* model to achieve stronger alignment between proper nouns and the artworks they depict under the restriction that it should not compromise its robustness on general descriptive queries of artworks. Indeed, we should focus more on finetuning our *CLIP* models on general queries because while proper nouns often lack synonyms, allowing for precise retrieval via metadata filters (e.g., finding all artworks with *Jesus* in any of the *Subject Matter* fields), general queries rely on *CLIP*'s ability to understand semantic relationships, abstract concepts, and synonyms within natural language descriptions of artworks.

A proper noun is defined as any term appearing either in the *Subject matter of the iconographies* or either in the *Subject matter of the concepts* with the first character being a capital letter. The table 4.1 presents the repartition of the proper nouns in the 3 *subject matter* fields.

To evaluate proper noun handling, we have created two datasets. The first dataset, named *Exploded Proper Nouns* denoted in the table using the suffix *-EXPLODED* generates separate entries for each proper noun associated with an artwork (e.g., a record with Jesus, Nel Wouters, and Amsterdam would have three entries), mirroring typical user queries where the user queries for only a single proper noun at once. The second dataset named *Attached Proper Nouns* denoted by the suffix *-ATTACHED*, combines all the proper nouns for an artwork into a single string (e.g., "Jesus, Nel Wouters, Amsterdam").

Field	**Unique proper nouns**	**Percentage**
Subject Terms	10	0.60%
Conceptual Terms	142	8.53%
Iconographic Terms	1368	82.16%
Appearing in more than 1 field	145	8.71%
Total	1665	100.00%

TABLE 4.1: Table presenting the repartition of the unique proper nouns in the 3 *subject matter* fields

Generalization capabilities This benchmark measures how well the model handle queries with artworks it has not seen. The approach to measure this capability is identical to the benchmark measuring queries on the *RMFAB* dataset 4.4.1 except that we will be using artworks from the *WikiArt* website. The *WikiArt* dataset [56] [55] that we used has 217 unique styles, from which we kept only the styles with at least 500 artworks. This process resulted in 55 unique styles each giving us 3 artworks resulting in a set of 165 artworks. The list of styles that we kept can be found in E. It must be noted that there is an overabundance of abstract-like styles compared to the *RMFAB* dataset. The relatively small size of this dataset should also be noted, if this work was meant to continue in a form or another, this weak point should be resolved.

For each artwork, we wrote a manual caption in French that was translated in English and in Dutch using *Opus-MT* models. The usual retrieval metrics 4.4.1 are used. This Benchmark serves mainly as an indicator of overfitting.

Captioning tool

Making manual captions and registering additional information on hundreds of artworks is **very** time-consuming. It is therefore very interesting to make this process as smooth and easy as possible. A captioning tool was developed during this project to assist us in writing the captions. The figure 4.6 presents a screenshot of the captioning tool.

FIGURE 4.6: Screenshot of the captioning tool made for this project

Summarization of the benchmark datasets

Queries on the *RMFAB* dataset There are 1816 tasks following the template presented in table 4.2. There are 454 unique artworks, each having 5 tasks linked to them.

Entry name	Description	Example
recordID	The identifier of the artwork linked to this task	1688
category	The category from 2.2.2 that the artwork belongs to	Tableau
focus	What this task focuses on. This is equivalent to the instruction given to the annotator	colors
caption_fr	The French caption written by the annotator for this artwork	Un bateau attaqué par des monstres marins, la tempête, des nuages gris et noirs, un phare sur la colline, un monstre
caption_en	The translated caption in English for this artwork	A boat attacked by sea monsters, storm, gray and black clouds, a lighthouse on the hill, a monster
caption_nl	The translated caption in Dutch for this artwork	Een boot aangevallen door zee monsters, storm, grijze en zwarte wolken, een vuurtoren op de heuvel, een monster
additional_info_fr	If the focus has additional information, the list of additional information given by the annotator	['Couleurs sombres', 'Couleur verte']
additional_info_en	The list of additional information given by the annotator translated in English	['Dark colors', 'Green color']
additional_info_nl	The list of additional information given by the annotator translated in Dutch	['Donkere kleuren', 'Groene kleur']

TABLE 4.2: Template of a task for the *Queries on the RMFAB dataset* benchmark

Proper nouns alignment There are two datasets used for that benchmark, one for the variant -*ATTACHED* and one for the variant -*EXPLODED*. An example task from the variant -*ATTACHED* is presented in table 4.3. An example task from the variant -*EXPLODED* is presented in table 4.4.

Entry name	Description	Example
recordID	The identifier of the artwork linked to this task	64
proper_nouns	A string consisting of the joined proper nouns linked to this artwork. The proper nouns are taken from the *subject matter* fields.	Jésus, Evangiles, Calvaire, Jérusalem, Passion, Christ, Vierge, Nouveau Testament , Crucifixion

TABLE 4.3: Template of a task for the *Proper nouns alignment* benchmark using the -*ATTACHED* variant

Entry name	Description	Example
proper_noun	The unique proper nouns assigned to this task.	Afrique
recordIDs	A list of recordIDs corresponding to the identifiers of the artworks containing this proper noun in at least one of their *subject matter* fields	[137, 717, 1465, 1848, 1849, 1854, 4672, 6163, 6164, 6165, 6166, 6167, 6168, 6460, 6462, 6463, 6464, 8445, 8720]

TABLE 4.4: Template of a task for the *Proper nouns alignment* benchmark using the - *EXPLODED* variant

Generalization capabilities

Entry name	Description	Example
id	The identifier of the artwork in the WikiArt dataset	1
style	The identifier of the artwork in the WikiArt dataset	Romanticism
caption_fr	The caption manually written in French	Portrait d'une femme avec une robe rouge
caption_en	The French caption translated in English	Portrait of a woman in a red dress
caption_nl	The French caption translated in Dutch	Portret van een vrouw in een rode jurk

TABLE 4.5: Template of a task for the *Generalization capabilities* benchmark

4.5 Finetuning a CLIP model on artworks

finetuning a CLIP model necessitates large amount of text-image pairs. A qualitative dataset can be found for many usual tasks like recognizing clothes for example [22]. Sadly, there are not many datasets linking artworks to captions. Moreover, in the initial scope of this project, we decided to only focus on making the models and the interface work in French, which makes finding a dataset even harder. This unavailability of a qualitative French dataset on pairs of artworks-captions can be resolved in at least two ways.

Manual captioning The approach would consist in simply manually writing French captions on the *RMFAB* artworks (and others possibly). It has the big advantage of producing a high quality training dataset that is tailored to our use case. But it is very time-consuming and potentially costly if done in a professional environment.

Synthetic Data This alternative approach consist of using several other *AI* models to generate and translate captions. An example pipeline would be to get an English caption from a VLM model by giving it the artwork as an input. The English caption can then be translated if needed. Using such a pipeline will **mimic** the captioning made by annotators. It is merely an approximation of the process but it has many advantages that made it the best approach in our use case and in similar use cases in our opinion.

Generating captions using a VLM model is not only quick and accessible, it also allows us to give instructions to our virtual annotator. Many VLM models offer the possibility to query it with an instruction (for example a question). This allows us to generate several captions per image by asking the VLM model different questions. Some VLM models can also be easily finetuned. This research path has not been explored in this paper as the results without finetuning the VLM model were plenty sufficient for our use case.

Because of the limited time this project had, we decided on using the latter approach. This section covers the pipelines that have been used during this project and in a broader way, **the pipelines that can be used to finetune a *CLIP* model cheaply and in many languages.**

4.5.1 Environment and parameters

The finetuning was done using a *Google's Colab* environment. This let us quickly and easily run our code on a *NVIDIA A100* GPU which has enough VRAM to fit a batch size of 32 images for the *CLIP L/14* model when finetuning, we believe that a larger batch size would have been better.

The training hyperparameters were found iteratively but it may be possible to find even better ones. Running a grid search on hyperparameters for a large model like *CLIP* can be quite costly. The models in this report were finetuned during 5 epochs as the loss would converge at that time. The learning rate was set to a low 5×10^{-7} to avoid modifying the model's weights to drastically. Finally, the weight decay was set to 0.2. We used the *Adam optimizer* [24].

4.5.2 Initial finetuning

As explained earlier in this paper, the prototypes presented in early February required a performant *CLIP* model on the *RMFAB* artworks. At that time, the scope of the project only called for a performant French search engine.

The side objective of aligning the artworks with the proper nouns linked to them was decided **after** this finetune, therefore the training data used does not contain any information from the *RMFAB* registries.

Pipeline used

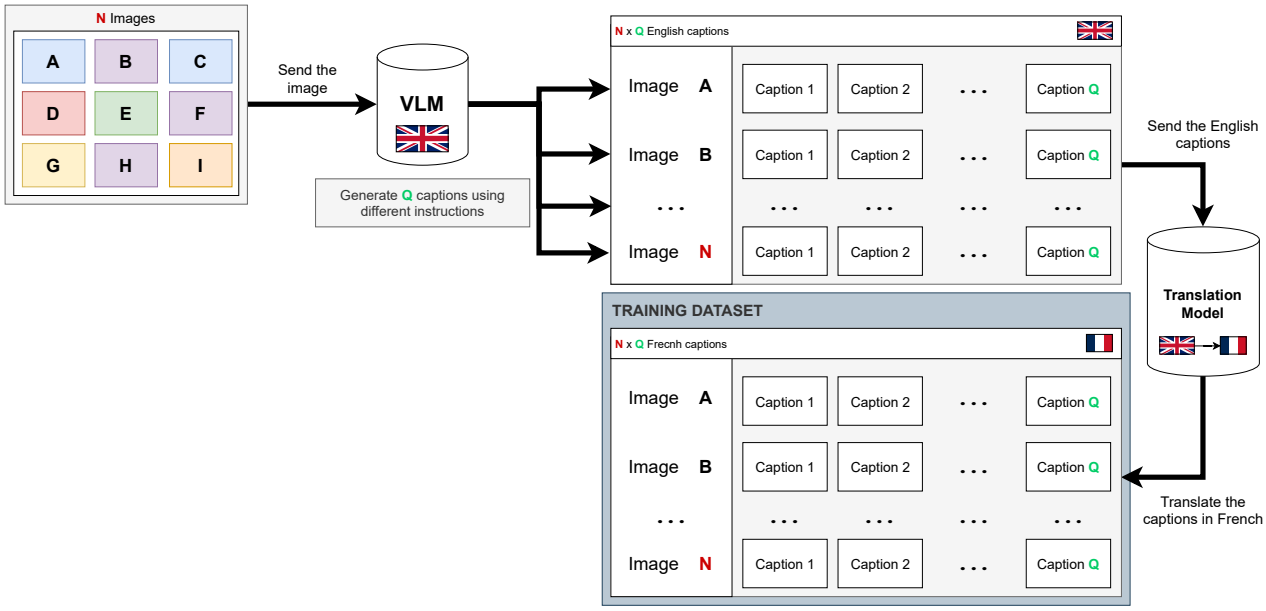


FIGURE 4.7: Diagram of the basic synthetic data pipeline that was used for the prototypes presented in early February

This is a pipeline that generate Q captions per image resulting in $N \times Q$ captions in the chosen language (French in our use case). The images go one by one in the VLM that answers Q prompts on each one. A prompt can be a simple *captioning query* (sometimes available as a native method in VLMs) or it can be a specific instruction like *Enumerate the main colours of this image*.

It allowed us to generate the training data used to finetune to *CLIP* model powering the prototypes presented to the *RMFAB* in early February is presented in the figure 4.7. A subset of $N = 2433$ artworks from the *RMFAB* gallery is passed through the pipeline one by one. First, a VLM (*Moondream2* made by *Moondream.ai* [**moondream\textit{AI}**] in our case) is used to generate $Q = 5$ captions based on the artwork. The following instructions are:

1. *Use the caption method from moondream directly*

2. What is a short caption for this image where you speak about objects ?
3. What is a short caption for this image where you speak about colors ?
4. What is a short caption for this image where you speak about luminosity ?
5. What is a short caption for this image where you speak about emotions ?

This process yielded $N \times Q = 12165$ captions in English from a subset of $N = 2433$ artworks ($Q = 5$ captions per artwork). Then a translation model from the *Opus-MT* family [] [49] (*opusenfr* specifically [34]) is used to translate the captions in French. The final output of this pipeline is a dataset of 12165 French captions describing 2433 unique artworks.

Benchmarks

Benchmark 1 Finetuning the *L/14 CLIP* model on the 12165 generated French captions improved significantly the *MRR* in French for both variants of the benchmark (figure 4.8). Interestingly, it also improved greatly the *MRR* in English and in Dutch. The *CLIP* model is overall performing better on artworks even if it was not finetuned on English or Dutch captions.

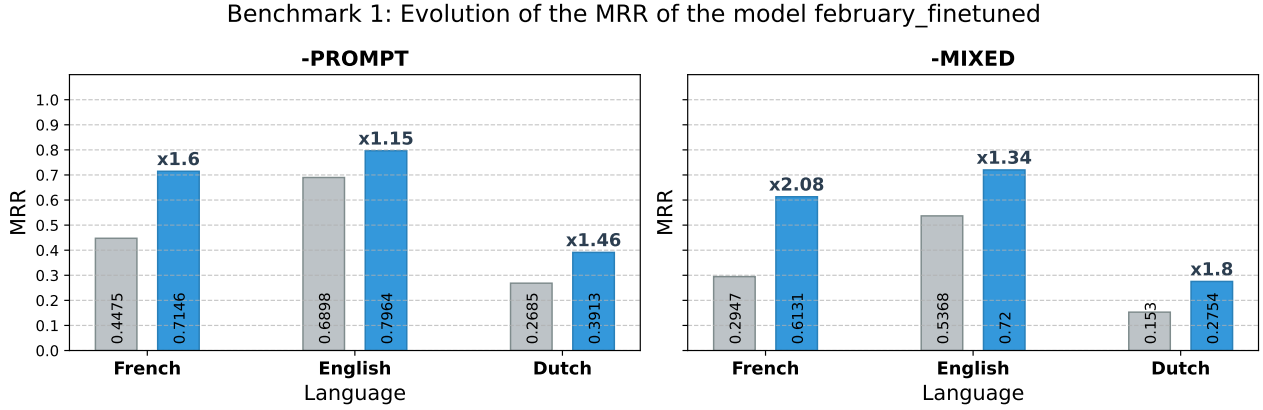


FIGURE 4.8: Benchmark 1: Evolution of the MRR for the February finetune

For the *-PROMPT* variant, an impressive 0.7146 French *MRR*[5] was achieved after finetuning, surpassing the *MRR* in English without finetuning (0.6898). The model stills performs the best in English, this is probably due to the initial training process done by *OpenAI* on mostly English captions. The table 4.6 summarizes the results per variant and per language of the Benchmark 1.

We could expect an hypothetical search engine using this model and the subset of artworks used in Benchmark 1 (454 artworks) to return a relevant artwork in the first 10 results 93.94% of the times in French, 96.70% in English and 66.91% in Dutch (when prompting naively by joining the content, colors, luminosity and emotions with a comma)[6] (fig. 4.9) (table 4.6). This is already an excellent result in French and English.

[5]This is the mean *MRR* over all the *tasks* of the Benchmark 1.
[6]The *recall* values are the mean recall values over all the *tasks* of the Benchmark 1.

Benchmark variant	Language	Avg. Position	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
-PROMPT	French	3.6476	0.7146	0.5909	0.8051	0.8816	0.9394	0.7167	0.7484	0.7672
-PROMPT	English	**2.3519**	**0.7964**	**0.6971**	**0.8789**	**0.9306**	**0.967**	**0.8034**	**0.8251**	**0.837**
-PROMPT	Dutch	19.5402	0.3913	0.2638	0.4361	0.5275	0.6691	0.3633	0.4008	0.4468
-MIXED	French	5.3392	0.6131	0.475	0.699	0.7856	0.8825	0.6059	0.6416	0.6737
-MIXED	English	3.4838	0.72	0.5999	0.8084	0.873	0.928	0.7232	0.75	0.7677
-MIXED	Dutch	31.6858	0.2754	0.1586	0.2944	0.3965	0.5382	0.237	0.279	0.3245

TABLE 4.6: Benchmark 1 results for February finetune

The observation that finetuning the model on French caption results in better performance in Dutch and in English is a key observation. The fact that the model improved in English indicates that the model did not lose capabilities in its native language (the model did not overfit on French captions).

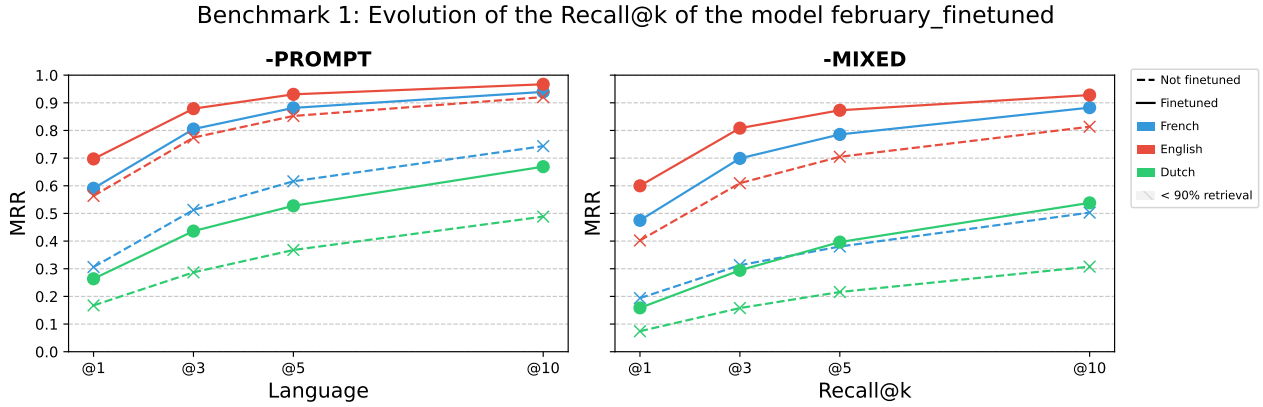


FIGURE 4.9: Benchmark 1: Evolution of the Recall for the February finetune

When looking at the impact of the additional information on the *MRR*, we see an unexpected effect. Giving additional information to our *CLIP* model like the colors, the luminosity or the emotion detected by the annotator results in worse *MRR* 4.10. This trend is observed in both variant of this Benchmark, but it is way stronger in the *-MIXED* variant. We believe that this is because the caption is *drowned* by the additional information. The *-MIXED* variant takes the arithmetic mean of all the embeddings, therefore, the information residing in the caption (the most discriminant information) will have less impact on the final textual embedding the more additional information there are. This is a major drawback of mixing textual vector to circumvent the token limitation of *CLIP*.

The fact that this trend is also visible on the *-PROMPT* variant is surprising as there is no *drowning* of the caption for this variant. Our *February finetune* model seems to have a hard time to grasp the emotion and the luminosity of an artwork. There seems to be little difference when providing colour information to our model.

The categories defined in 2.2.2 seems to have an impact of the performance of the model. As the table F.1 describes, on average a painting (*Tableau*) will be better ranked than a drawing (*Dessin*) or a sculpture (*Sculpture*). This may be because *CLIP* has been trained on photographic images that share more features with painting than with the other two categories.

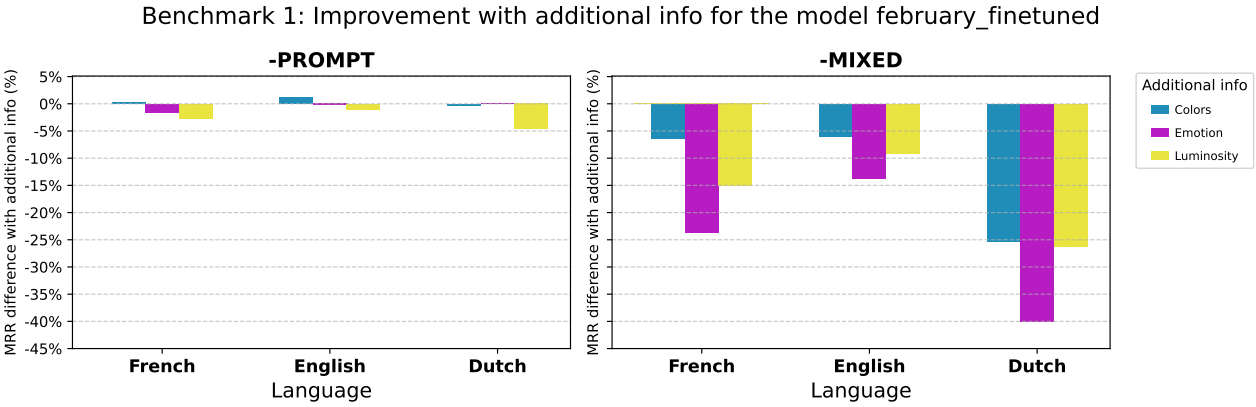


FIGURE 4.10: Benchmark 1: Impact on the MRR per focus on the February finetune

Benchmark 2 When testing the search engine prototype using this model, the *RMFAB* staff quickly noticed that it was not very capable when querying for proper nouns like the name of a city or a person. This observation is confirmed by the Benchmark 2. There is a very slight improvement over the baseline but the *February finetune* model is logically not capable on this aspect has it has not been trained on any data involving proper nouns. The table 4.7 summarizes the metrics for the *-ATTACHED* variant of this benchmark.

	Avg. Position	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
Base model	333.1152	0.0748	0.0347	**0.0773**	0.1005	0.1394	0.0593	0.0688	0.0813
February finetuned	**314.8459**	**0.0788**	**0.0379**	0.0752	**0.1026**	**0.1547**	0.0594	**0.0706**	**0.0873**
Difference (abs)	-18.2693	0.004	0.0032	-0.0021	0.0021	0.0153	0.0	0.0018	0.006
Difference (%)	-5.4844%	5.3311%	9.0909%	-2.7211%	2.0942%	10.9434%	0.0740%	2.6298%	7.3855%

TABLE 4.7: Benchmark 2 - Attached Metrics Comparison (*February finetune*)

Benchmark 3 As the figure 4.11 supports, the *February finetune* model improves on artworks it has never seen in all three languages. This is a great news as our model do not seem to overfit on the *RMFAB* artworks.

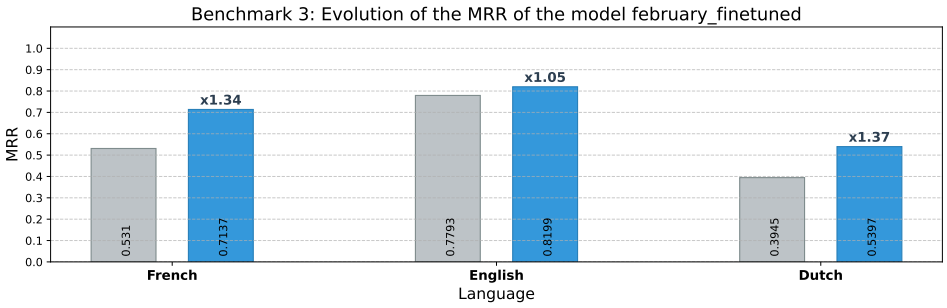


FIGURE 4.11: Benchmark 3: Evolution of the MRR of the model (*February finetune*)

4.5.3 Improving the proper nouns performance

Although impressive, the demonstration of the search engine made using the *February finetune* model lacked in one specific area. When searching for example for *Toison d'Or*, the search engine returned unrelated results. This is to be expected as the training data for this model did not contain any information outside the knowledge of *CLIP* and the VLM model used (*moondream2*).

From this point forward, the full dataset of 5301 artworks made available by the *RMFAB* is used. This of course impacts the results and should be taken into account when comparing the model *February finetuned model* to the next models.

Pipeline

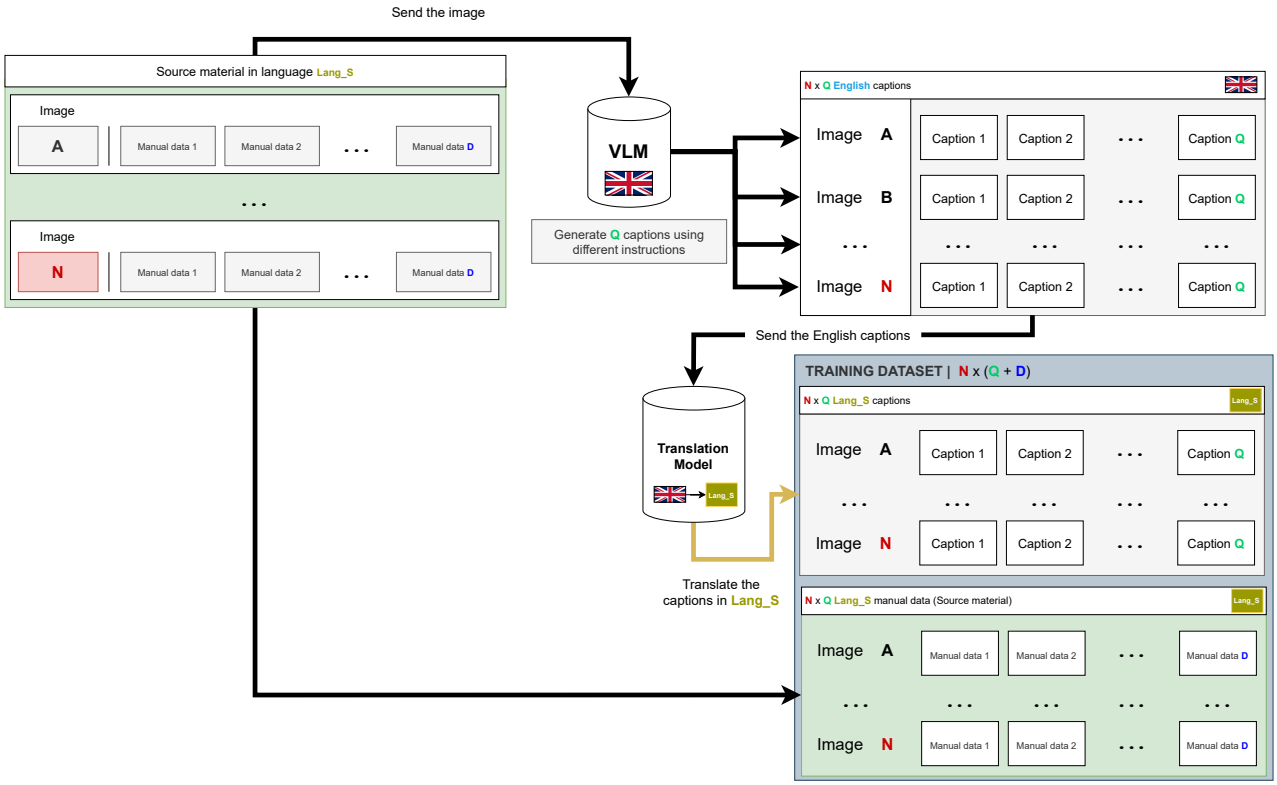


FIGURE 4.12: Diagram of the basic synthetic data pipeline with the addition of manual data

This pipeline generates a set of captions in a chosen language *Lang_S*. Let N be the number of images, Q the number of instructions given to the *VLM* and D the (maximum) number of manual information about each image; we can generate at most $N \times (Q + D)$ captions for N images. Obviously its biggest drawback is the need for the pre-existing information.

In the *RMFAB* digital gallery we can leverage the *subject matter* fields as the manual information. Using this source, we recovered 7301 manual information[7] on 3642 unique artworks. The table G.1 describes the exact composition of this manual information dataset. This pipeline has been used to finetune the *March finetune* model. With $N = 5301$, $Q = 5$, the pipeline generated 26505 captions. After truncation and after adding the 7301 manual information, the training dataset contained 33792 pairs of image-caption.

The instructions given to the *VLM* were slightly tweaked. The exact instructions can be found below:

1. *Use the caption method from moondream directly*
2. What objects do you see ?
3. What colors do you see ?
4. Is this image bright or dark ?
5. What emotion do you feel when looking at this image ?

Benchmarks

Benchmark 1 The mean *MRR* improved by 6% in French, 3% in English and 15% in Dutch (see figure 4.13) compared to the *February finetune* model. This is probably due to a larger pool of artworks, better *VLM* instructions and the addition of the manual information. Once again, training the *CLIP* model only on French captions improved the metrics in the other two languages, even more surprisingly, the largest relative improvement is in Dutch.

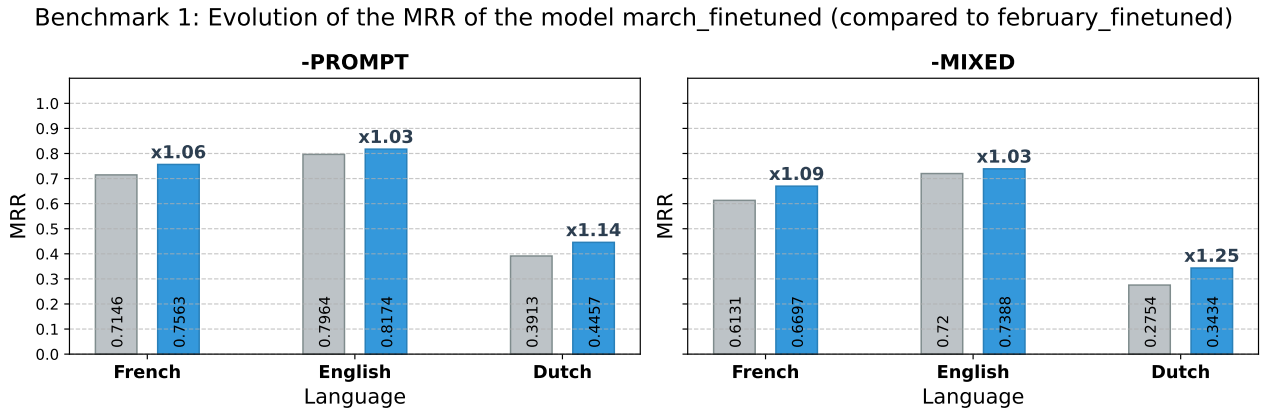


FIGURE 4.13: Benchmark 1: Evolution of the MRR for the *March finetune* (compared to *February finetune*)

[7] An overflow on 10 tokens was allowed since for these cases, even with the truncation of those tokens most of the information stayed relevant.

As the section 4.2 explained, the *L/14* pretrained *CLIP* model that we use already aligns artworks textual descriptions of them quite well. This zero-shot capability is the strongest in English but is also present to a lesser extent in French and Dutch. The *CLIP* embedding of an English caption and the embeddings of its translations are therefore similar. We believe that the improvement in Dutch and English when training on French captions arises mostly from the realigning of the image encoder of *CLIP* to better suit the style of artworks. Ideally (for a single model), the embedding of a caption in French should be equal to the embedding of the translated caption in another language. This idea opens up the door to a two step training process. First, training the full *CLIP* model on pairs of image to caption in a single language. Secondly, aligning the textual encoder such that the embedding of a caption and its translations are closer. This idea of treating different textual inputs as paraphrase has been explored in a paper by *Kim et al. (2024)* [23]. This path has not been explored extensively in this thesis.

Benchmark 2 Adding the manual information from the *RMFAB* digital gallery improves significantly the the proper nouns benchmark with a two-fold increase in all metrics in the *-ATTACHED* variant. But even with this improvement, the *MRR* is still low at only 0.1474. This may be due to the limited number of artworks per proper nouns in as on average, a proper noun is evoked in only 2.85 artworks[8].

	Avg. Position	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
February finetuned	314.8459	0.0788	0.0379	0.0752	0.1026	0.1547	0.0594	0.0706	0.0873
March finetuned	**126.1799**	**0.1474**	**0.071**	**0.1478**	**0.2004**	**0.3135**	**0.1146**	**0.1363**	**0.1728**
Difference (abs)	-188.666	0.0685	0.0331	0.0726	0.0978	0.1589	0.0552	0.0657	0.0855
Difference (%)	-59.9233%	86.9213%	87.5000%	96.5035%	95.3846%	102.7211%	92.9861%	92.9531%	97.8993%

TABLE 4.8: Benchmark 2 - Attached Metrics Comparison (*March finetune*)

But even with more training rows on proper nouns, recognizing faces, places or events on artworks is quite hard. There are two reasons why *CLIP* might have a low performance ceiling on this task. The first reason is that before encoding an image into its embedding representation, the processor of *CLIP* resizes the image to a somewhat low resolution of only 224×224 pixels for the *B/32* and *L/14* models (336×336 for the largest model used in this thesis *L/14@336px*). Fine details like the characteristics of a face or the flags on a tower that might give us the necessary hints to recognize the proper nouns assigned to an artwork are simply not visible on such low resolutions. The second reason is that two artworks depicting the same person or the same event can have vastly different styles. For example, one could be a drawing while the other could be a painting. This is an inherent difficulty when working with artworks.

It may be possible to further increase the metrics on proper nouns, specifically on face recognition by training the *CLIP* model on cutouts of the faces (by first cropping the face manually or via a face detection model) [2]. Of course, if the data is available like in the *RMFAB* case, using a search input that queries by proper nouns associated or using *Named Entity Recognition* (NER) on the textual input of the user would be a more efficient way to search for an artwork depicting a specific person, place or event.

[8]This value of 2.85 artworks per proper noun is the mean length of the *recordIDs* field in the *-EXPLODED* variant of the second benchmark 4.4.

A possible side effect of naively mixing the training rows in the pipeline used 4.12 is the imbalance between *captions* rows and *manual information* rows. In our case, for 1 *manual information* row there are 3.63 *caption* rows; we have decided against using a balancing mechanism because the search engine that we developed offers a method to query by proper nouns directly. A simple method could be to modify the cross entropy loss used when finetuning *CLIP*.

Suppose that we have a n_X *captions* training rows and n_Y manual training rows. Let $L^{(i)}_{\text{text}}$ and $L^{(i)}_{\text{image}}$ be the per-sample cross-entropy losses for text and image directions, let $t_i \in \{X, Y\}$ denote the type of the sample i and let w_i denote the weight of the sample i equal to $\frac{1}{n_X}$ if $t_i = X$ or $\frac{1}{n_Y}$ if $t_i = Y$. For a batch B_j, we can compute the per batch loss L_j:

$$L_j = \frac{1}{2}\left(\sum_{i=1}^{B_j} w_i L^{(i)}_{\text{text}} + \sum_{i=1}^{B_j} w_i L^{(i)}_{\text{image}}\right) \tag{4.1}$$

Benchmark 3 Between the *March finetune* model and the *February finetune* model, the *MRR* improves by about 4% on the third Benchmark (see fig. 4.14). This is coherent with the slight improvement seen in Benchmark 1 (see 4.13).

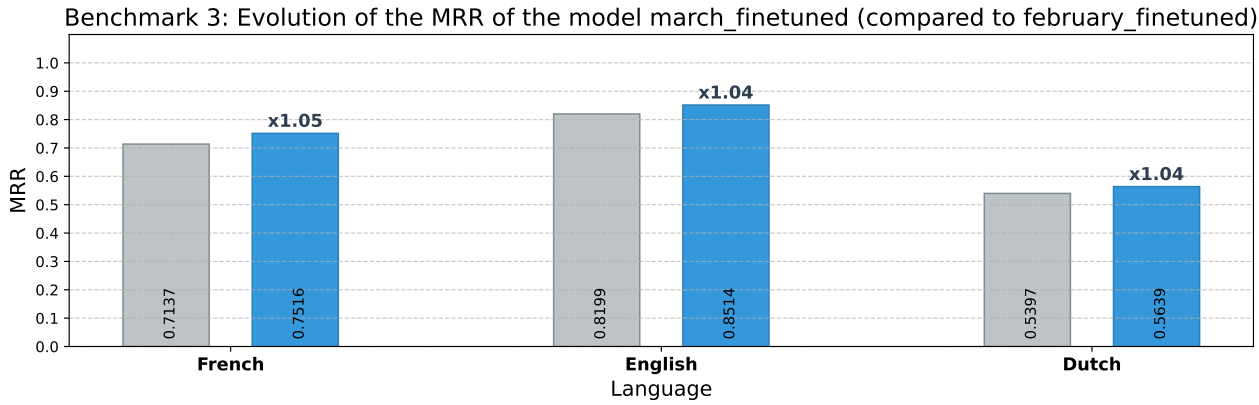


FIGURE 4.14: Benchmark 3: Evolution of the MRR of the model (*March finetune*)

4.5.4 Multilingual model

The previous two pipelines lead to more performant *CLIP* models on artworks when querying in French, but also in English and Dutch. This language generalization capability opens the door to a search engine working in the three languages simultaneously. This is great news for the *RMFAB* as theses are the three official languages in Belgium. We decided on finetuning a final version of the models in the three languages named *art-base*.

Pipeline used

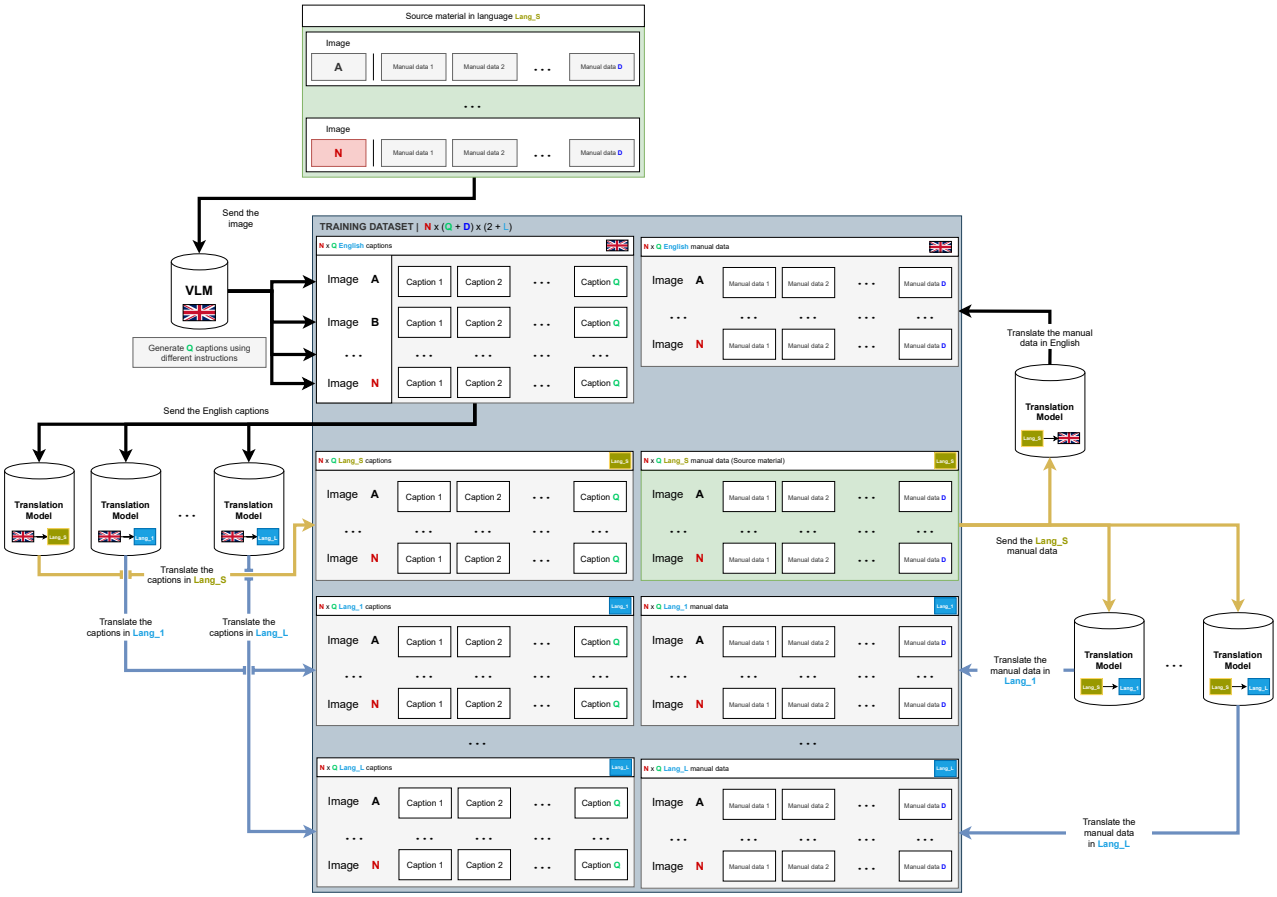


FIGURE 4.15: Diagram of the multilingual synthetic data pipeline that can be used to finetune a *CLIP* model in a multilingual context

This pipeline generates multilingual captions and multilingual manual information. In our case, we have the manual information in French, in the diagram 4.15 the French language is denoted as *Lang_S* (our source language). The N images are passed through a *VLM* producing Q captions per artworks using different instructions. The English generated captions are translated in *Lang_S* and in L additional languages resulting in a total of $N \times (L + 2)$ captions[9]. The manual information in

[9]In total we will have $L + 2$ languages since we add to the set of L languages English and *Lang_S*

Lang_S is translated in English and in the L additional languages. If an image has at most D manual information, this pipeline can generate at most $N \times (Q + D) \times (2 + L)$ training rows.

The final training dataset used in this project was generated using this pipeline. Our source language was French. The manual information set comprised of 7301 rows, it was translated in English and in Dutch. There wasn't an *Opus-MT* model to translate directly from French to Dutch so it was done using two models, first from French to English and then from English to Dutch. This adds another layer of approximation that we believe impacts the quality of the captions. The set of 26505 English captions generated by *moondream2* was translated in French and in Dutch. After removing the rows surpassing the 10 tokens overflow allowed, the final training dataset contained 100873 training rows from only 5301 images.

Benchmarks

Benchmark 1 The *art-base* model performed remarkably well in the three languages with the highest improvement in Dutch. But even when finetuning with the same number of rows in the three languages we still observe that English is toping the benchmark with French following closely and Dutch last by quite a margin. *CLIP* has been finetuned on mostly English captions and therefore it is not surprising to see English topping the benchmarks. The fact that Dutch does not catch up with French in the same manner as French caught up with English is hard to definitely explain. Although we can suspect that the 2-hop translation technic used on the *manual information* is partly responsible. It may also be because the translation capabilities of the English to Dutch *Opus-MT* model differs from the English to French *Opus-MT* model or simply because the *CLIP* checkpoint we start from is already more aligned with French than with Dutch as the figure 4.16 supports.

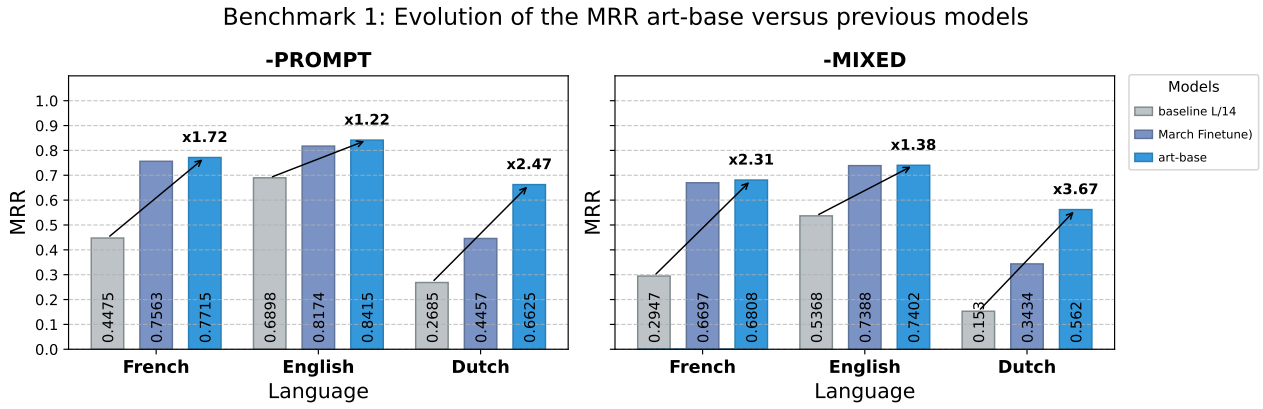


FIGURE 4.16: Benchmark 1: Evolution of the MRR art-base versus previous models

This model is the first one to have *recall@10* values superior to 90% for all languages on the *-ATTACHED* benchmark (see table 4.9). This is a sufficient model for our use case. We believe that it is possible make the models even better in a multilingual context. For example, *M-CLIP* [5] is a improvement over the initial *CLIP* model that replaces the textual encoder with *XLM-RoBerta* and that has been finetuned on translations pairs (coincidentally automatically translated by *Opus-MT* models like us !).

Benchmark variant	Language	Avg. Position	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
-PROMPT	French	2.6966	0.7715	0.6591	0.8612	0.9135	0.9587	0.7794	0.801	0.8158
-PROMPT	English	**1.7594**	**0.8415**	**0.7494**	**0.9251**	**0.9565**	**0.9835**	**0.8542**	**0.8672**	**0.8759**
-PROMPT	Dutch	5.3078	0.6625	0.5358	0.7528	0.8216	0.9042	0.662	0.6903	0.7172
-MIXED	French	4.3598	0.6808	0.5573	0.7592	0.8326	0.909	0.677	0.7076	0.7325
-MIXED	English	2.9339	0.7402	0.6256	0.8245	0.8935	0.9493	0.7423	0.7708	0.789
-MIXED	Dutch	9.0962	0.562	0.4295	0.6351	0.7269	0.8238	0.5497	0.5873	0.6186

TABLE 4.9: Benchmark 1 results for *art-base*

Benchmark 2 The proper nouns benchmark improved significantly as the table 4.10 describes. This is somewhat surprising as we did not add new proper nouns information in the training dataset compared to the *March finetune* dataset. The *manual information* training rows were translated in 2 languages tripling the number of *manual information* training rows, but by its nature, a proper noun cannot be translated. It would appear that finetuning the model on a larger number of rows allowed the model to surpass the plateau it hit during the *March finetune* training.

	Avg. Position	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
March finetuned	126.1799	0.1474	0.071	0.1478	0.2004	0.3135	0.1146	0.1363	0.1728
art-base	**68.9563**	**0.2435**	**0.1315**	**0.272**	**0.3519**	**0.4724**	**0.213**	**0.2459**	**0.2849**
Difference (abs)	-57.2236	0.0961	0.0605	0.1241	0.1515	0.1589	0.0984	0.1096	0.1121
Difference (%)	-45.3508%	65.2349%	85.1852%	83.9858%	75.5906%	50.6711%	85.8608%	80.4082%	64.8616%

TABLE 4.10: Benchmark 2 - Attached Metrics Comparison (*art-base*)

Benchmark 3 Interestingly, the third Benchmark for the *art-base* model shows a slightly negative evolution in French and English with our multilingual dataset compared to the *art-base* model (see fig. 4.17). As expected, Dutch improved by a large amount (about 30%). This is the first sign of overfitting encountered that may be due the amount of rows we are training on, especially the *Subject Matter* rows as they are very specific to the *RMFAB*.

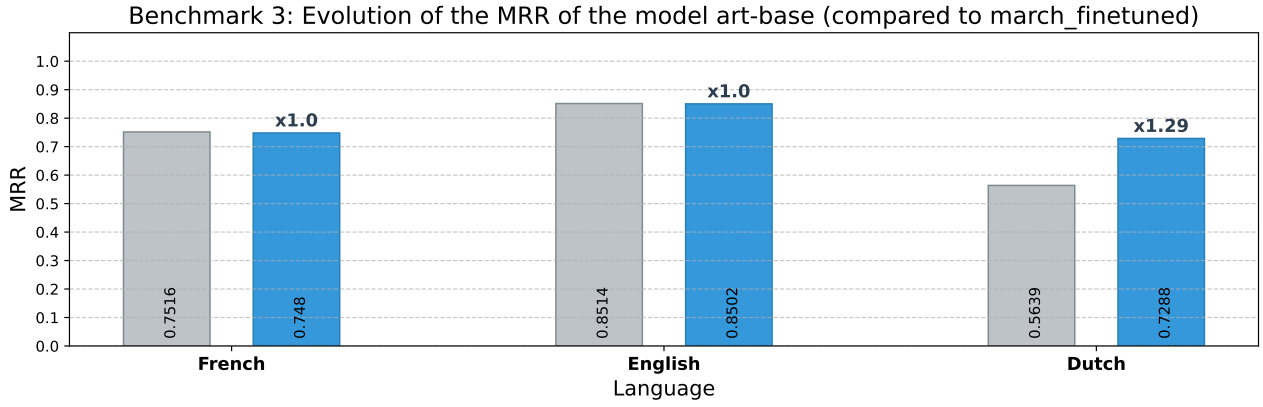


FIGURE 4.17: Benchmark 3: Evolution of the MRR of the model (*art-base*)

4.5.5 Varying the model size

The models finetuned in the previous section were all based on the *L/14 CLIP* model published by *OpenAI* [35]. But there are many smaller or larger models published by *OpenAI* or by other entities like the *OpenCLIP* project [19] [9] [43] [47]. If a smaller model proves to be as capable as the *L/14* model it would be sensible to use it as it would reduce the inference latency and the training time. On the opposite end, if a larger model produces significantly better results, it could be worth the additional computational cost. This section compares the metrics of three models (see table 4.11) finetuned on the multilingual dataset described in section 4.5.4:

Model name	Architecture	Input resolution	Number of parameters	Embedding dimension	Indicative forward time	Forward time ratio versus art-base
art-mini	B/32	224x224	151277313	512	0.022 +- 0.004	0.5946
art-base	L/14	224x224	427616513	768	0.037 +- 0.001	1.0000
art-large	L/14@336px	336x336	427944193	768	0.065 +- 0.002	1.7568

TABLE 4.11: *art-* model family description

Training hyperparameters

We deliberately choose to keep the training hyperparameters defined in section 4.5.1 except for the batch size. For the *art-mini* model using the smaller *B/32* architecture, the batch size used was set to 256 . For the *art-large* model using the larger *L/14@336px* architecture, the batch size was set to 16. Of course, by tweaking the hyperparameters, one could possibly improve the models performance further.

Benchmarks

Benchmark 1 As the figure 4.18 presents, the best model on the *-PROMPT* variant of the first benchmark is *art-base* no matter the language. Seeing this intermediate model perform better than the smaller *art-mini* is expected, but seeing the larger *art-large* model using higher resolution images as inputs perform worse is surprising. The baseline models *basic-base* and *basic-large* do perform similarly which may explain why their finetune versions also do. We believe that by tweaking the hyperparameters when finetuning the larger model we could squeeze out a few percentage points but not much.

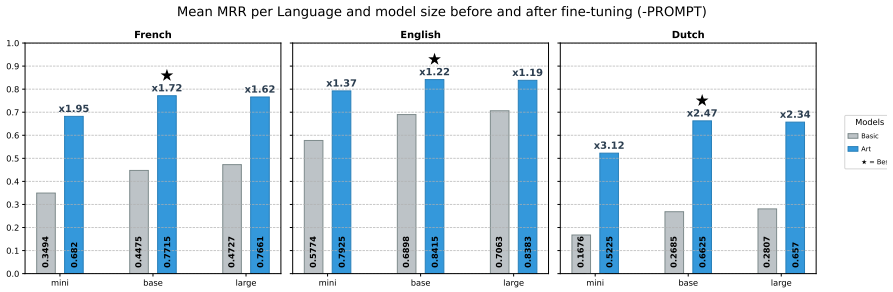


FIGURE 4.18: Benchmark 1: Mean MRR per Language and model size before and after finetuning

Although inferior to the other two models, the performance shown by *art-mini* is still quite impressive as it contains less than 40% of the parameters of the *art-base* model.

Benchmark 2 As table 4.12 presents, the model the most aligned with the proper nouns is consistently the *art-base* model with the *art-large* model slightly behind. This is quite surprising as we expected the larger image resolution used by the *L/14@336px* architecture to improve significantly the capabilities of the model to recognize the fine details necessary to infer the person or the event represented.

A discernible performance drop is evident in the *art-mini* model when contrasted with the *art-base* and *art-large* variants. This drop suggests that the reduced number of parameters of the *B/32* model architecture constrains its representational capacity too much for it to extract and generalize the *salient* features necessary for accurate proper noun recognition.

Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
basic-mini	409.7033	0.0467	0.0205	0.0426	0.0584	0.0821	0.0333	0.0398	0.0474
basic-base	333.1152	0.0748	0.0347	0.0773	0.1005	0.1394	0.0593	0.0688	0.0813
basic-large	312.4955	0.0774	0.0363	0.0773	0.1026	0.1499	0.0596	0.0701	0.0853
art-mini	179.4371	0.1301	0.0673	0.1341	0.1767	0.2504	0.1058	0.1234	0.1472
art-base	**68.9563**	**0.2435**	**0.1315**	**0.272**	**0.3519**	**0.4724**	**0.213**	**0.2459**	**0.2849**
art-large	75.4093	0.2136	0.1142	0.2283	0.3083	0.4156	0.18	0.2129	0.2474

TABLE 4.12: Metrics table with the 3 baseline models and the 3 *art-* models on Benchmark 2 *-ATTACHED*

Benchmark 3 When looking at the first Benchmark, the best model is consistently the *art-base* model, this is not especially the case for the third Benchmark as out of the three languages, two perform the best using the largest model *art-large* (see fig. 4.19). Similarly to the first Benchmark, English performs the best followed by French and Dutch. The relatively small size of the third Benchmark dataset could explain the small difference between the best *MRR* in Dutch and in French compared to what can be observed in figure 4.18.

Use cases

We decide to set the default model used by our search engine to *art-base* as it is the most capable model out of the three sizes. We still allow the user to select the *mini* or *large* models if he prefers their outputs.

We did not find a usage that would require the *art-large* model. It performs worse than *art-base* on all metrics and it is more computationally expensive. This is quite disappointing as we had hoped that the higher resolution would open new possibilities for our search engine and the following research based on our models.

The *art-mini* model on the other hand is very interesting. It could be used in budget-sensitive application like a public search engine. The performance of this smallest model are good enough

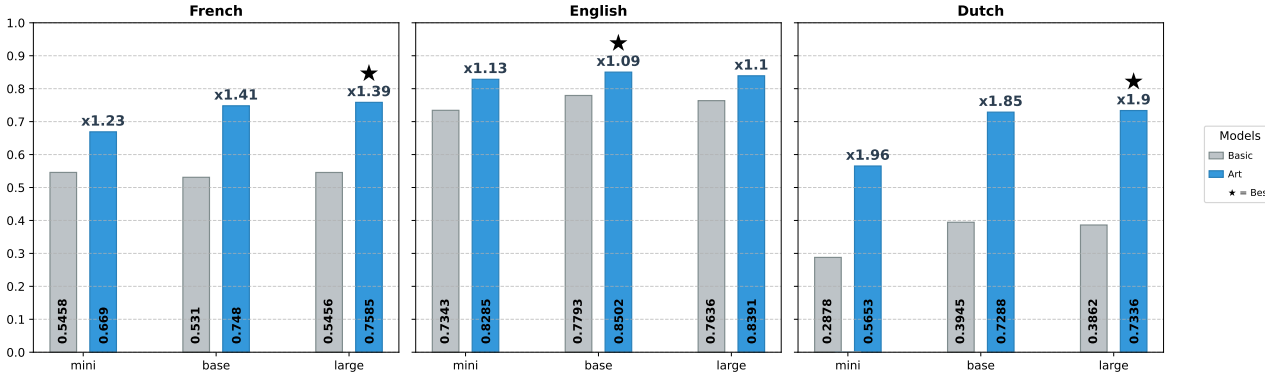


FIGURE 4.19: Benchmark 3: Mean MRR per Language and Model Size

for most uses cases particularity in French and English. The relatively low *MRR* in Dutch (see figure 4.18) indicates that the *art-mini* model might benefit from **more qualitative** training data in Dutch (for example: manually translating captions).

4.5.6 Summary of the models

Benchmark 1 - *PROMPT*

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	21.443	0.349	0.226	0.383	0.478	0.593	0.319	0.358	0.395
	L/14	basic-base	12.979	0.448	0.306	0.513	0.616	0.743	0.425	0.468	0.509
	L/14@336px	basic-large	12.331	0.473	0.336	0.539	0.629	0.743	0.454	0.492	0.529
cap-FR-1	L/14	February Finetune	3.648	0.715	0.591	0.805	0.882	0.939	0.717	0.748	0.767
cap-FR-2 & ico-FR		March Finetune	2.929	0.756	0.64	0.85	0.903	0.952	0.765	0.786	0.802
cap-TRI & ico-TRI	B/32	art-mini	3.73	0.682	0.546	0.782	0.86	0.928	0.685	0.717	0.739
	L/14	art-base	**2.697**	**0.772**	**0.659**	**0.861**	**0.914**	**0.959**	**0.779**	**0.801**	**0.816**
	L/14@336px	art-large	2.839	0.766	0.656	0.849	0.911	0.957	0.77	0.796	0.811

TABLE 4.13: Table containing the results of Benchmark 1 (variant -*PROMPT*) in French

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	7.292	0.577	0.448	0.644	0.736	0.829	0.564	0.601	0.632
	L/14	basic-base	3.732	0.69	0.563	0.774	0.852	0.921	0.688	0.72	0.743
	L/14@336px	basic-large	3.579	0.706	0.577	0.806	0.878	0.937	0.711	0.741	0.76
cap-FR-1	L/14	February Finetune	2.352	0.796	0.697	0.879	0.931	0.967	0.803	0.825	0.837
cap-FR-2 & ico-FR		March Finetune	1.938	0.817	0.716	0.91	0.947	0.978	0.83	0.846	0.856
cap-TRI & ico-TRI	B/32	art-mini	2.112	0.792	0.68	0.888	0.941	0.979	0.803	0.825	0.837
	L/14	art-base	**1.759**	**0.841**	**0.749**	**0.925**	**0.956**	0.983	**0.854**	**0.867**	**0.876**
	L/14@336px	art-large	1.845	0.838	0.747	0.914	0.947	**0.985**	0.848	0.861	0.874

TABLE 4.14: Table containing the results of Benchmark 1 (variant -*PROMPT*) in English

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	61.206	0.168	0.083	0.178	0.244	0.349	0.136	0.164	0.197
	L/14	basic-base	39.613	0.268	0.167	0.287	0.368	0.488	0.235	0.269	0.308
	L/14@336px	basic-large	38.787	0.281	0.178	0.309	0.378	0.492	0.252	0.281	0.318
cap-FR-1	L/14	February Finetune	19.54	0.391	0.264	0.436	0.528	0.669	0.363	0.401	0.447
cap-FR-2 & ico-FR		March Finetune	17.358	0.446	0.316	0.506	0.594	0.718	0.426	0.462	0.502
cap-TRI & ico-TRI	B/32	art-mini	10.018	0.522	0.367	0.616	0.695	0.823	0.514	0.546	0.588
	L/14	art-base	**5.308**	**0.662**	**0.536**	0.753	**0.822**	**0.904**	**0.662**	**0.69**	**0.717**
	L/14@336px	art-large	5.459	0.657	0.523	**0.755**	0.815	**0.904**	0.659	0.684	0.713

TABLE 4.15: Table containing the results of Benchmark 1 (variant -*PROMPT*) in Dutch

Benchmark 1 - *MIXED*

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	40.432	0.25	0.151	0.26	0.331	0.448	0.216	0.245	0.283
	L/14	basic-base	33.143	0.295	0.194	0.313	0.38	0.503	0.263	0.291	0.331
	L/14@336px	basic-large	33.707	0.311	0.203	0.343	0.416	0.521	0.284	0.315	0.349
cap-FR-1	L/14	February Finetune	5.339	0.613	0.475	0.699	0.786	0.883	0.606	0.642	0.674
cap-FR-2 & ico-FR		March Finetune	**4.257**	0.67	0.532	**0.77**	**0.838**	**0.914**	0.673	0.701	0.726
cap-TRI & ico-TRI	B/32	art-mini	4.996	0.63	0.491	0.725	0.802	0.896	0.628	0.66	0.691
	L/14	art-base	4.36	**0.681**	**0.557**	0.759	0.833	0.909	**0.677**	**0.708**	**0.733**
	L/14@336px	art-large	4.684	0.672	0.551	0.75	0.83	0.907	0.668	0.701	0.725

TABLE 4.16: Table containing the results of Benchmark 1 (variant -*MIXED*) in French

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	12.921	0.483	0.358	0.544	0.625	0.733	0.466	0.499	0.534
	L/14	basic-base	9.098	0.537	0.402	0.609	0.705	0.814	0.522	0.562	0.597
	L/14@336px	basic-large	9.014	0.558	0.426	0.627	0.721	0.832	0.543	0.581	0.618
cap-FR-1	L/14	February Finetune	3.484	0.72	0.6	0.808	0.873	0.928	0.723	0.75	0.768
cap-FR-2 & ico-FR		March Finetune	**2.891**	0.739	0.62	**0.828**	0.885	**0.949**	0.743	0.766	0.788
cap-TRI & ico-TRI	B/32	art-mini	3.211	0.716	0.596	0.804	0.872	0.941	0.717	0.745	0.768
	L/14	art-base	2.934	0.74	0.626	0.825	**0.894**	**0.949**	0.742	0.771	0.789
	L/14@336px	art-large	3.096	**0.747**	**0.637**	0.826	0.886	0.946	**0.749**	**0.774**	**0.793**

TABLE 4.17: Table containing the results of Benchmark 1 (variant -*MIXED*) in English

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	93.093	0.096	0.036	0.092	0.137	0.21	0.068	0.086	0.11
	L/14	basic-base	70.828	0.153	0.074	0.158	0.216	0.308	0.122	0.145	0.175
	L/14@336px	basic-large	71.416	0.162	0.082	0.169	0.233	0.325	0.13	0.157	0.187
cap-FR-1	L/14	February Finetune	31.686	0.275	0.159	0.294	0.396	0.538	0.237	0.279	0.324
cap-FR-2 & ico-FR		March Finetune	27.313	0.343	0.219	0.391	0.484	0.597	0.317	0.356	0.392
cap-TRI & ico-TRI	B/32	art-mini	14.125	0.452	0.304	0.529	0.631	0.749	0.434	0.477	0.515
	L/14	art-base	**9.096**	**0.562**	**0.43**	**0.635**	**0.727**	0.824	**0.55**	**0.587**	**0.619**
	L/14@336px	art-large	9.275	0.555	0.419	0.626	0.721	**0.825**	0.54	0.58	0.613

TABLE 4.18: Table containing the results of Benchmark 1 (variant -*MIXED*) in Dutch

Benchmark 2 - *EXPLODED*

The table 4.19 measures the percentage of times when querying for a proper noun, an image having it in one of its *Subject Matter* fields was in the first k results for $k \in \{1, 3, 5, 10, 20, 30\}$. When using the search engine, the user sees about 10 to 20 artworks directly depending on the resolution of his screen.

Model name		**Artwork containing the proper nouns in the first k results**					
	k	1	3	5	10	20	30
basic-mini		0.0288	0.0539	0.0671	0.0978	0.1436	0.1737
basic-base		0.0493	0.0813	0.1023	0.1337	0.1867	0.2323
basic-large		0.0497	0.0851	0.1119	0.1467	0.1877	0.2356
March finetune		0.089	0.1568	0.1889	0.2579	0.3396	0.3929
February finetune		0.0512	0.0882	0.1172	0.1499	0.2023	0.232
art-mini		0.0845	0.1296	0.1563	0.2019	0.2603	0.3054
art-base		0.1493	0.2515	0.3011	0.3845	0.4674	0.5151
art-large		0.1332	0.2203	0.2748	0.3481	0.4304	0.4806

TABLE 4.19: Table containing the results of Benchmark 2 (variant -*EXPLODED*)

Benchmark 2 - *ATTACHED*

Dataset	**ViT**	**Model**	**Avg. Pos.**	**MRR**	**Recall@1**	**Recall@3**	**Recall@5**	**Recall@10**	**nDCG@3**	**nDCG@5**	**nDCG@10**
N/A	B/32	basic-mini	409.703	0.047	0.02	0.043	0.058	0.082	0.033	0.04	0.047
	L/14	basic-base	333.115	0.075	0.035	0.077	0.1	0.139	0.059	0.069	0.081
	L/14@336px	basic-large	312.496	0.077	0.036	0.077	0.103	0.15	0.06	0.07	0.085
cap-FR-1	L/14	February Finetune	126.18	0.147	0.071	0.148	0.2	0.314	0.115	0.136	0.173
cap-FR-2 & ico-FR		March Finetune	314.846	0.079	0.038	0.075	0.103	0.155	0.059	0.071	0.087
cap-TRI & ico-TRI	B/32	art-mini	179.437	0.13	0.067	0.134	0.177	0.25	0.106	0.123	0.147
	L/14	art-base	**68.956**	**0.244**	**0.132**	**0.272**	**0.352**	**0.472**	**0.213**	**0.246**	**0.285**
	L/14@336px	art-large	75.409	0.214	0.114	0.228	0.308	0.416	0.18	0.213	0.247

TABLE 4.20: Table containing the results of Benchmark 2 (variant -*ATTACHED*)

Benchmark 3

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	11.376	0.546	0.436	0.606	0.673	0.8	0.532	0.559	0.599
	L/14	basic-base	7.976	0.531	0.4	0.612	0.697	0.8	0.521	0.556	0.588
	L/14@336px	basic-large	8.012	0.546	0.4	0.636	0.715	0.812	0.538	0.571	0.603
cap-FR-1	L/14	february_finetuned	4.024	0.714	0.606	0.794	0.848	**0.927**	0.714	0.737	0.763
cap-FR-2 & ico-FR		march_finetuned	3.406	0.752	**0.655**	0.812	0.867	0.909	0.75	**0.772**	0.786
cap-TRI & ico-TRI	B/32	art-mini	5.382	0.669	0.552	0.745	0.83	0.891	0.664	0.699	0.719
	L/14	art-base	3.436	0.748	**0.655**	0.794	**0.873**	0.909	0.739	0.771	0.783
	L/14@336px	art-large	**3.291**	**0.758**	**0.655**	**0.83**	0.855	0.921	**0.762**	**0.772**	**0.795**

TABLE 4.21: Table containing the results of Benchmark 3 in French

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	2.939	0.734	0.606	0.855	0.897	0.939	0.751	0.769	0.782
	L/14	basic-base	2.867	0.779	0.655	0.897	0.915	0.939	0.802	0.809	0.818
	L/14@336px	basic-large	3.079	0.764	0.648	0.842	0.903	0.945	0.768	0.793	0.806
cap-FR-1	L/14	february_finetuned	2.079	0.82	0.727	0.891	0.939	0.97	0.827	0.846	0.855
cap-FR-2 & ico-FR		march_finetuned	1.824	**0.851**	**0.776**	0.909	0.939	**0.988**	0.856	0.868	**0.884**
cap-TRI & ico-TRI	B/32	art-mini	**1.77**	0.828	0.721	**0.945**	**0.958**	**0.988**	0.853	0.858	0.868
	L/14	art-base	1.8	0.85	0.764	0.921	0.952	**0.988**	**0.859**	**0.872**	**0.884**
	L/14@336px	art-large	1.976	0.839	0.752	0.909	0.952	0.976	0.846	0.864	0.872

TABLE 4.22: Table containing the results of Benchmark 3 in English

Dataset	ViT	Model	Avg. Pos.	MRR	Recall@1	Recall@3	Recall@5	Recall@10	nDCG@3	nDCG@5	nDCG@10
N/A	B/32	basic-mini	27.012	0.288	0.17	0.321	0.376	0.503	0.261	0.284	0.326
	L/14	basic-base	16.921	0.395	0.267	0.455	0.521	0.648	0.376	0.404	0.444
	L/14@336px	basic-large	18.588	0.386	0.267	0.424	0.503	0.624	0.36	0.392	0.431
cap-FR-1	L/14	february_finetuned	9.297	0.54	0.412	0.612	0.691	0.782	0.529	0.561	0.591
cap-FR-2 & ico-FR		march_finetuned	7.139	0.564	0.424	0.636	0.758	0.848	0.549	0.598	0.627
cap-TRI & ico-TRI	B/32	art-mini	9.188	0.565	0.436	0.648	0.709	0.788	0.561	0.587	0.613
	L/14	art-base	**3.685**	0.729	0.618	**0.806**	**0.867**	**0.933**	0.73	0.755	**0.776**
	L/14@336px	art-large	3.77	**0.734**	**0.63**	**0.806**	**0.867**	0.909	**0.734**	**0.759**	0.773

TABLE 4.23: Table containing the results of Benchmark 3 in Dutch

Chapter 5

Qualitative Analysis

The models presented in the previous chapter 4 performed remarkably on the benchmarks. This is a good sign but we must also test them qualitatively. This section analyses the top 3 results on interesting queries taken from the *captions* written for the first benchmark. The qualitative analysis is done using queries in French and only considers the 454 artworks used to construct the Benchmark 1 dataset 4.4.1. The captions that were used to query the models **were not used during the finetuning process**.

5.1 All models perform well

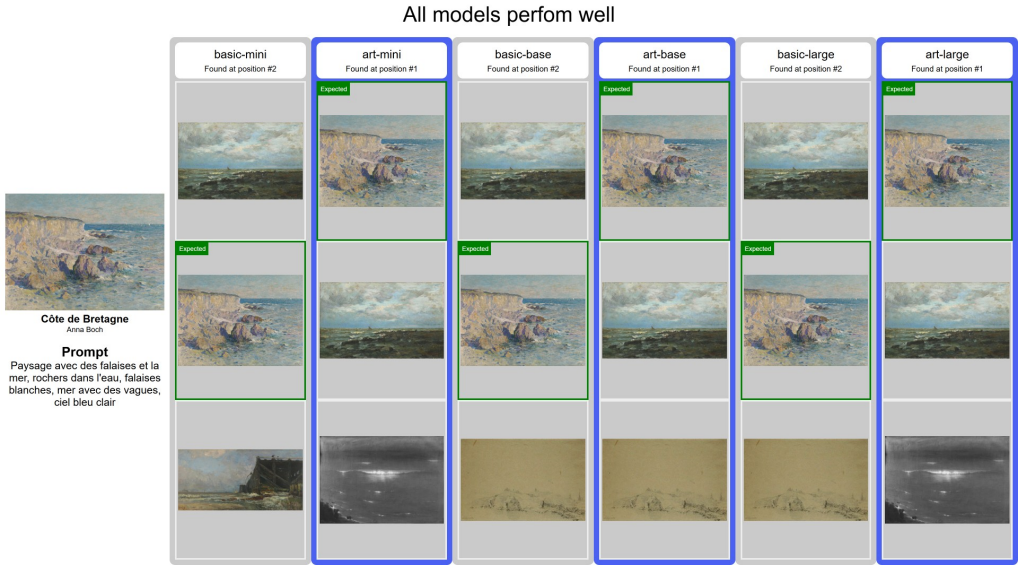


FIGURE 5.1: Qualitative analysis: All models perform well

The figure 5.1 presents a *task* where the baseline models and their finetuned version performed very well. The three baseline models first retrieved artwork is depicting many features that intersect with the caption like the waves, the sea, a light blue sky, ... However, the first retrieved artwork by the baseline models is missing the cliff. The finetuned models improve on this aspect by finding the expected artwork.

5.2 Finetuning helps as lot

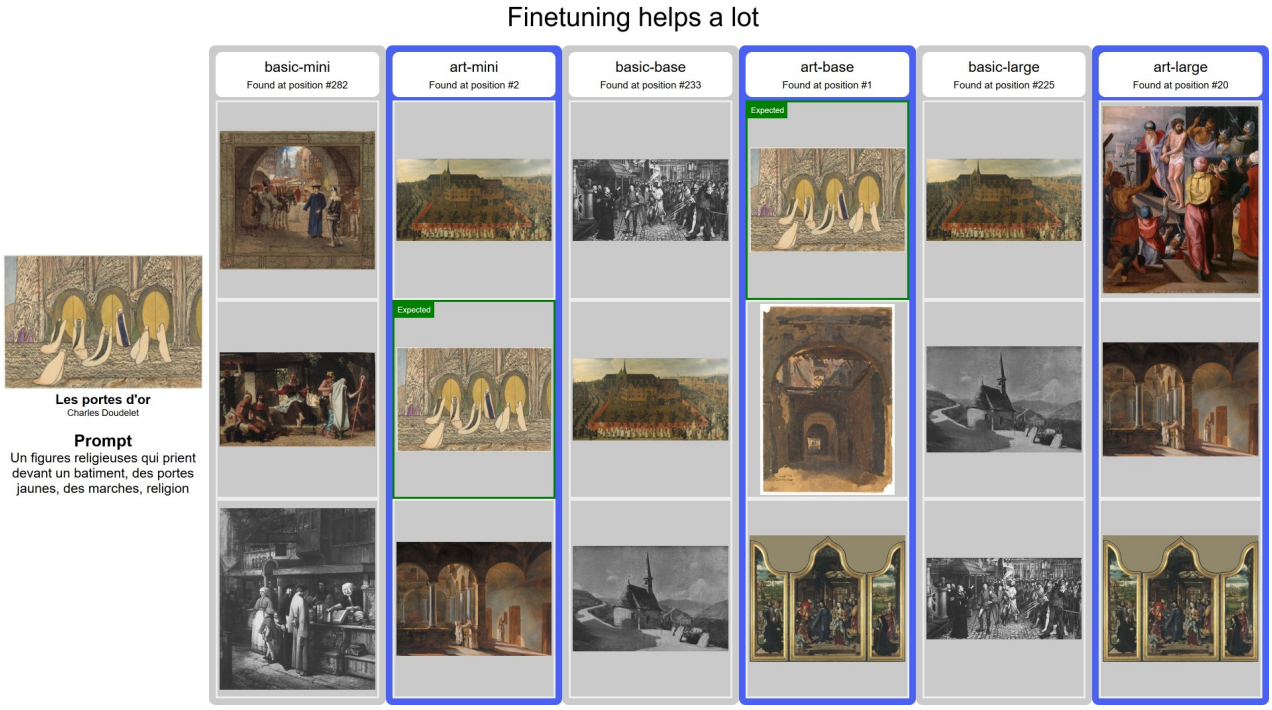


FIGURE 5.2: Qualitative analysis: Finetuning helps as lot

The figure 5.2 presents a *task* where the finetuned models performed significantly better than the baseline models. We believe that the reason the baseline models perform so poorly is the style of the artwork that we are trying to find. Its style is clearly different from the styles of the images that *CLIP* was trained on.

The first result retrieved by *art-mini* could be seen as a valid first retrieval. This is a common observation with image retrieval benchmarks as a caption can describe well more than one image.

The finetuned model *art-large* seems to struggle compared to the other two smaller models. The first artwork it retrieves is very aligned with the concept of religion but not with the other features described in the query.

5.3 Baseline models perform better than finetuning

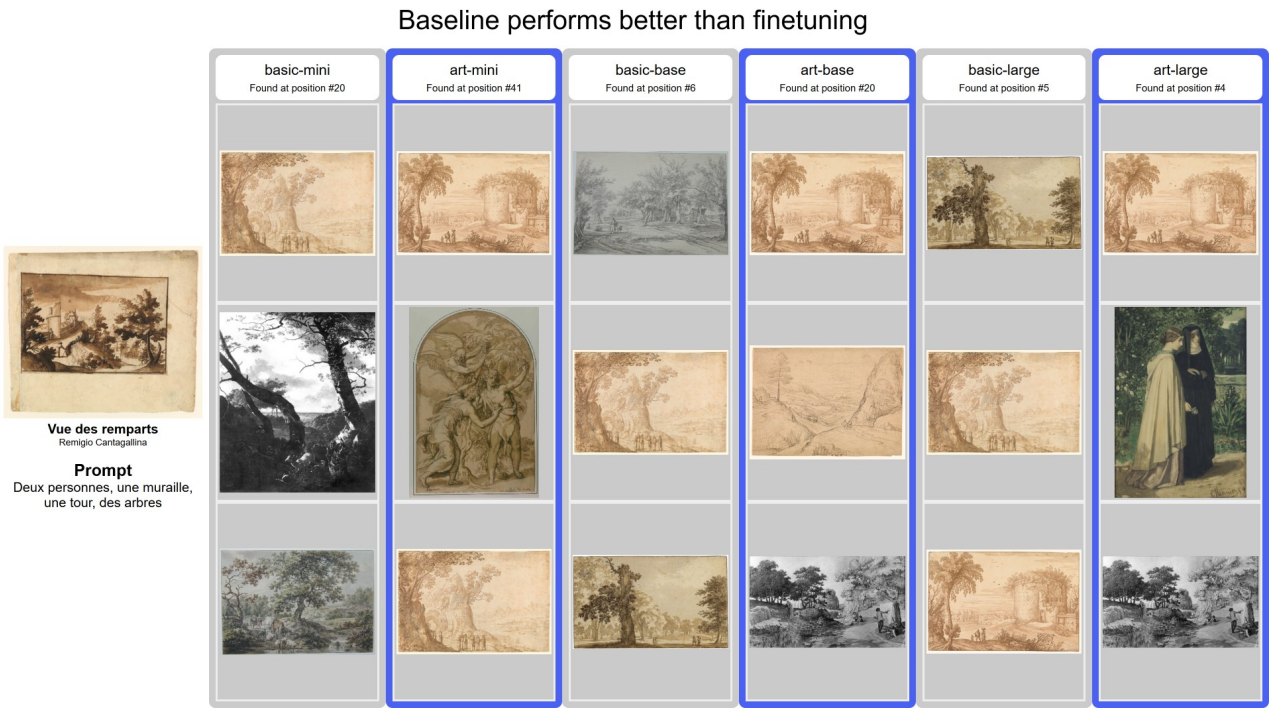


FIGURE 5.3: Qualitative analysis: Baseline models perform better than finetuning

The figure 5.3 presents a *task* where the baseline models performed better than the finetune models. Similarly to the previous section 5.2, the first artwork retrieved by the *art-* models fits the query. We would even argue that their first result fits the query more than the artwork we wrote it for.

That being said, it is surprising to see that for the *art-mini* and *art-base* models, the expected artwork is pushed further from the top result compared to their baseline version. In our opinion, this is mostly due to the artwork being a scan of a drawing with large empty margins around it. It may be interesting to crop and transform these images such that the artistic content takes the full canvas.

5.4 Finetuned models perform badly

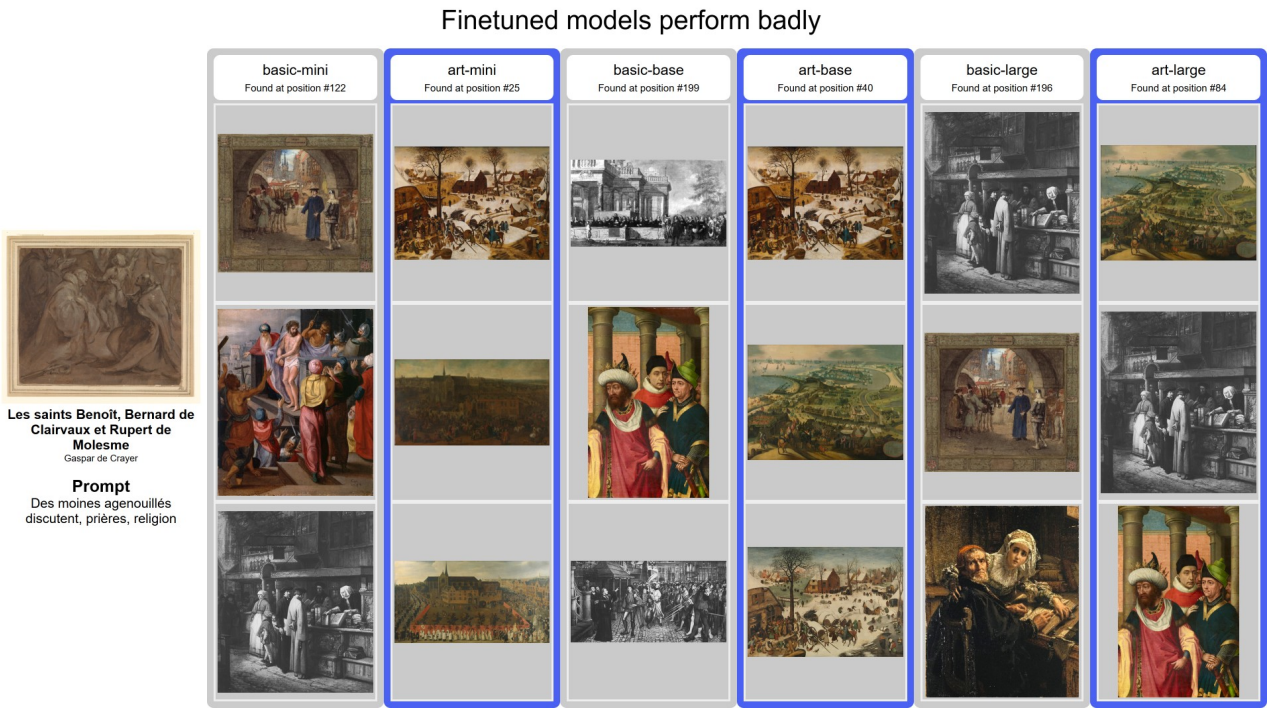


FIGURE 5.4: Qualitative analysis: Finetuned models perform badly

The figure 5.4 presents a *task* where the finetuned models perform poorly (no matter how the baseline models perform). We did not find a possible explanation to this poor performance. The finetuned models retrieve artworks mostly depicting groups of people, often in a city or a village. To investigate this underperformance, we tested paraphrases of this caption using our Search Engine[1]. We managed to get the artwork in the first results by removing from the initial caption the words *religion* and *discutent* (discussing) and by replacing the word *moines* (monks) simply by *personnes* (people). We see two conclusions from this experiment. First, having many words in the query that are in the semantic field of *religion* (monks, religion, pray) makes its embeddings strongly aligned with artworks depicting churches, often seen in artworks depicting cities or villages. Secondly, the word *discutent* (discussing) aligns a lot the embedding of the query with artworks depicting many people. These two effects make the model retrieve artworks depicting a group of people in a city context, which is very misaligned with an artwork depicting mainly only two people on a brown background.

[1]The queries were made in French and using *art-base*

5.5 High variance between finetuned models and worst performing *task* for *art-mini*

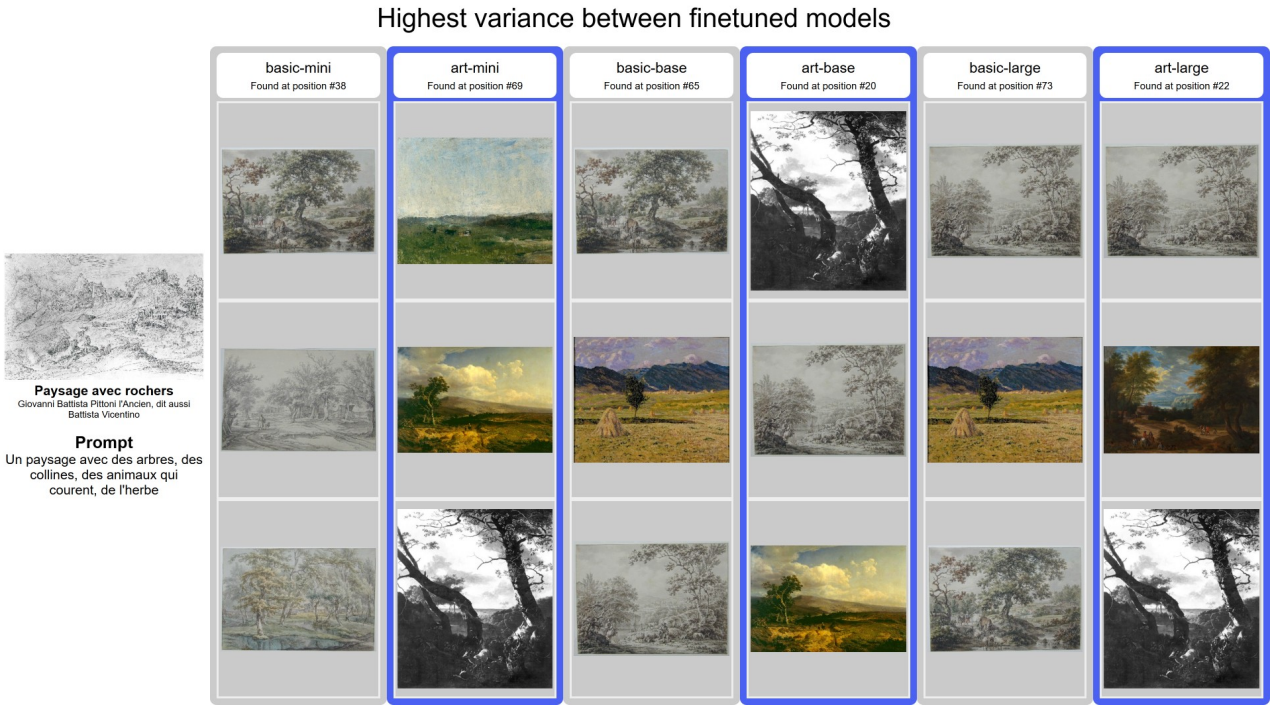


FIGURE 5.5: Qualitative analysis: High variance between finetuned models and worst performing *task* for *art-mini*

The figure 5.5 presents a *task* where the *art-* models performed very differently from one another. The retrieved artworks for all models were very aligned with the query. The style of the artwork for which the query was written is quite specific and the low resolution of the image available makes this *task* very difficult for the models. This is a good example of a *task* that suggests that our models are not performing well when in reality, the style and the resolution of the expected artwork might make it impossible to perfectly fulfil this *task*.

Coincidentally, this *task* is the worst-performing *task* for the *art-mini* model.

5.6 Worst performing *task* for *art-base*

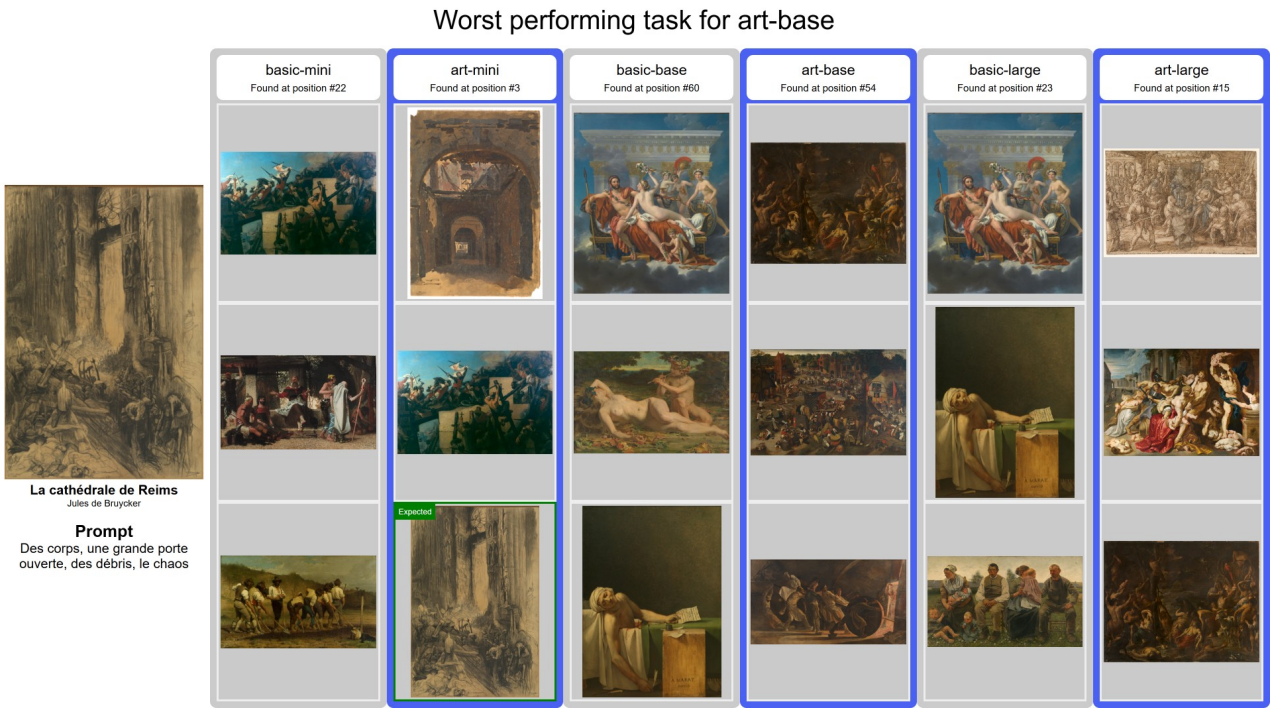


FIGURE 5.6: Qualitative analysis: Worst performing *task* for *art-base*

The figure 5.6 depicts the worst performing *task* for our best model, *art-base*. The artworks it retrieves are not especially bad; we would even argue that *art-mini* is returning the worst first 3 artworks. The query and the artworks returned seem to be aligned in the concepts they describe/depict. We therefore suspect that the *art-base* model *sees* something quite different than what we have seen when writing this query.

We ran a little experiment by querying our Search Engine[2] with the word *débris* which means *rubble* in English. It surprisingly retrieved artworks with groups of people. When querying with the word *rubble*, we found the artwork presented in the left of figure 5.6 in the first results. It seems that the embedding for the word *débris* is very aligned with the embedding of the word *people*.

[2]With the model *art-base*.

5.7 Worst performing *task* for *art-large*

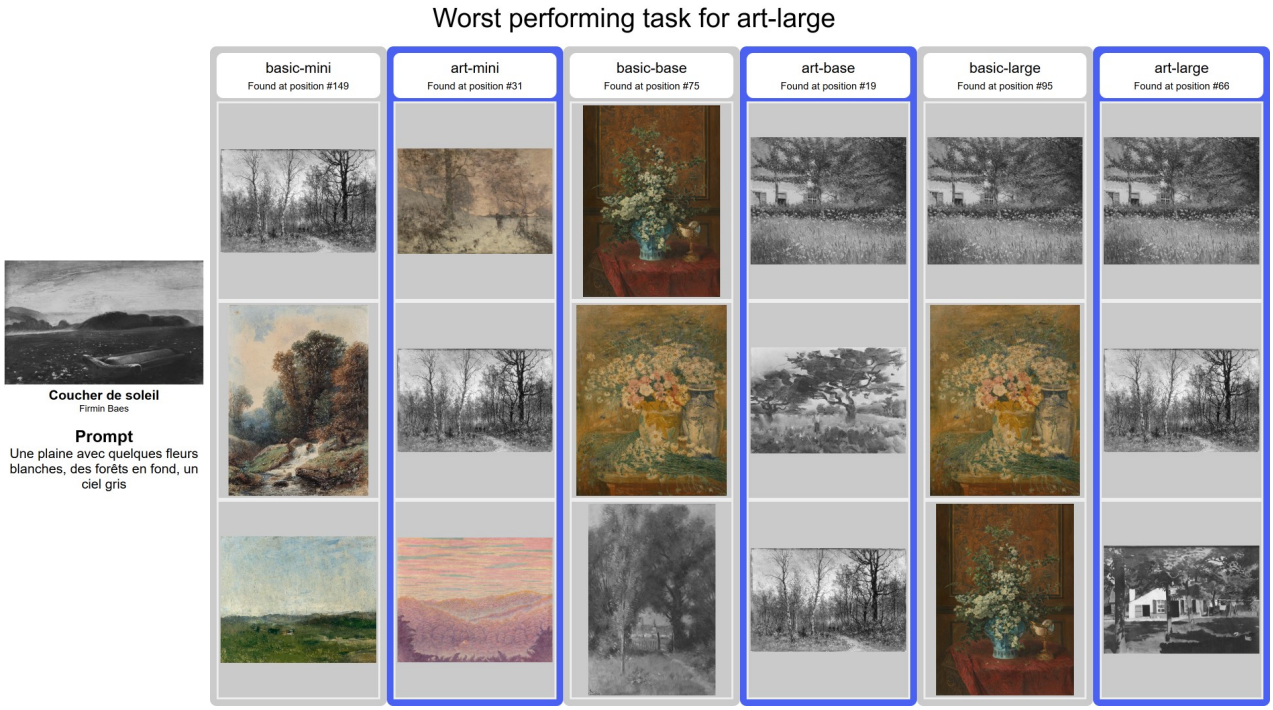


FIGURE 5.7: Qualitative analysis: Worst performing *task* for *art-large*

The figure 5.7 depicts the worst performing *task* for *art-large*. Once again, the retrieved artworks somewhat fits the query we gave the models. The *basic-base* and *basic-large* models seem to focus on the *fleurs* (flowers) word in the query, their finetuned version seem to focus on the *gris* (grey) word.

Chapter 6

Making the Search Engine

Our improved models, *art-mini*, *art-base* and *art-large*, significantly improve our ability to query the *RMFAB* digital gallery in French, English, and Dutch, as demonstrated in chapter *Qualitative Analysis* 5. However, a robust model is only one component of an effective exploration tool. This section details the development of the finished search engine prototype, building upon the version introduced in Section 4.3.2.

6.1 Current search engine

The current search engine used by the RMFAB is called *Fabritius*. It is already a fairly powerful search engine allowing to find artworks by their title, the artist, the inventory number and the subject(s) present[1]. A user can do a simple search with just a single textual input with an auto complete feature to search on all fields at once. Users can also conduct more complex searches by combining up to three conditions using **AND**, **OR**, or **EXCEPT** operators. These conditions can be applied to specific fields such as **All fields**, **Artist**, **Title**, or **Subject(s) present**. Additionally, complex searches can be refined by filtering results based on criteria like **period**, **materials**, and **type of artworks**. The figure H.5 presents screenshots of the *Fabritius* search engine.

Our development will take into account this set of features as we believe that it is necessary to offer **at least** what's already offered by *Fabritius*.

6.2 Technical overview of the project

6.2.1 Managing the data

The subset of the digital dataset of the *RMFAB* that I had access to was given to me in a format that is not suitable for a modern, efficient and complex application. It is therefore necessary to transform this dataset in a clean and accessible way format. The diagram 6.1 describes the database schema used in this project.

In total, 5301 artworks made by 870 artists were parsed. Each artwork has 3 embeddings, one per *art-* model size. The *subject matter* fields are also stored for the artworks having one. For the

[1] It seems like it uses the *subject matter* fields or a subset of them as an index for this querying method.

subject matter fields following the *Thesaurus Garnier* format, two versions are stored: a flattened (list) version and a tree version (stored using *PostgreSQL JSONB* format)[2]

The database that was chosen to store our data was *PostgreSQL* with the *pgvector* addon allowing to store the *CLIP* embeddings of the images and the keywords and allowing us to quickly retrieve the closest ones based on their *cosine similarity* with the a provided embedding.

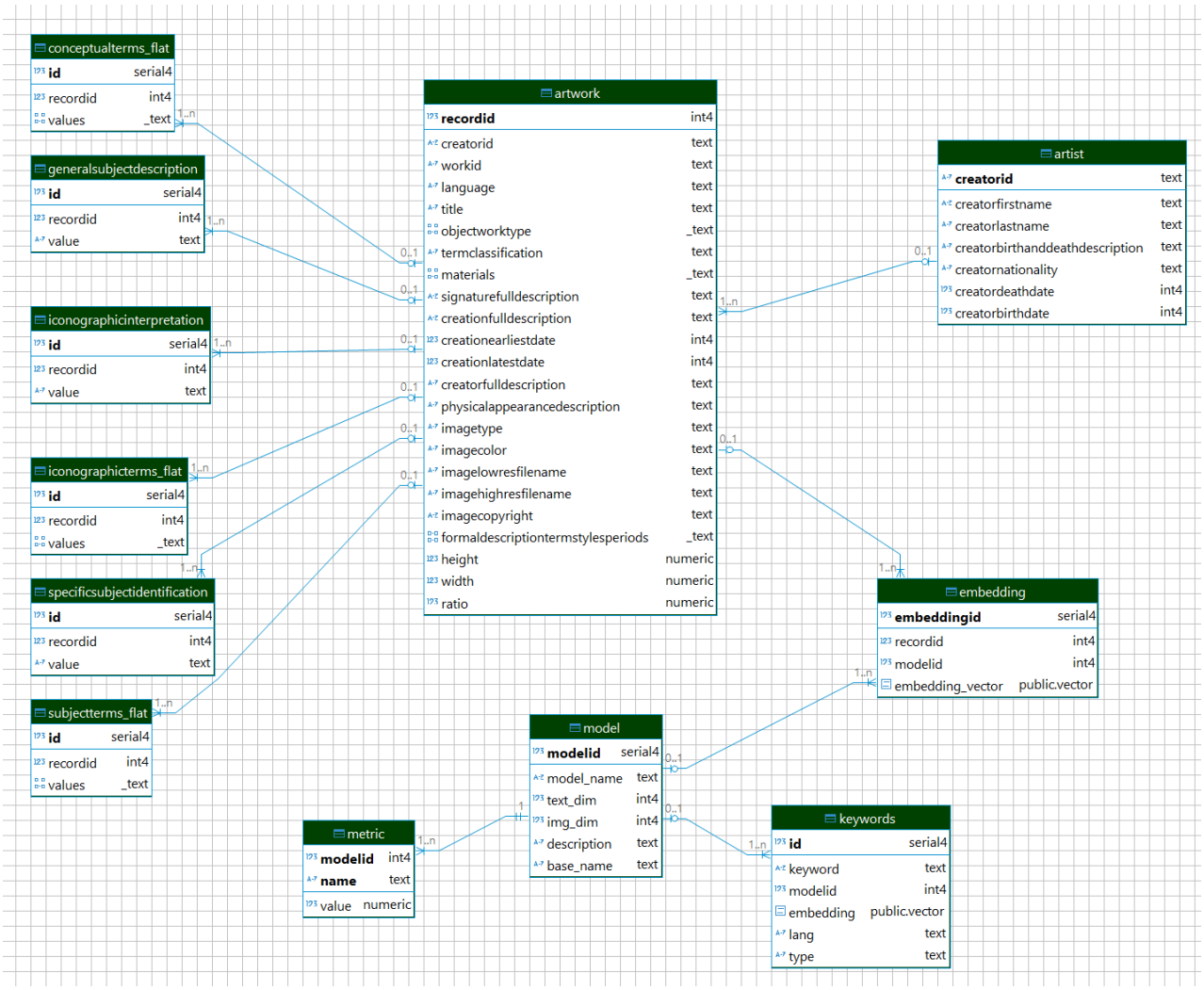


FIGURE 6.1: Crow's foot notation of the database

[2]The *JSONB* format has not been used in the final project due to time constraints.

6.2.2 Back-end

The back-end was done using *Python* with *Fastapi* to deliver the endpoints of the *API*. The endpoints of the *API* are:

1. **/api/artwork/<int:record_id>/similar** (POST)
   - *Description*: Retrieves artworks similar to a given artwork identifiable by its *record_id*.
   - *URL Arguments*:
     (a) *record_id*: The *record_id* of the artwork.
   - *JSON Body Arguments*:
     (a) *page*: The page number for results.
     (b) *page_size*: The number of results per page.
     (c) *keep_original_record*: Whether to include the original record in the results.
     (d) *model_name*: The name of the model to use.
2. **/api/artwork/<int:record_id>** (GET)
   - *Description*: Retrieves detailed information for a specific artwork by its *record_id*.
   - *URL Arguments*:
     (a) *record_id*: The *record_id* of the artwork.
3. **/api/artwork/<int:record_id>/image** (GET)
   - *Description*: Serves the image file for a given artwork *record_id*.
   - *URL Arguments*:
     (a) *record_id*: The *record_id* of the artwork.
4. **/api/artist/<string:creator_id>** (GET)
   - *Description*: Retrieves information about an artist using their *creator_id*.
   - *URL Arguments*:
     (a) *creator_id*: The *creator_id* of the artist.
5. **/api/query** (POST)
   - *Description*: Executes a complex query on the artwork database with hard and soft constraints.
   - *JSON Body Arguments*:
     (a) *hard_constraints*: A list of hard constraints.
     (b) *soft_constraints*: A list of soft constraints.
     (c) *model_name*: The name of the model to use.
     (d) *page*: The page number for results.
     (e) *page_size*: The number of results per page.

(f) *version*: The query version ("classic", "power", or "rocchio").
(g) *rocchio_k*: Parameter for Rocchio algorithm.
(h) *rocchio_scale*: Scale factor for Rocchio algorithm.

6. **/api/collection/augment** (POST)
   - *Description*: Augments a collection of artworks based on specified *recordIDs* and augmentation method.
   - *JSON Body Arguments*:
     (a) *recordIDs*: A list of artwork *record_id*s.
     (b) *method*: The augmentation method ("convex_fill" or "shortest_path").
     (c) *numberOfImages*: (For "convex_fill" method) Number of images to add.
     (d) *similarityThreshold*: (For "convex_fill" method) Similarity threshold.
     (e) *decayRate*: (For "convex_fill" method) Decay rate.
     (f) *patience*: (For "convex_fill" method) Patience parameter.
     (g) *model_name*: The name of the model to use.
7. **/api/collection/sort_by_similarity** (POST)
   - *Description*: Sorts a given list of *recordIDs* by similarity using a specified model.
   - *JSON Body Arguments*:
     (a) *recordIDs*: A list of artwork *record_id*s to sort.
     (b) *model_name*: The name of the model to use for similarity.
8. **/api/collection/path_from_two_terms** (POST)
   - *Description*: Finds a path between two terms within a collection of artworks.
   - *JSON Body Arguments*:
     (a) *recordIDs*: A list of artwork *record_id*s to consider for the path.
     (b) *model_name*: The name of the model to use.
     (c) *term1*: The first term.
     (d) *term2*: The second term.
9. **/api/autocomplete** (POST)
   - *Description*: Provides autocomplete suggestions for a given prefix and column.
   - *JSON Body Arguments*:
     (a) *prefix*: The prefix to autocomplete.
     (b) *column*: The column to search in.
10. **/api/get_settings_infos** (GET)
    - *Description*: Retrieves various settings and informational details about the API, including models, keywords, methods, and parameter bounds.

The enumeration above is a summary of the endpoints, the section 6.3 will detail their exact usage and inner workings.

As all the embeddings from the artworks were precomputed per model size and stored in the *SQL* database, our server does not have to encode any image which is very computation-intensive. But as we would like to offer the user the possibility to query by any textual description he wishes to enter, the models still have to run to encode the textual prompts received. Running the three models on *GPU* requires about 5.5*GB* or *VRAM*. Of course, the models can run on also on *CPU* since the inference is light enough for textual encoding. If the hardware the project is running on does not allow to run multiple models at once, the back-end can be configured such that only one model (the *art-mini* for example) is available.

An even less computational-intensive version of our search engine was discussed during the development. The user could search using a large set of keywords but could not enter his own prompt. This would still allow someone to explore the large digital gallery of the *RMFAB* without the need for the models to run on the server as the embeddings of the keywords and the artworks could be precomputed and stored in the database. This option may be better suited to a general public search engine to reduce the server costs in a high traffic environment.

6.2.3 Front-end

The frontend interface is accessed through a webpage. The chosen stack is **React** with **TypeScript** (and **Vite**). This stack was selected primarily due to the development team's existing experience, but it could be interchanged with other frameworks. This flexibility is largely thanks to the backend being exposed via an **API**, decoupling the frontend and backend architectures.

6.3 Interface and features

The search engine that was developed for the *RMFAB* contains many interfaces and features. A user can search through the database of artworks, get the details of an artwork or an artist, add artworks to a collection and so on. This section will present the features implemented using screenshots and example usage.

6.3.1 Overall interface

When opening the website, the user will find a screen divided into two sections (see figure 6.2). The first section on the left is the *control panel*, it contains the control for creating an query, the collections the user has created, the error logs and the settings of the search engine. The second section on the right is the *tabs container*. As its name hints, it contains the *tabs*. There are several types of *tabs*:

1. *Search results*: The resulting artworks of a query are shown in a masonry fashion.
2. *Artwork profile*: The profile of an artwork containing its details and the similar artworks according to the model
3. *Artist profile*: The profile of an artist containing its details and its artworks

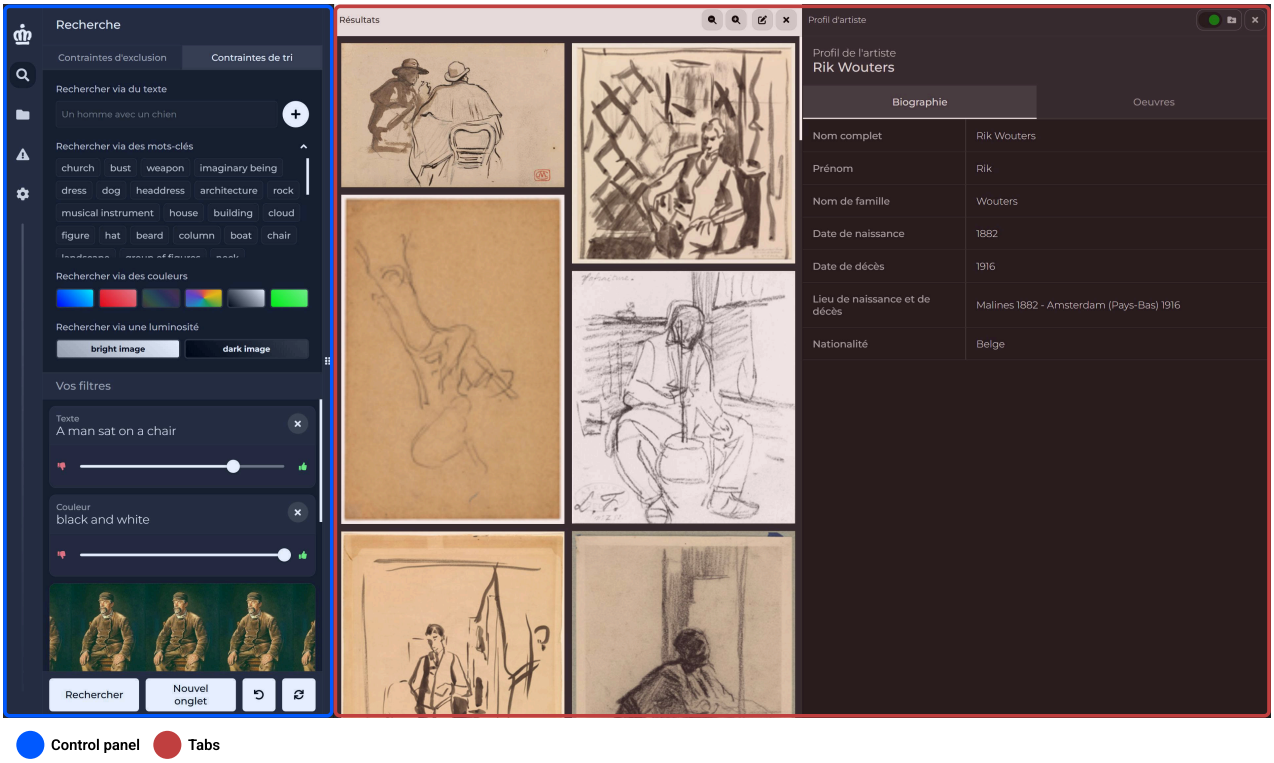


FIGURE 6.2: Main components of the Search Engine

4. *Collection profile*: The artworks added to a collection.

6.3.2 Control panel

Hard sub-queries

A query sent to the *API* can be composed of many sub-queries. There are two types of sub-queries, *soft* and *hard* sub-queries. Let us first explain *hard* sub-queries as it most resemble how a classic search engine like *Fabritius* operates.

A *hard* sub-query is a constraint put on the dataset of artworks. It *removes* from the results the artworks not respecting the constraint. The user can create infinity complex *hard* sub-queries by adding *Blocks* combined with *AND* and *OR* operator. In front of each *Block*, the user can selected modifiers:

1. *NOT*: The artworks **NOT** matching this constraint will be kept instead of the ones matching it
2. *Exact match*: The column must contain *exactly* the same entry (not case-sensitive)
3. *All*: The column must contain *all* the entries of the user (for querying on array-like columns)
4. *Case sensitive*: The column must contain the value of the input with case-sensitivity

5. *Include NULL*: Include the artworks that do not posses a value for the selected column

There are 4 *Blocks* available to construct *hard* sub-queries:

1. *EQUAL*: The chosen column must be equal (in part or fully) to the input by the user. For example searching all artworks containing a specific word in their title.
2. *BETWEEN*: The chosen column must $\geq$ to a minimum value and $\leq$ to a maximum value. For example searching all artworks made between 1910 and 1920.
3. *INCLUDES*: The chosen column must contain some or all the entries by the user. For example, searching all artworks having the terms *maison* (house) and *arbre* (tree) in their *subject matter* fields
4. *GROUP*: A group groups *Blocks* so that their combined constraints (combined by *OR* and *AND* **in** their group) is treated as a *Block*. This allows to do intricate *hard* sub-queries.

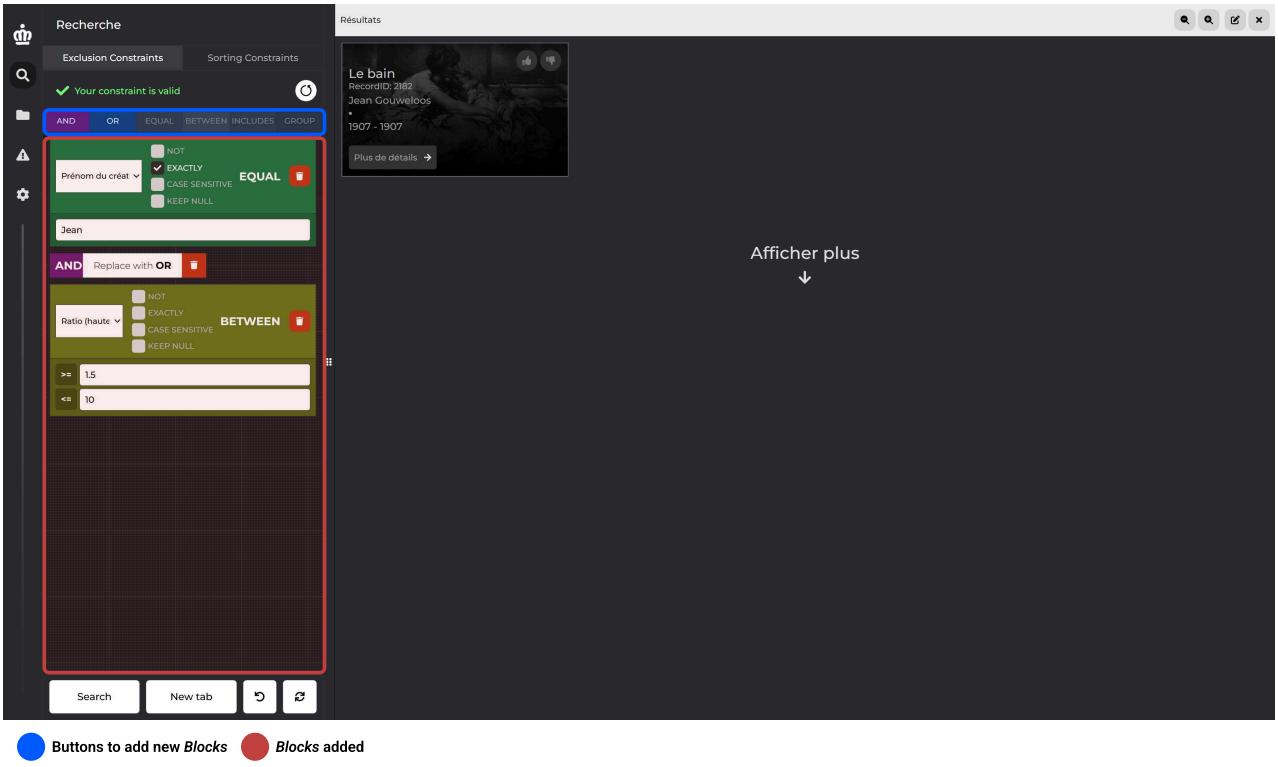


FIGURE 6.3: *Hard* sub-queries creator

The figure 6.3 presents a screenshot of the *hard* sub-queries creator. By adding *Blocks*, one can made very complex queries to restrict the artworks taken into account by the *CLIP* model. In the screenshot provided, the *hard* sub-queries were:

1. The *first name* of the artist must be *exactly Jean*
2. *AND*
3. The *width to height ratio* of the artwork must $1.5 \leq ratio \geq 10.0$ (a landscape image)

The user can add more *Blocks* if he wishes using the top *Blocks* selector highlighted in blue on the screenshot. The web UI automatically restrict the *Blocks* that the user can add to avoid making illogical sub-queries. When the back-end receives the *hard* sub-queries, it transforms them into *SQL* clauses. Only the restricted subset of artworks is then taken into account for the *soft* sub-queries.

Usage Querying using this method is **very** powerful as the user can see **exactly** what respects his constraints. But it can be a little hard to use and it requires some learning. As explained in the section 2.2.1 analyzing the *RMFAB* data, the *subject matter* fields (especially the *Subject Terms*) describes some of the objects present in an image. Not all artworks have them but for the ones with it, the user can click on the objects he wishes to see directly in the search results *tab* as shown in figure 6.4. This action automatically create a *hard* sub-queries greatly facilitating the querying process.

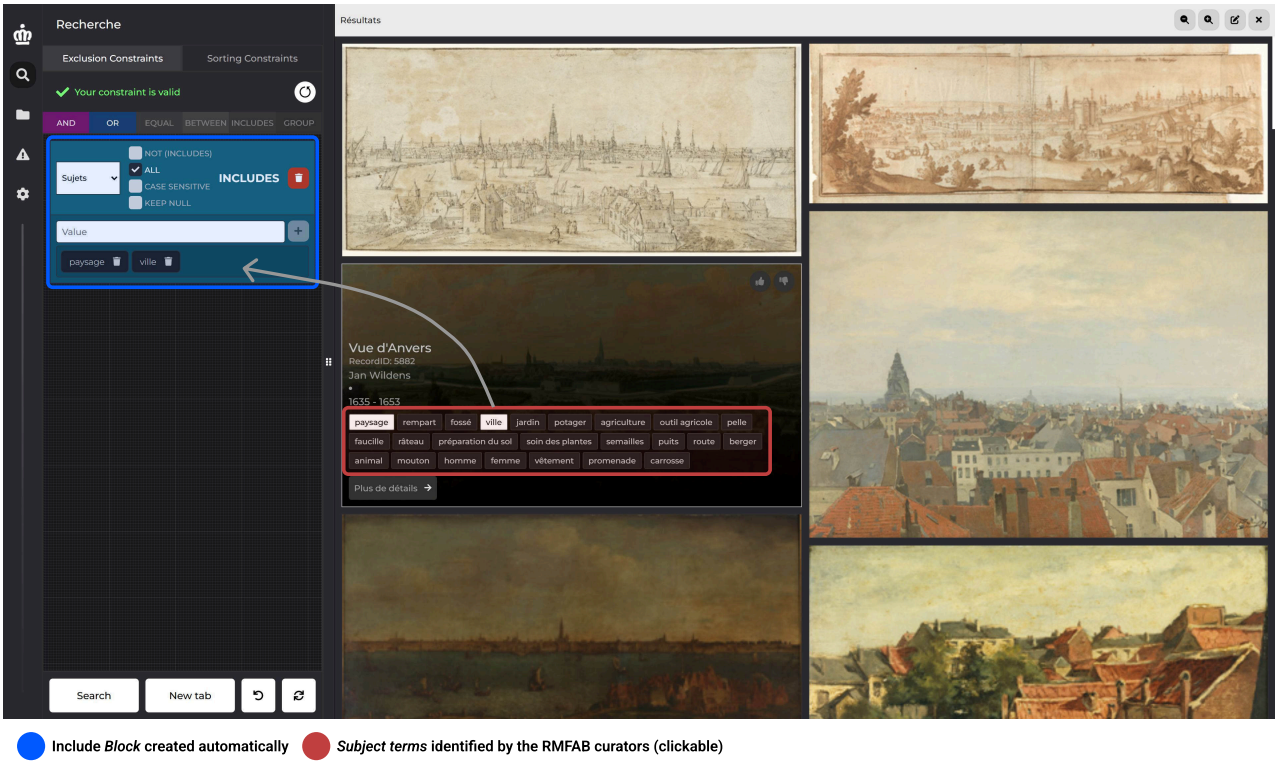


FIGURE 6.4: Automatic creating of *hard* sub-queries based on the *Subject Terms*

We have found this feature very useful when testing the search engine as we often first made a query with a textual description, then by clicking on the objects we wanted to see, we refined our search to find an artwork matching the idea we had in mind.

Soft sub-queries

A *soft* sub-query has a **sorting** impact of the subset of artworks respecting the *hard* sub-queries. The artworks are **ordered** by their cosine similarity with the embedding of the combined *soft* sub-queries.

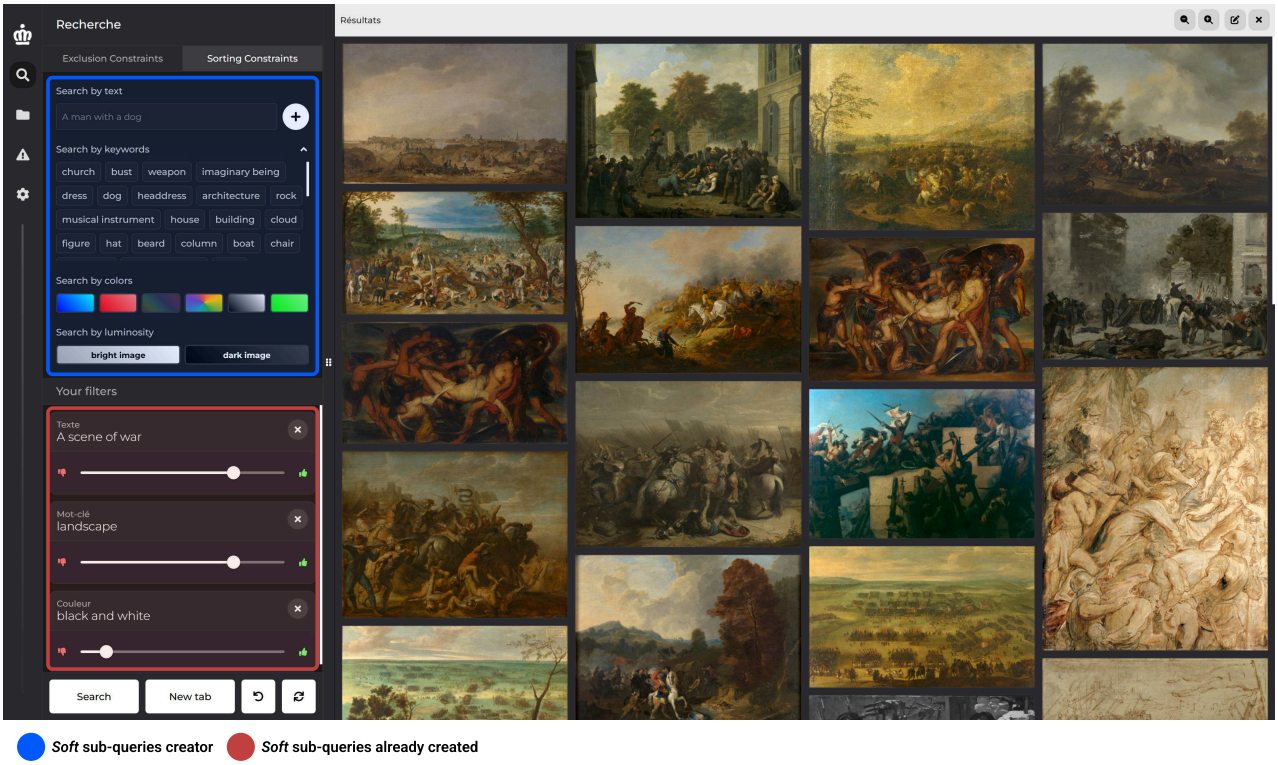


FIGURE 6.5: *Soft* sub-queries interface

The screenshot presented in figure 6.5 depicts a search made using three *soft* sub-queries:

1. The textual prompt *A scene of war* with a **positive** weight (the user wants to see that)
2. The keywords *landscape* with a **positive** weight
3. The color *black and white* with a **negative** weight (the user does not want to see that)

Precomputed *soft* sub-queries The user can quickly add *soft* sub-queries like *keywords*, *colors* or *luminosity*. This is an intuitive way to interact with the search engine. Yet as observed before, mixing queries together instead of using a single prompt offer worse results 4.5.2. We would therefore advise the user to always first try to describe his desired artwork in a single prompt as the textual encoder of *CLIP* will *understand* the prompt better than naively mixing embeddings of smaller

prompts. If the user has a textual prompt that overflows the token limitation of the *CLIP* model, then we advise him to tweak the weights (represented by the thumbs up and down on the figure 6.5) to get his desired results. The final embedding that will be used to order the artworks will be the normalized sum of the *soft* sub-queries multiplied by the power of their respective weights (with the sign added back).

The choice of using the power of the weights instead of the weights themselves came from letting colleagues test the search engine. We noticed that some users were frustrated when a term they strongly disliked still showed in the results. A simple yet effective way to reduce this sensation of incapacity was to make the weight scale not linearly but exponentially. Doubling a term's dislike by sliding the slide to the right means the model will square its associated weight, causing it to be penalized four times as much (the sign is kept after the squaring).

Using the algorithm *Rocchio* The user can also like or dislike an artwork present in the results *tab*. This is a very direct way to tell the model what is a good answer and what is not. Initially, the same weighting system as with the terms was used, but we had the hypothesis that if *CLIP* is well trained as in our case, disliking an image would also mean disliking the images that look like it. Coincidentally, this algorithm is very similar to the *Rocchio* algorithm developed in the 1960's at *Cornell University* [46] for the *SMART Information Retrieval System*.

Our implementation of *Rocchio*'s algorithm for *CLIP* embeddings work as such. Let S be the embedding of an artwork the user likes or dislikes with weight w_S, let α be a scaling factor controlling the power of *Rocchio* and let k be the maximum number of neighbours of S we take into account. We first get the embeddings of the k closest neighbours of S: E_i for $i \in [1, k]$. Then we can compute the new embedding S' of the *soft* sub-query liking or disliking S with the following formula:

$$S' = w_S * (S + (\sum_{i=1}^{k} (\alpha * max(cosine\_similarity(S, E_i), 0)^2)) \tag{6.1}$$

The $max(\dots)$ is required since we do not want to influence the search in a direction and its opposite. If the neighbour i is dissimilar ($cosine\_similarity(S, E_i) < 0$), then without clipping the cosine similarity to 0 the algorithm would consider the opposite of S as a good alternative representation of S. The default values that we used were $k = 5$ and $\alpha = 1.0$. These values can be tweaked in the settings.

Collections

The features we presented in the previous sections focus mainly on searching for artworks. This exploration aspect is very useful for the curators of the museums but could also benefit the general public. As explained in section 2.1, the activities done by the *Musée sur Mesure* are organized in two phases. The first phase often done at the facilities of the patients can be assisted with our search engine as it allows to explore the digital gallery by prompting what the patients would want to see.

To build upon this potential, we have decided to add *Collections* to our tool. A *Collection* is simply a collection of images that the user has decided to group together. For example it could be a thematic

set of wedding paintings to trigger wedding memories in an Alzheimer's patient. A *Collection* can also tell a story, for example from painting of snowy landscapes to paintings or sunny beaches. It is a creative tool to present the RMFAB digital library.

The figure 6.6 presents a screenshot of the profile of a *Collection* named *Portrait of old man*, it contains 5 artworks manually added. On the upper right of the screenshot, some buttons are highlighted. These buttons help to interact with the artworks from the *Collection*.

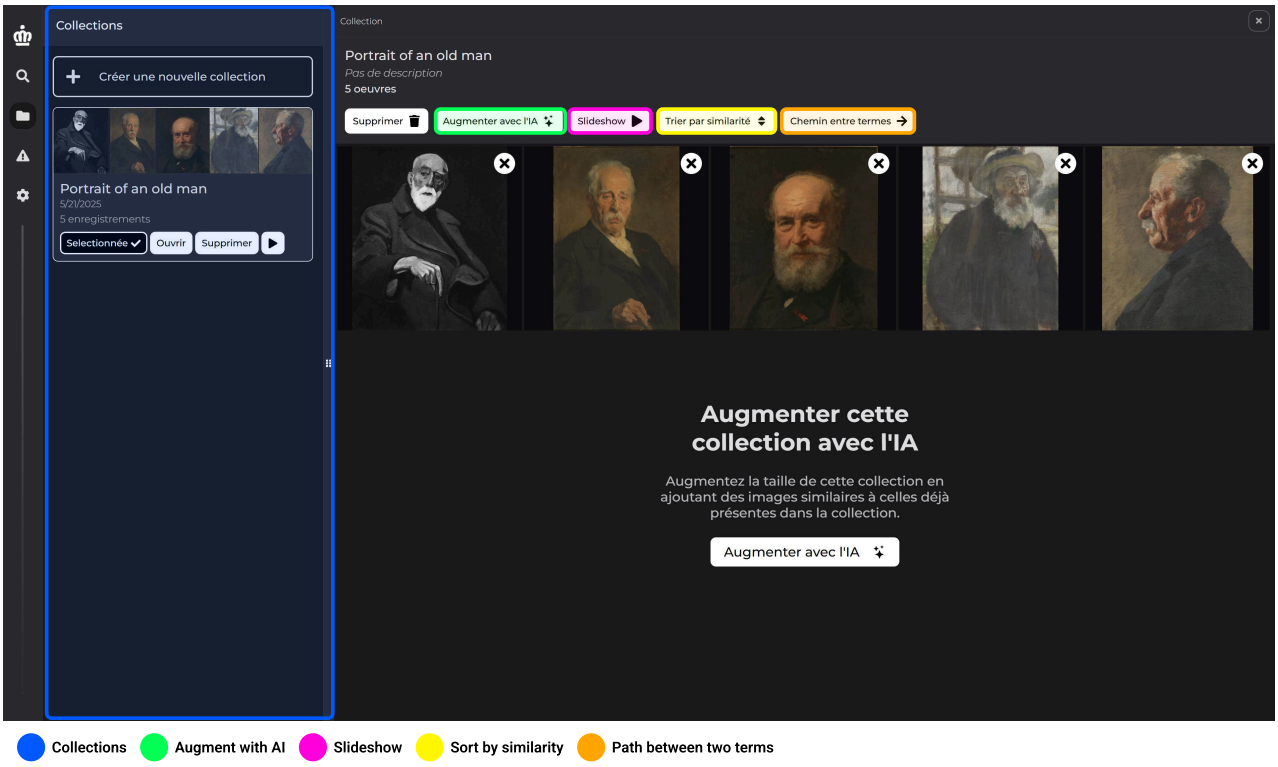


FIGURE 6.6: Screenshot of *Collection* tab and *Collection* profile

Augmenting a collection The first button highlighted in green in 6.6 opens up a modal to augment the collection using *AI*. Augmenting a *Collection* is simply finding new artworks that are relevant according to our *CLIP* model. This augmentation can be done using two methods.

Augmenting a collection - *Convex Fill* The first method is called *Convex Fill*. It is the simplest method out of the two and runs the fastest. It is an iterative method that iterates until it has found N new artworks to add to the collection. Let $P = \{N, \tau, \delta, K\}$ be the set of input parameters, where:

- N is the **number of images** we would like to add to the *Collection*
- τ is **similarity threshold**, ideally the minimum cosine similarity between the embeddings e_1 and e_2 we use to get the centroid
- δ is the **decay rate** at which the similarity threshold decays if we don't find good enough e_1 and e_2
- K is the **maximum number of times** we will try to find new e_1 and e_2 in case our current candidates are not accepted (= maximum patience)

Let $A_{init} = \{a_1, a_2, \ldots, a_m\}$ be the set of the m artworks currently in the *Collection*
Let A_{all} be the set of all artworks
Let $E(a_i)$ be the function giving us the *CLIP* embedding of the artwork a_i
We first get the artworks that may be added to the *Collection*:

- $A_{candidates} = A_{all} \setminus A_{init}$
- $A_{augmented} = \emptyset$ (the new artworks)
- $k_{curr} = 0$ (current patience counter)
- $\tau_{curr} = \tau$ (current cosine similarity threshold)

Until the $A_{augmented}| < N$ the algorithm select randomly two artworks from A_{init} called a_1 and a_2 (they can be the same). It recovers their respective *CLIP* embeddings e_1 and e_2 and it computes their cosine similarity $cosine\_similarity(e_1, e_2)$. If is it above τ_{curr} or if $k_{curr} \geq K$ it will use these two embeddings for the rest of the algorithm, else it will increment k_{curr} and multiply τ_{curr} by δ. Once two candidates e_1 and e_2 pass this step, their centroid $C = \frac{e_1+e_2}{2}$ is computed and the closest artwork a_c to it is added to $A_{augmented}$. Of course, a_c is removed from $A_{candidates}$.

By applying the *Convex Fill* method to our example in 6.6 we got the result in figure 6.7[3].

Augmenting a collection - *Shortest Path* The second method, *Shortest Path*, potentially yields superior results by using a **Traveling Salesperson Problem (TSP) path**. This path connects the embeddings of the N artworks within the *Collection*. Subsequently, the algorithm identifies the $N-1$ artworks not already in or added to the *Collection* and that are the closest to the $N-1$ centroids formed by adjacent nodes along this computed path. It is implemented with the TSP approximation function from *NetworkX* [16].

By applying the *Shortest Path* method to our example in 6.6 we got the result in figure 6.8.

[3]Since this algorithm is based on random sampling of e_1 and e_2 this results is obviously not deterministic.

FIGURE 6.7: Artworks found by Convex Fill (number of images=5, similarity threshold=0.8, decay rate=0.95, maximum number of times=20)

Slideshow The second button, highlighted in purple in 6.6, opens a modal that allows users to configure and launch a fullscreen slideshow of the *Collection*. Within this modal, users can enable or disable automatic sliding, set the time interval between artworks, and choose to run the slideshow indefinitely. Once the launch button is pressed, the artworks from the *Collection* are presented sequentially, following the order displayed in the collection profile.

Sorting the collection In the following of the previous comment about presenting the slideshow in the order of the *Collection*. The user must have options to his *Collection*. The first option, accessible via the button highlighted in yellow in 6.6 sorts the artworks of the *Collection* by computing the shortest path (in the same manner as when augmenting a collection). By sorting with this first option our example in 6.6 we get the order presented in figure 6.9.

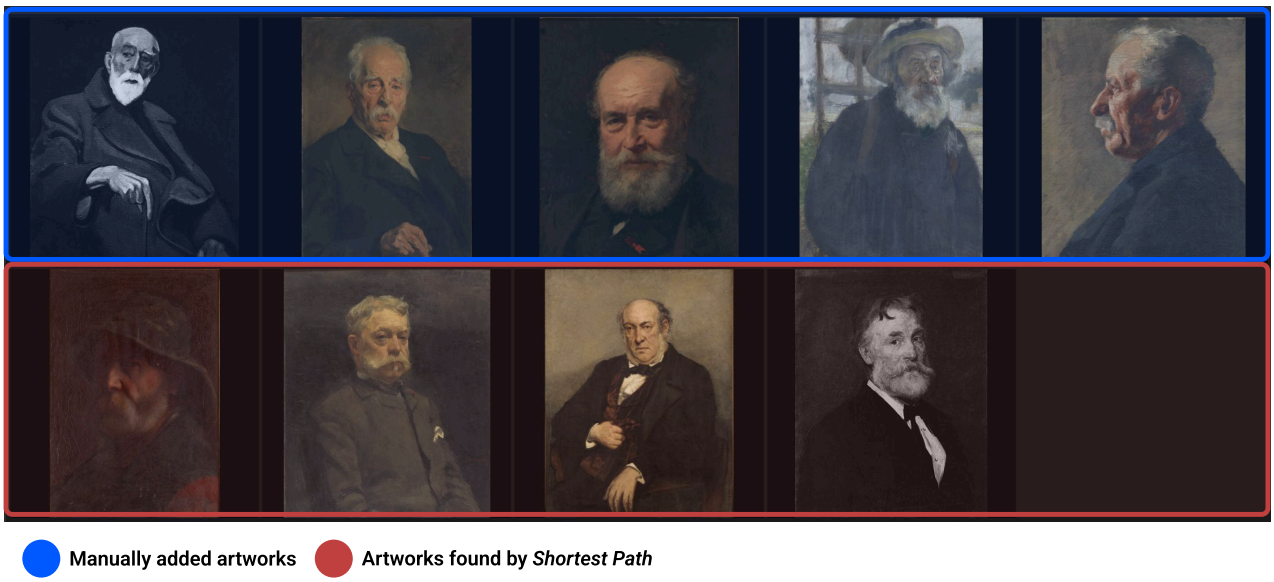


FIGURE 6.8: Artworks found by Shortest Path

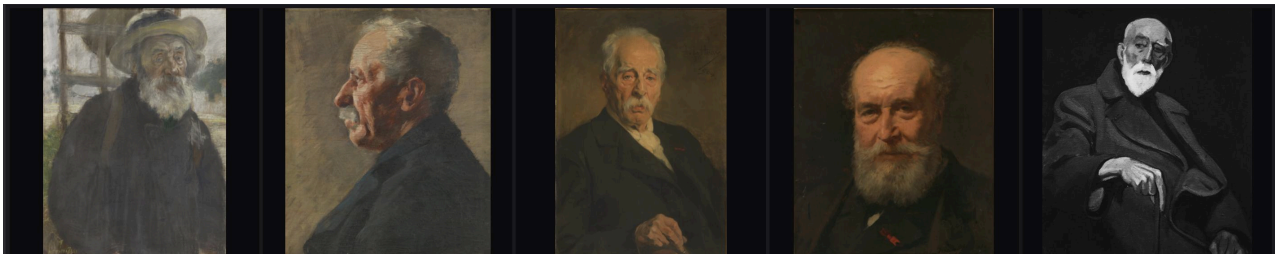


FIGURE 6.9: Sorting of the artworks using the *Sort by similarity* button

The second option is accessible via the button highlighted in orange in 6.6. Upon pressing it, the user is presented with a modal where he can enter two textual prompts. The first prompt represents the *beginning* of his *Collection* and the second one the *end*. The idea of this sorting option is to project the embeddings of the artworks in the *Collection* on the line defined by the embeddings of these two prompts. Let t_1 be the embedding of the first prompt and t_2 be the embedding of the second prompt. For an artwork i with embedding e_i in the *Collection*, we calculate its scalar projection α_i onto the vector $v = t_2 - t_1$. For the artwork i, The formula for α_i is given by:

$$\alpha_i = \frac{(e_i - t_1) \cdot v}{v \cdot v} \tag{6.2}$$

This α_i value indicates the position of the artwork's embedding along the semantic axis defined by the two prompts[4]. The artworks are then sorted based on their α values.

By sorting our example in 6.6 with this second option with the first prompt being *A man with a beard* and the second prompt being *A man without a beard*, we get the order presented in figure 6.10.

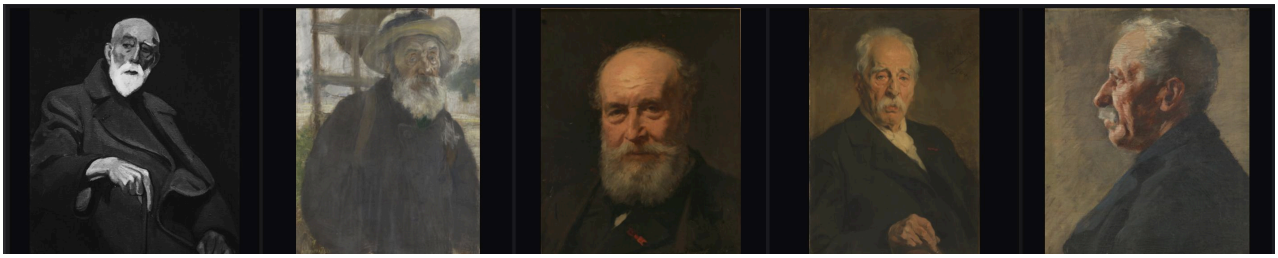


FIGURE 6.10: Sorting of the artworks using the *Path between two terms* button with the first prompt being *A man with a beard* and the second being *A man without a beard*

Settings

The settings tab provides the essential options. The user can select the model size that will power his queries between *art-mini*, *art-base* and *art-large*. The user can also modify the *soft* sub-queries weighting method (linear, squared or *Rocchio*).

[4]This is a **big** assumption that there exists a coherent linear semantic path between the two.

6.3.3 Tabs

Search results

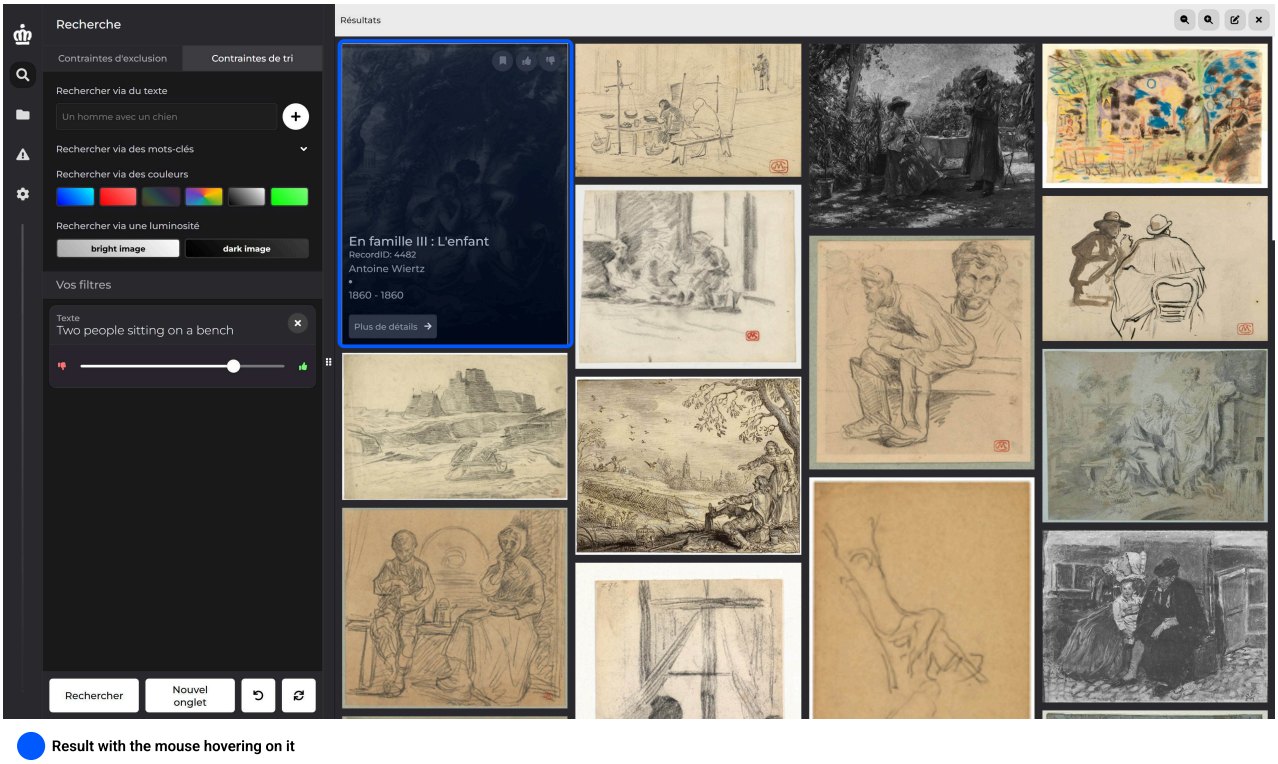


FIGURE 6.11: Screenshot of a results tab

The figure 6.11 presents a screenshot of a results *tab*. Artworks are arranged in a masonry-style grid. The number of columns is modifiable via buttons in the top-right of the *tab*. Hovering over an artwork reveals its title, the artist's name, estimated creation date, and *Subject Terms* if available. Each artwork features a *bookmark button* for adding it to a selected collection, along with like and dislike buttons. Liking (resp. disliking) an artwork will add a *soft* sub-query with a positive (resp. negative) weight for it. Additionally, a button is provided to open the artwork's profile.

Artwork profile

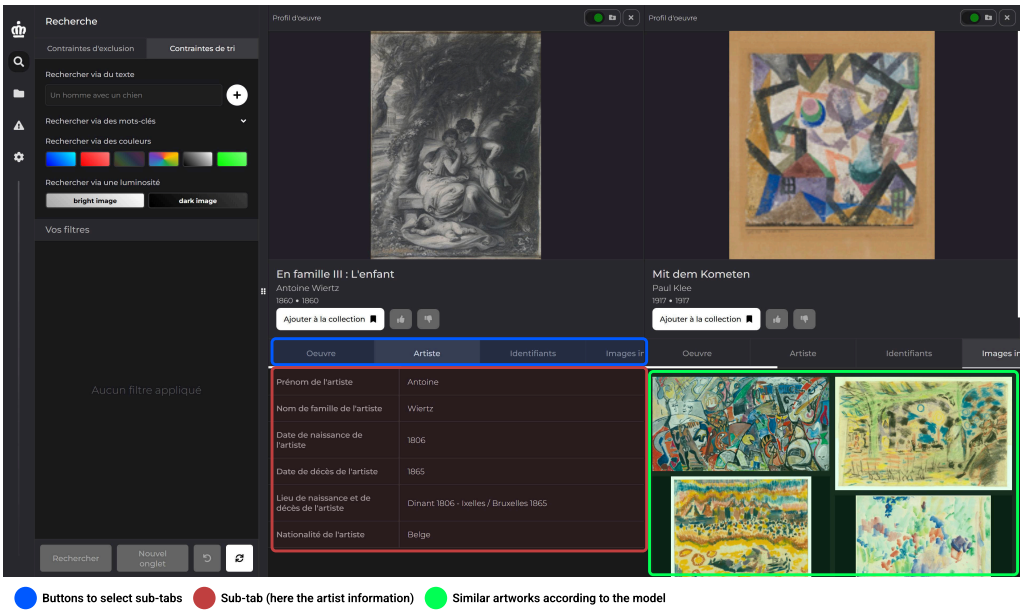


FIGURE 6.12: Screenshot of two artworks profile

The figure 6.12 presents a screenshot of two artwork's profiles. An artwork's profile contains all the information available in the database by using sub-tabs to separate the information into 10 categories:

1. *Information about the artwork*: its height and width, where it is signed, the type of canvas it is painted on, . . .
2. *Information about the artist*: its name, nationality, date of birth, . . .
3. *Identifiers*: Unique identifiers given by the RMFAB
4. *Similar artworks*: A grid of similar artworks according to the *CLIP* model. These artworks are the ones with the highest cosine similarity with the artwork presented in this profile. It is a very powerful way to find artworks similar to something that triggered something in us.
5. *Concepts*: List of concepts (*Subject matter*)
6. *Iconographic Terms*: List of iconographic terms (*Subject matter*)
7. *Subjects*: List of subjects (*Subject matter*)
8. *Interpretation of the iconography*: the textual interpretation of the iconography (*Subject matter*)
9. *General subject description*: Textual description of the image (*Subject matter*)
10. *Identification of the subjects*: Textual identification of the subjects (*Subject matter*)

Artist page

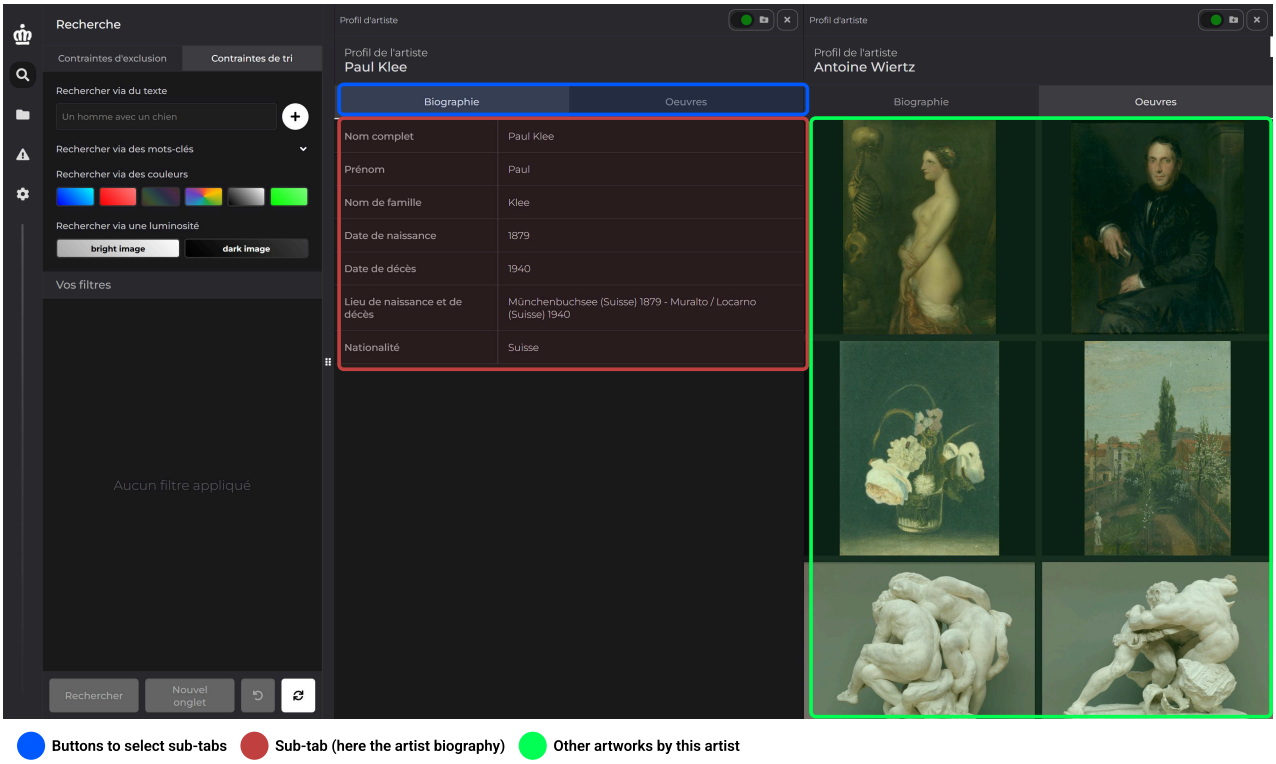


FIGURE 6.13: Screenshot of two artists profile

The figure 6.13 presents a screenshot of two artist's profiles. An artist's profile contains all the information available in the database by using 2 sub-tabs. The first sub-tab contains the biography of the artist while the other contains a grid display of the artworks made by this artist. By using the *CLIP* embeddings and the *subject matter* fields of the artworks, we wanted to add a third sub-tab showcasing the common subjects this artist likes to paint on. Sadly, we couldn't finish this feature in time.

Chapter 7

Data Generation

This thesis has produced three performant *CLIP* models on artworks and a search engine powered by them. This is already a very useful tool for the museums as finding artworks in their digital gallery is currently cumbersome. But another usage of AI for museums has been suggested during our discussions with the *RMFAB*. *AI* could be used to help curators to annotate artworks. Indeed, making an iconographic description of an artwork like the ones present in the *Subject Matter* fields is a very time-consuming task that requires a comprehensive and critical review of an artwork's complete documentation including scholarly articles, historical notes compiled since the artwork's acquisition and other notes from previous curators. But by looking at the artwork, a curator can already have an idea of the *iconographic terms* that might be relevant to it. The idea would be to *predict* relevant keywords for an artwork by having our models look at the artwork and at the *iconographic terms* already selected or already discarded by the curator. In practise, this could be implemented in the annotation software as an autocomplete feature for example.

In our opinion, this idea is in itself a complete master's thesis, and therefore we will only offer a possible approach to solving this problem, a brief analysis of the metrics we got and a small demonstrative prototype of how it could be used. We used our *art-base CLIP* model to generate the embeddings used throughout this chapter.

7.1 Hypothesis

We have several hypothesis behind our approach:

1. **Image propagation**: Two artworks judged as similar by *art-base* share common *iconographic terms*
2. **Term influence**: The *CLIP* embedding of an *iconographic term* is more similar to the *CLIP* of an artwork containing it than not
3. **Positive conditional**: If we already have selected one or more *iconographic terms*, the next *iconographic terms* we will select will probably intersect with the *iconographic terms* of the artworks containing similar keywords to what we have selected
4. **Negative conditional**: If we already have discarded one or more *iconographic terms*, the next *iconographic terms* we will select will probably not intersect with the *iconographic terms* of the artworks containing similar keywords to what we have discarded

7.2 Algorithm

7.2.1 Choosing a subset of *iconographic terms*

We only kept *iconographic terms* that do not contain any uppercase letter and that appear in at least two artworks. The latter condition is specific to our algorithm as having an ***iconographic term* that appears only once could never be predicted**.

7.2.2 Context and notation

Let a_i be the identifier of the target artwork for which we want to predict *iconographic terms* for having its *CLIP* embedding $e_i \in \mathbb{R}^d$ (d being the *CLIP* embedding dimension). Let $A_{train} = \{a_1, a_2, \ldots, a_m\}$ be a set of M training artworks, each a_j has a *CLIP* embedding E_j in E_{train} and a set of ground-truth *iconographic terms* denoted by $Terms(a_j)$. Let $U = \{u_1, u_2, \ldots, u_P\}$ be the set of P unique *iconographic terms* present in the training artworks, each $u_k \in U$ has a *CLIP* embedding $T_k \in \mathbb{R}^d$. The set *Selected* are selected *iconographic terms* and the set *Discarded* are discarded *iconographic terms*.

7.2.3 Computing similarities

The similarity measure is the *cosine similarity* as usual when working with a *CLIP* model. The first step is to compute the *artworks cosine similarity matrix* C with $C_{ij} = cosine\_similarity(e_i, E_j)$ and the *terms cosine similarity matrix* L with $C_{ik} = cosine\_similarity(e_i, T_k)$.

7.2.4 Hyperparameters

Our algorithm has several hyperparameters that control the strengths of the hypothesis described previously:

1. θ_{sim}: The minimum *artwork cosine similarity* C_{ij} required so that a_j is taken into account. If during the prediction process no neighbour is found using the θ_{sim} provided, it is set the the maximum value of C where $i \neq j$.
2. ρ: A higher *proximity* ρ gives disproportionately more weight to closer and/or positively conditioned neighbours.
3. α: The strength of **conditioning** must ≥ 0 and ≤ 1.
4. N: Either an integer > 0 that is the number of predictions to return or either a float ≥ 0 and < 1 which is the threshold above or equal which a term should be returned

7.2.5 Computing term score

A score S_k needs to be computed for each *iconographic terms* u_k. This is done by iterating over the neighbours of a_i defined as each $a_j \in A_{train}$ having $C_{ij} \geq \theta_{sim}$ and $j \neq i$.

Conditional Multiplier m_j

This multiplier adjusts the influence of a neighbour a_j based on its terms' overlap with the provided *Selected* and *Discarded* lists. Let $K_j = Terms(a_j) \cap Selected$ be the set of terms of neighbour a_h that are in the *Selected* list for a_i. And let $B_j = Terms(a_j) \cap Discarded$ be the set of terms of neighbour a_h that are in the *Discarded* list for a_i. The multiplier m_j is defined as:

$$m_j = (1 - \alpha) + \alpha \times \frac{exp(|K_j|)}{exp(2 \times |B_j|)} \tag{7.1}$$

Notice the 2 multiplier for the number of banned terms, we penalize *discarded* terms stronger than *selected* terms. This is an assumption.

Neighbour Score Contribution s_j

The contribution of the neighbour a_j is calculated by combining its *artwork cosine similarity* C_{ij} with the multiplier m_j, and raising it to the power of the *proximity* hyperparameter ρ:

$$s_j = (C_{ij} \times m_j)^{\rho} \tag{7.2}$$

Score aggregation

For every term $u_k \in Terms(a_j)$, its aggregate score S_k is incremented by s_j:

$$S_k = S_k + s_j \forall u_k \in Terms(a_j) \tag{7.3}$$

7.2.6 Term influence

After computing the terms scores, we do an element-wise multiplication of the matrices S and L such that each term's score is multiplied by its *term cosine similarity* with the embedding e_i of the artwork a_i:

$$S = S \odot L \tag{7.4}$$

7.2.7 Post-processing

There are a few remaining steps before providing the user with the predictions.

Removing *selected* and *discarded* terms

If $\alpha > 0$, then we set the term score S_k of every *selected* and *discarded* terms to 0.0. This step is required to have a correct normalization of the term scores later. The terms are also excluded from appearing in the results. If $\alpha = 0$, **this can results in empty predictions** for a given $N < 1$. This

weak point is necessary as setting their scores to 0.0 would modify the scores of the other predictions which contradicts the point of having the strength of **conditioning** α being equal to 0.

Normalization

The score are finally normalized to a $[0, 1]$ range to simplify the usage of the algorithm with a threshold N.

7.2.8 Performance improvements

A simple caching system can be used so that modifying the N parameter does not require to recompute the scores. On a modern desktop, predicting the *iconographic terms* from scratch for an artwork takes less than a tenth of a second.

7.3 Results

We ran benchmarks with 5 splits, each split using 3141 artworks to train and 500 artworks to test. Of course the splits are made of different artworks and there is no overlap between the training and the testing set.

7.3.1 Accuracy over N terms

The first metric we will look at is akin to an *accuracy* metric. For an artwork a_{test} with the ground truth terms $Terms(a_{test})$, we measure the quality of the predictions $Pred(a_{test}, N)$ by dividing the size of the overlap of the first $min(|Terms(a_{test})|, N)$ terms of the predicted list by the $min(|Terms(a_{test})|, N)$. Effectively measuring how well the prediction algorithm tested predicts:

$$acc_{a_{test}} = \frac{Pred(a_{test}, N) \cap Terms(a_{test}}{min(|Terms(a_{test})|, N)} \tag{7.5}$$

The figure 7.1 presents the accuracy over N terms with the baseline simply being an algorithm predicting the N most frequent terms in the training set. We see an improvement of a factor of 2 for every value of N between 1 and 10 with a very impressive accuracy of 91% on $N = 1$.

7.3.2 Training on the whole set

The second experiment we ran was to train the algorithm on the whole set, predict with $N = 0.5$ and measure how good are the predictions. We wanted to measure how well first-shot predictions with no input from the user (i.e. no *selected* or *discarded* terms) would be usable by annotators. In other words, this benchmark is a proxy to how much an annotator would click the predictions.

We measure the percentage of correct and incorrect predictions, the percentage of true terms missed by the predictions, the *accuracy*@1 and the *accuracy*@$MAX3$ (the accuracy if we wanted to predict a maximum of 3 terms, less strict than *recall*@3).

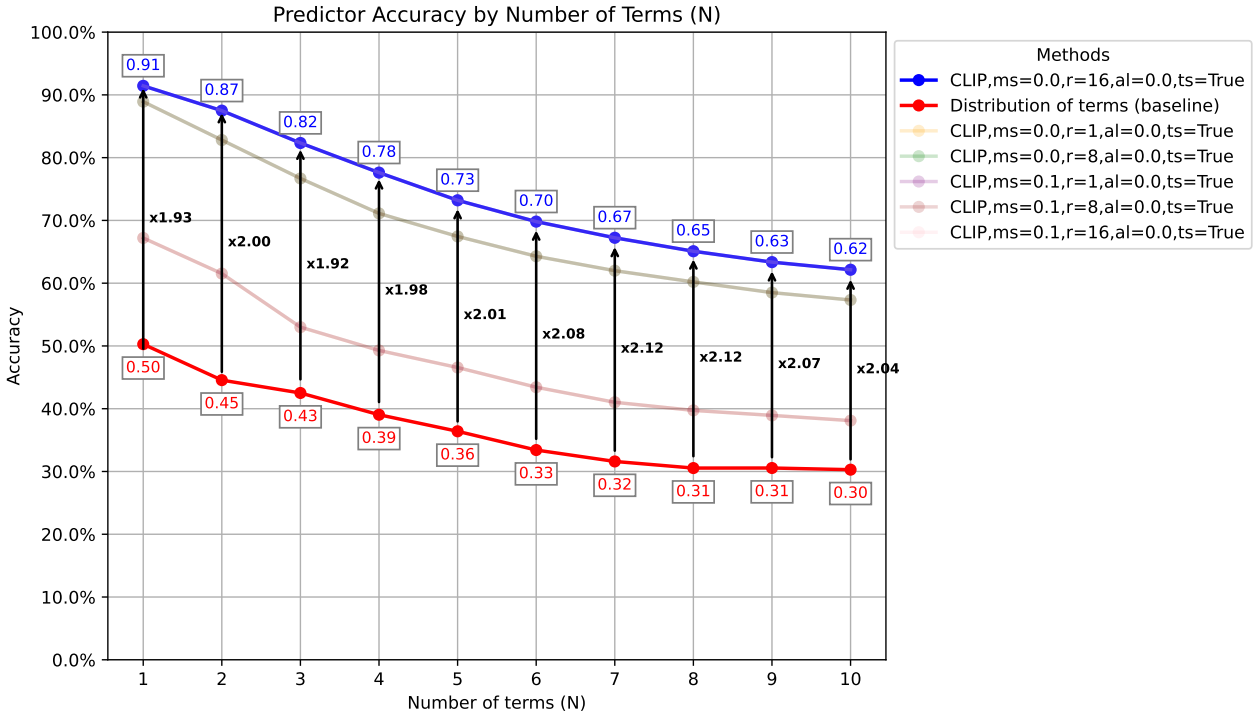


FIGURE 7.1: Predictor Accuracy by Number Of Terms (N)

With $N = 0.5$, a prediction was correct $80.66\% \pm 0.2278$ of the time ($19.34\% \pm 0.2278$ of the time it was incorrect). This is quite good at the intuitive threshold of 0.5 seems to produce good results. The algorithm clearly could not produce an iconography alone as $58.52\% \pm 0.2312$ of the ground truth terms were not present in the predictions. This is coherent with our scope of using this algorithm only as an autocomplete tool. The accuracy at 1 was really high at $91.35\% \pm 0.2811$ and the *accuracy@MAX3* was also high at $85.37\% \pm 0.2300$. The latter metric is interesting as it opens the door to providing up to 3 predictions quite confidently to the annotator.

7.3.3 Training on the whole set + generated iconographies

In this third experiment we trained the algorithm on the whole set of artworks including the artworks not having an any *iconographic terms* initially. We augmented our training set by simply using the predictions with $N = 0.5$ for these artworks. To be clear, the predictor that was used to augment the training set did not see the artworks it was predicting for. The benchmarking procedure is the same as the previous one.

With $N = 0.5$, a prediction was correct slightly more with a percentage of $81.95\% \pm 0.2326$ of the time ($18.05\% \pm 0.2326$ of the time it was incorrect). Interestingly, more terms were missing than before in the predictions at $61.14\% \pm 0.2309$ ($+2.62$). The accuracy at 1 was slightly lower at $90.14\% \pm 0.2981$ and the *accuracy@MAX3* was also slightly lower at $85.27\% \pm 0.2340$. It would be

interesting to do this experiment with a lower N as on average with $N = 0.5$ there were only 4.47 terms predicted.

7.3.4 Impact of N

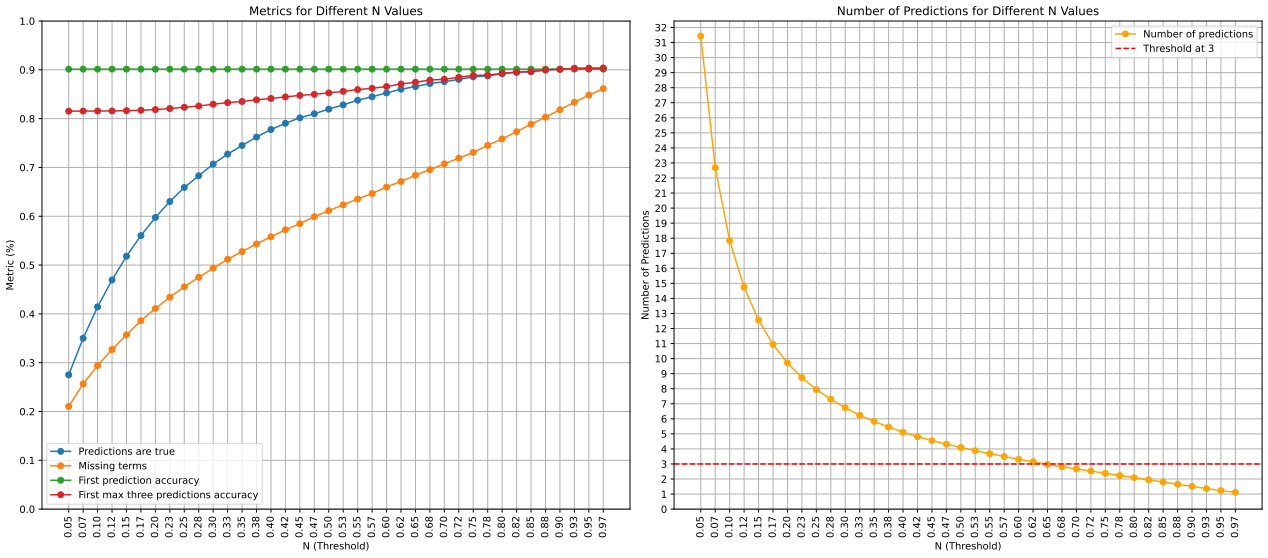


FIGURE 7.2: Metrics for different N values

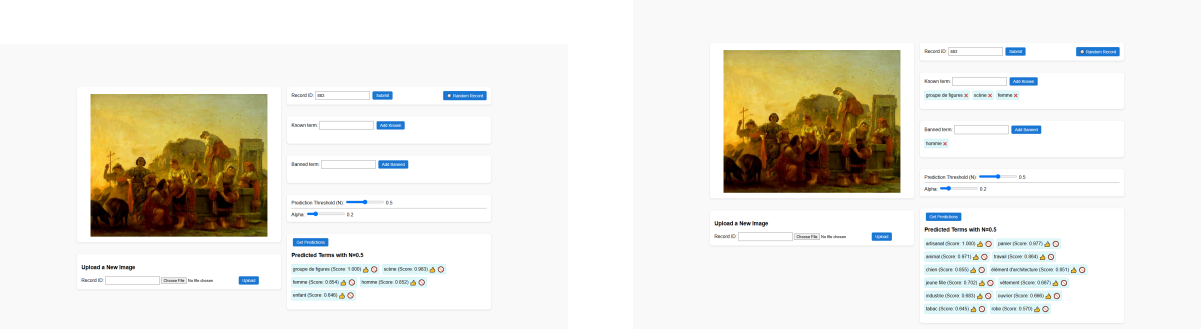
The figure 7.2 describe interesting phenomenons:

1. Even with a very low N, the *accuracy@MAX3* is only rising from just above 80% to 90%.
2. Increasing N logically improves the accuracy of the predictions.
3. Even with $N = 0.05$, there are still about 20% of the terms that are not predicted. This slowly rises to hit about 85% at $N = 0.975$.
4. The predictions accuracy improves slower as N increases.
5. If we wanted to offer 3 predictions to the user, we can use a N value close to 0.6 and expect about 85% of our predictions to be relevant.

7.4 Prototype using this algorithm

A simple webpage prototype was made to illustrate how the algorithm could be used. This section presents a few example of its usage.

7.4.1 Example 1: Predicting on an artwork from the *RMFAB*



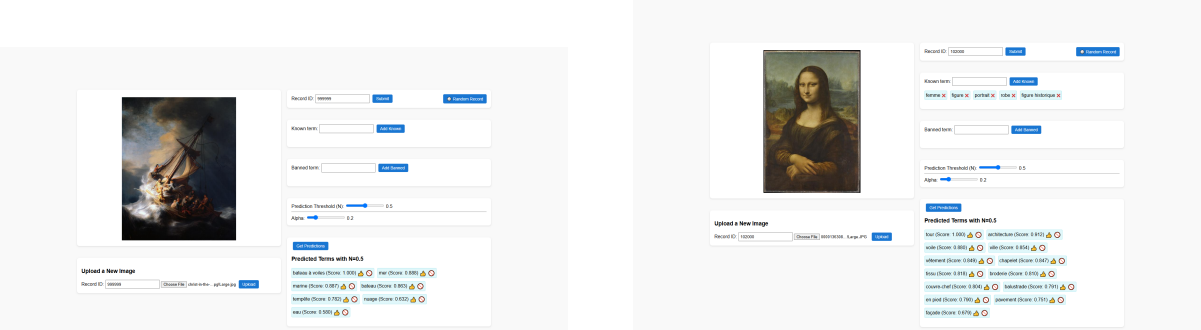
(A) First predictions

(B) Predictions after interaction with the *Selected* and *Discarded* inputs

FIGURE 7.3: Example 1: Predicting on an artwork from the *RMFAB*

In the example presented in 7.3 we can see that the one-shot predictions are quite good, when interacting with the *Selected* and *Discarded* inputs, we start to get rarer terms like *artisanat*, *travail*, *animal*.

7.4.2 Example 2: Predicting on an artwork from *WikiArt*



(A) *The Storm on the Sea of Galilee* by *Rembrandt*

(B) *Mona Lisa* by *Leonardo da Vinci*

FIGURE 7.4: Example 2: Predicting on an artwork from *WikiArt*

In the example presented in 7.4, we tested our algorithm on artworks it had never seen from the *WikiArt* collection. The one-shot predictions for *The Storm on the Sea of Galilee* by *Rembrandt* were

really accurate and impressive. We also tested our algorithm on the *Mona Lisa* by *Leonardo da Vinci* where we clicked on the first prediction 5 times. The resulting terms were: femme (woman), figure (figure), portrait (portrait), robe (dress/robe) and figure historique (historical figure). This iterative approach with $alpha = 0.2$ seems to take advantage of the additional context we give the algorithm.

7.5 Limitations and improvements

Our algorithm is a strong first proposition, yet we can note several issues. The first important issue is the high number of hyperparameters makes this approach difficult to tune at time. Although we have found that using $N = 0.65$, $\alpha = 0.20$ and $\rho = 16.0$ perform well in general on the *RMFAB* dataset, we have no clue as to how that would evolve with more *iconographic terms* added or with another *CLIP* model to generate the embeddings. The second issue is a phenomenon that we have observed qualitatively when testing the prototype where we would be "*trapped*" in a *thematic cage*. As more terms are added to the *Selected* set, the predictions are closer and closer to similar terms, for example, by selecting *arme* (weapon) and *uniforme* (uniform), we start to get almost exclusively terms related to the thematic field of war. This behaviour is desirable most of the time but can also be annoying when we are trying to add an unrelated term to this thematic field.

Chapter 8

Further Research

Further improving the multilingual capabilities Although our models perform well enough in the languages we decided to cover, we believe that there are still many improvements possible in that area. As mentioned through this thesis, the simplest possibility would be to replace the textual encoder of *CLIP* by a more adequate one, for example one borrowed from *M-CLIP* [5]. This has not been implemented in this thesis because *art-* models were already sufficient for our use case and because the timing did not allow to implement *M-CLIP* and to retrain the models.

A simple yet interesting idea would be to prepend a *conditioning token* at the start of a query depending on the language selected by the user. For example, if the user set the Search Engine language to Dutch, we would prepend to his query the prefix *NL:*. Of course, the training set used to finetune the models would also be modified by prepending every caption with the prefix of its language. We thought about using this approach when noticing that querying for *pain* in French (which means *bread*) retrieved artworks depicting tormented faces. This is simply because the word *pain* also exists in English. Of course, by giving more context to the query such as *pain du boulanger*, correct results can be retrieve.

Predicting objects using *CLIP* embeddings and neighbour's information The chapter *Data Generation* 7 presented a first step in generating *iconographic terms* using the knows terms of other artworks and the *CLIP* embeddings produced by our models. But we believe that this direction is in itself a complete master's thesis.

A more intricate solution to this problem would see the artworks as nodes in a graph G linked together by edges whose weight is their *cosine distance*[1]. This vision opens the door to many pre-existing algorithms. An adaptation of the famous *PageRank* [36] algorithm was discussed as a possible research path.

[1] The *cosine distance* is simply $1 - cosine\_similarity(A - B)$

Multi VLM pipelines As explained in the *Theoretical Background* chapter of this thesis 3, distilling the knowledge from VLMs is a proposed approach to using their broad knowledge into specific models. In our case, we used a single small and limited VLM model called *moondream2* [51]. If this project was to continue, we would like to further increase the performance of the *CLIP* models by generating more captions with more performant VLM models.

The ideal data generation pipeline would use several VLMs that would produce more varied captions. The top-*X*% best performing captions on a *BLEU* score benchmark could be kept to further improve the quality of the generated training set. Furthermore, the bottleneck of only 5301 images available to us could be circumvented by using artworks from the *WikiArt* dataset. This could potentially multiply the number of artworks that the models trained on by a factor of 20. Yet, we must acknowledge the potential copyright issues that we may encounter as the weights of a model trained on copyrighted images could fall under their copyright potentially. This is a blurry area that must be clarified before any training is done.

Editor in the search engine It has been discussed with the *RMFAB* to implement edition features in the *Search Engine*. Indeed, when doing live presentations, we noticed small mistakes like typos or missing fields that could be easily filled. By adding simple edition interfaces sending the modifications to the API, the museum could have an efficient internal tool with which they could slowly but surely improve their digital dataset.

Chapter 9

Conclusion

This thesis successfully showcased how collaborating with an innovative museum like the *Royal Museums of Fine Arts of Belgium* (*RMFAB*) can use the recent *AI* advancements to train several cutting-edge models that can then be used to explore a large digital gallery of artworks with ease. Furthermore, we have demonstrated how these kinds of models can be reused for other tasks, like predicting *iconographic terms*.

The team behind this thesis has explored many paths and presented them to the *RMFAB* which chose the project that would help them most in their quest to bring art to everyone, especially Alzheimer patients. We have meticulously navigated the challenges of working with specialized and often imperfect datasets by developing innovative multilingual synthetic data generation pipelines. The iterative fine-tuning process demonstrated robust performance improvements in French, English, and Dutch across several benchmarks made especially for this project.

The resulting prototype search engine is a complete software, integrating powerful hard and soft querying capabilities, relevance feedback mechanisms and creative tools like similar artworks or *Collections*. These features enhance the exploratory search experience for curators and researchers but also hold potential for engaging the public and supporting therapeutic activities, such as Reminiscence Therapy (RT) for Alzheimer's patients.

While the research has achieved impressive results already, we also highlights areas for future development, including further enhancements to multilingual robustness and the possibility of using our models for semi-automated artwork annotation. This work serves as a strong proof that AI can **help** the cultural sector, providing a valuable blueprint for similar institutions seeking to unlock the richness of their collections.

9.1 Usage of AI

Gemini was employed for the final proofreading to detect typos or incoherent formulation and DeepL assisted with translations.

9.2 Code repository

The full code for this projects is available at `https://github.com/on-victorrijks/Museum-Search-Engine`.

Appendix A

Table containing the 34 unique Object Work Type present in the February Subset of the RMFAB dataset

Object Work Type	Number of appearances	Percentage of appearances	Object Work Type	Number of appearances	Percentage of appearances	Object Work Type	Number of appearances	Percentage of appearances
tableau (toile)	1243	41.27%	croquis	14	0.46%	figurine	4	0.13%
dessin	748	24.83%	estampe	14	0.46%	fragment	4	0.13%
tableau (panneau)	522	17.33%	projet	13	0.43%	retable (partie de)	4	0.13%
esquisse	177	5.88%	triptyque (partie de)	11	0.37%	photographie	3	0.1%
aquarelle	158	5.25%	carnet de dessins	10	0.33%	étude académique	2	0.07%
pastel	46	1.53%	groupe	8	0.27%	affiche	1	0.03%
étude	38	1.26%	dessin aux 3 crayons	5	0.17%	album	1	0.03%
huile sur papier	27	0.9%	détrempe	5	0.17%	miniature	1	0.03%
gouache	25	0.83%	tondo	5	0.17%	ébauche	1	0.03%
dessin aux 2 crayons	19	0.63%	statuette	4	0.13%	tête	1	0.03%
triptyque	19	0.63%	polyptyque (partie de)	4	0.13%			
sanguine	18	0.6%	haut-relief	4	0.13%			

TABLE A.1: Table containing the 34 unique Object Work Type present in the February Subset of the RMFAB dataset

Appendix B

Centroid coordinates when running the algorithm 23 on the February subset of the RMFAB dataset

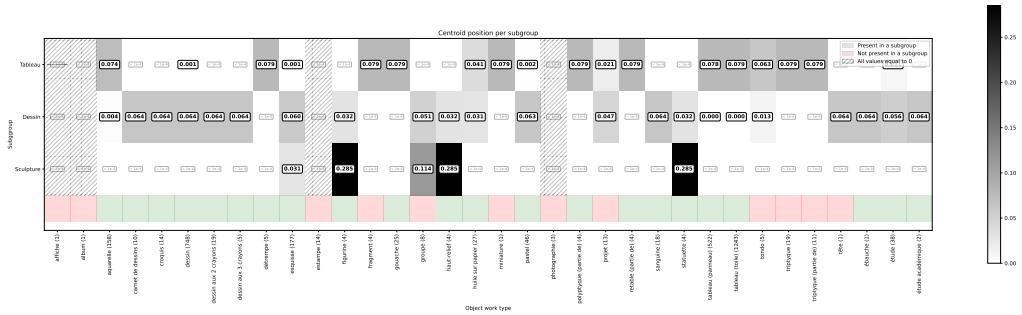


FIGURE B.1: Centroid coordinates when running the algorithm 23 on the February subset of the RMFAB dataset

Appendix C

Pseudocode of the centroid finder algorithm

Algorithm 1 Centroid finder algorithm

$d =$ Number of unique values
$c =$ Number of centroid
$v =$ One hot encoding of the artworks
$\alpha =$ A very small positive real ($1e-9$)
Let $A \in \mathbb{N}^{c \times d}$ ▷ A is a matrix storing the count of object work type for each subgroup

for all $subgroup \in subgroups$ **do**
    for all $o \in subgroup$ **do**
        $artworks \leftarrow$ artworks containing the object work type o
        $A_{subgroup} \leftarrow A_{subgroup} + sum(v_{artworks})$ ▷ Add the one hot encodings
    end for
end for

Normalize per object work type
$G \leftarrow sum(A, axis = 0)$ ▷ Get the sum per object work type
for all $subgroup \in subgroups$ **do**
    $A_{subgroup} \leftarrow \mathrm{A}_{subgroup} / (G + \alpha)$ ▷ α required to avoid dividing by 0
end for

# Normalize per subgroup
for all $subgroup \in subgroups$ **do**
    $A_{subgroup} \leftarrow \mathrm{A}_{subgroup} / sum(A_{subgroup})$
end for

Appendix D

Table containing the explicit object work types for the three subgroups

Tableau	Dessin	Sculpture
aquarelle	dessin aux 2 crayons	figurine
détrempe	dessin aux 3 crayons	haut-relief
gouache	croquis	statuette
huile sur papier	esquisse	
polyptyque (partie de)	pastel	
retable (partie de)	étude	
tableau (panneau)	étude académique	
tableau (toile)	carnet de dessins	
	sanguine	
	ébauche	
	étude	
	étude académique	
	dessin	

TABLE D.1: Table containing the explicit object work types for the three subgroups

Appendix E

List of styles kept for the WikiArt dataset

Abstract-Art (1980), Abstract-Expressionism (3600), Academicism (2769), Art-Deco (1122), Art-Informel (1888), Art-Nouveau-(Modern) (3600), Baroque (3600), Color-Field-Painting (1629), Conceptual-Art (2111), Concretism (797), Constructivism (598), Contemporary (797), Contemporary-Realism (1755), Cubism (3403), Early-Renaissance (1876), Expressionism (3600), Fauvism (1176), Feminist-Art (532), Hard-Edge-Painting (639), High-Renaissance (1737), Impressionism (3600), Ink-and-wash-painting (835), Kitsch (888), Lyrical-Abstraction (1208), Magic-Realism (1758), Mannerism-(Late-Renaissance) (2429), Minimalism (2242), Naturalism (880), Naïve-Art-(Primitivism) (3600), Neo-Expressionism (1924), Neo-Impressionism (1165), Neo-Pop-Art (705), Neo-Romanticism (828), Neoclassicism (3600), Northern-Renaissance (3239), Op-Art (1219), Orientalism (1467), Pictorialism (526), Pointillism (591), Pop-Art (2708), Post-Impressionism (3600), Post-Minimalism (747), Realism (3600), Regionalism (581), Rococo (3541), Romanticism (3600), Social-Realism (925), Socialist-Realism (708), Street-art (502), Surrealism (3600), Symbolism (3600), Tachisme (576), Tenebrism (552), Transavantgarde (601), Ukiyo-e (1857),

Appendix F

Mean rank per language and category (February finetune)

Language	Category	Mean	Std	N
French	Dessin	4.5214	10.069	152.0
French	Tableau	3.2331	7.1141	296.0
French	Sculpture	1.9583	0.7558	6.0
English	Dessin	3.3289	5.6899	152.0
English	Tableau	1.8564	3.0789	296.0
English	Sculpture	2.0417	0.9176	6.0
Dutch	Dessin	26.6661	48.6101	152.0
Dutch	Tableau	14.7035	30.0287	296.0
Dutch	Sculpture	77.625	87.697	6.0
Mean	Dessin	11.5055	21.4563	152.0
Mean	Tableau	**6.5977**	**13.4072**	296.0
Mean	Sculpture	27.2083	29.7901	6.0

TABLE F.1: Table presenting the mean and the standard deviation of the rank for the Benchmark 1 on the February model depending on the language and the category

Appendix G

Subject matter dataset in details

Field	<= TOKEN_LIMIT	>TOKEN_LIMIT	% <= TOKEN_LIMIT	% >TOKEN_LIMIT	Average overflow (tokens)
Textual interpretation of the iconography	21	5	80.77%	19.23	2.40
Textual description of the image (caption)	92	3	96.84%	3.16	3.67
Textual identification of the subjects present in the image	106	2	98.15%	1.85	9.50
Subject matter objects present in the image	3581	10	99.72%	0.28	3.40
Subject matter of the iconographies present in the image	2809	2	99.93%	0.07	3.00
Subject matter of the concepts present in the image	670	0	100%	0%	0
Total	7279	22	99.70%	0.30%	

TABLE G.1: Table summarizing the manual information dataset from the RMFAB digital gallery

Appendix H

Fabritius screenshots (1)

FIGURE H.1: Fabritius - Simple search

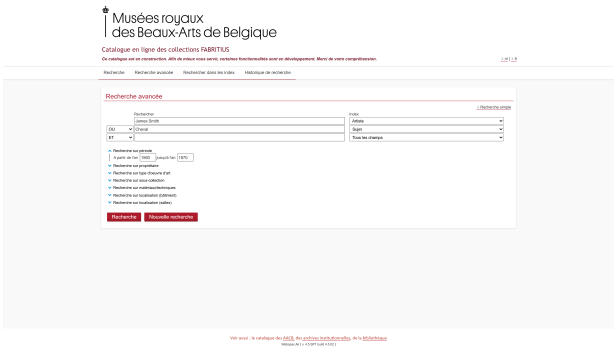


FIGURE H.2: Fabritius - Complex search

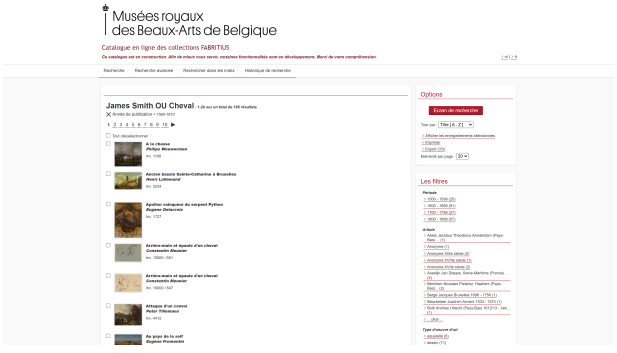


FIGURE H.3: Fabritius - Results page

FIGURE H.4: Fabritius - Artwork's profile

FIGURE H.5: Screenshots of the *Fabritius* search engine

Appendix I

Fabritius screenshots (2)

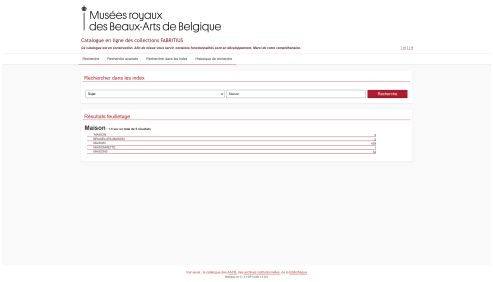


FIGURE I.1: Fabritius - Index

FIGURE I.2: Screenshots of the *Fabritius* search engine

Bibliography

[1] Marcia J Bates. "The design of browsing and berrypicking techniques for the online search interface". In: *Online review* 13.5 (1989), pp. 407–424.

[2] Aaditya Bhat and Shrey Jain. *Face Recognition in the age of CLIP Billion image datasets*. 2023. arXiv: 2301.07315 [cs.CV]. URL: https://arxiv.org/abs/2301.07315.

[3] Bruno Bouchard et al. "Developing Serious Games Specifically Adapted to People Suffering from Alzheimer". In: *Serious Games Development and Applications*. Springer Berlin Heidelberg, 2012, 243–254. ISBN: 9783642336874. DOI: 10.1007/978-3-642-33687-4_21. URL: http://dx.doi.org/10.1007/978-3-642-33687-4_21.

[4] Hongping Cai et al. *The Cross-Depiction Problem: Computer Vision Algorithms for Recognising Objects in Artwork and in Photographs*. 2015. arXiv: 1505.00110 [cs.CV]. URL: https://arxiv.org/abs/1505.00110.

[5] Fredrik Carlsson et al. "Cross-lingual and Multilingual CLIP". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, June 2022, pp. 6848–6854. URL: https://aclanthology.org/2022.lrec-1.739/.

[6] Diana Castilla et al. "Process of design and usability evaluation of a telepsychology web and virtual reality system for the elderly: Butler". In: *International Journal of Human-Computer Studies* 71.3 (Mar. 2013), 350–362. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2012.10.017. URL: http://dx.doi.org/10.1016/j.ijhcs.2012.10.017.

[7] Fine-Grained Visual Categorization and iMet. *iMet Collection 2019 - FGVC6 — kaggle.com*. https://www.kaggle.com/c/imet-2019-fgvc6. [Accessed 29-05-2025]. 2019.

[8] Mehdi Cherti and Romain Beaumont. *CLIP benchmark*. May 2025. DOI: 10.5281/zenodo.15403103. URL: https://doi.org/10.5281/zenodo.15403103.

[9] Mehdi Cherti et al. "Reproducible scaling laws for contrastive language-image learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2818–2829.

[10] Brendan Ciecko. "AI sees what? The good, the bad, and the ugly of machine vision for museum collections". In: *MW2020: Museums and the Web* 5.1 (2020).

[11] Marcos V. Conde and Kerem Turgutlu. "CLIP-Art: Contrastive Pre-training for Fine-Grained Art Classification". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, June 2021. DOI: 10.1109/cvprw53098.2021.00444. URL: http://dx.doi.org/10.1109/CVPRW53098.2021.00444.

[12] Ora Coster and Theo Coster. *Guess Who?* Board Game. Board game designed by Theo Coster and Ora Coster (Theora Design). Original publisher: Milton Bradley (later acquired by Hasbro). 1979.

[13] Yufeng Cui et al. *Democratizing Contrastive Language-Image Pre-training: A CLIP Benchmark of Data, Model, and Supervision*. 2022. arXiv: 2203.05796 [cs.CV]. URL: https://arxiv.org/abs/2203.05796.

[14] Simon Douglas, Ian James, and Clive Ballard. "Non-pharmacological interventions in dementia". In: *Advances in Psychiatric Treatment* 10.3 (May 2004), 171–177. ISSN: 1472-1481. DOI: 10.1192/apt.10.3.171. URL: http://dx.doi.org/10.1192/apt.10.3.171.

[15] Ministère de la Culture Française. *Thésaurus iconographique, système descriptif des représentations de François Garnier | Ministère de la Culture — culture.gouv.fr*. https://www.culture.gouv.fr/thematiques/musees/pour-les-professionnels/conserver-et-gerer-les-collections/informatiser-les-collections-d-un-musee-de-france/vocabulaires-scientifiques-du-service-des-musees-de-france/thesaurus-iconographique-systeme-descriptif-des-representations-de-francois-garnier. [Accessed 26-05-2025].

[16] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. "Exploring network structure, dynamics, and function using NetworkX". In: Los Alamos National Laboratory (LANL), Los Alamos, NM (United States). Jan. 2008. URL: https://www.osti.gov/biblio/960616.

[17] Toshimi Hamada et al. "Preliminary Study on Remote Assistance for People with Dementia at Home by Using Multi-media Contents". In: *Universal Access in Human-Computer Interaction. Addressing Diversity*. Ed. by Constantine Stephanidis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 236–244. ISBN: 978-3-642-02707-9.

[18] Fumio Hattori et al. "Socialware for People with Disabilities". In: *6th IEEE International Conference on Cognitive Informatics*. IEEE, Aug. 2007. DOI: 10.1109/coginf.2007.4341905. URL: http://dx.doi.org/10.1109/COGINF.2007.4341905.

[19] Gabriel Ilharco et al. *OpenCLIP*. Version 0.1. If you use this software, please cite it as below. July 2021. DOI: 10.5281/zenodo.5143773. URL: https://doi.org/10.5281/zenodo.5143773.

[20] Frederick Imbeault, Bruno Bouchard, and Abdenour Bouzouane. "Serious games in cognitive training for Alzheimer's patients". In: *2011 IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, Nov. 2011, 1–8. DOI: 10.1109/segah.2011.6165447. URL: http://dx.doi.org/10.1109/SeGAH.2011.6165447.

[21] Tremblay Jonathan, Bouchard Bruno, and Bouzouane Abdenour. "Understanding and Implementing Adaptive Difficulty Adjustment in Video Games". In: *Algorithmic and Architectural Gaming Design*. IGI Global, 2012, 82–106. DOI: 10.4018/978-1-4666-1634-9.ch005. URL: http://dx.doi.org/10.4018/978-1-4666-1634-9.ch005.

[22] Myong Chol Jung and Jesse Clark. *Marqo-FashionCLIP and Marqo-FashionSigLIP*. Version 1.0.0. Aug. 2024. URL: https://github.com/marqo-ai/marqo-FashionCLIP.

[23] Hyunjae Kim et al. "Fine-tuning CLIP Text Encoders with Two-step Paraphrasing". In: *Findings of the Association for Computational Linguistics: EACL 2024*. Ed. by Yvette Graham and Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 2175–2184. URL: https://aclanthology.org/2024.findings-eacl.144/.

[24] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[25] Kodak et al. *Memory Shots - Kodak*. https://memory-shots.org/. [Accessed 26-05-2025].

[26] Noriaki Kuwahara et al. "Networked reminiscence therapy for individuals with dementia by using photo and video sharing". In: *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets '06. Portland, Oregon, USA: Association for Computing Machinery, 2006, 125–132. ISBN: 1595932909. DOI: 10.1145/1168987.1169010. URL: https://doi.org/10.1145/1168987.1169010.

[27] Amanda Lazar, Hilaire Thompson, and George Demiris. "A Systematic Review of the Use of Technology for Reminiscence Therapy". In: *Health Education amp; Behavior* 41.1$_s$*uppl* (Oct. 2014), 51S–61S. ISSN: 1552-6127. DOI: 10.1177/1090198114537067. URL: http://dx.doi.org/10.1177/1090198114537067.

[28] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[29] Jesse Clark Marquo.ai. *Benchmarking Models for Multi-modal Search — marqo.ai*. https://www.marqo.ai/blog/benchmarking-models-for-multimodal-search. [Accessed 27-05-2025].

[30] Louie Meyer et al. "Algorithmic Ways of Seeing: Using Object Detection to Facilitate Art Exploration". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24. ACM, May 2024, 1–18. DOI: 10.1145/3613904.3642157. URL: http://dx.doi.org/10.1145/3613904.3642157.

[31] Harvard Art Museums. *Home | AI Explorer | Harvard Art Museums — ai.harvardartmuseums.org*. https://ai.harvardartmuseums.org/. [Accessed 29-05-2025].

[32] *Musée sur mesure — fine-arts-museum.be*. https://fine-arts-museum.be/fr/education/musee-sur-mesure. [Accessed 26-05-2025].

[33] Pavol Navrat. "Cognitive traveling in digital space: from keyword search through exploratory information seeking". In: *Open Computer Science* 2.3 (Oct. 2012), 170–182. ISSN: 2299-1093. DOI: 10.2478/s13537-012-0024-6. URL: http://dx.doi.org/10.2478/s13537-012-0024-6.

[34] Helsinki NLP Opus-MT. *Helsinki-NLP/opus-mt-en-fr · Hugging Face — huggingface.co*. https://huggingface.co/Helsinki-NLP/opus-mt-en-fr. [Accessed 27-05-2025].

[35] OpenAI. *OpenAI — openai.com*. https://openai.com/. [Accessed 27-05-2025].

[36] Lawrence Page et al. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, 1999. URL: http://ilpubs.stanford.edu:8090/422/.

[37] Emilie Palagi. "Quelle méthode ergonomique élaborer pour évaluer les moteurs de recherche exploratoire ?" In: *COnférence en Recherche d'Information et Applications 2015 (CORIA 2015)*. Paris, France, Mar. 2015. URL: https://inria.hal.science/hal-01150631.

[38] Emilie Palagi et al. "A Survey of Definitions and Models of Exploratory Search". In: *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics*. ESIDA '17. Limassol, Cyprus: Association for Computing Machinery, 2017, 3–8. ISBN: 9781450349031. DOI: 10.1145/3038462.3038465. URL: https://doi.org/10.1145/3038462.3038465.

[39] Lulu Pei et al. "Employing AI-Based Tools to Support Exhibition Design: A Science and Technology Museum Case Study". In: *Creativity in the Age of Digital Reproduction*. Ed. by Giancarlo Di Marco, Davide Lombardi, and Mia Tedjosaputro. Singapore: Springer Nature Singapore, 2024, pp. 217–224. ISBN: 978-981-97-0621-1.

[40] *Programa de estimulación cognitiva con eficacia probada — smartbrain.es*. https://www.smartbrain.es/. [Accessed 26-05-2025].

[41] Aung Pyae et al. "Serious games and active healthy ageing: a pilot usability testing of existing games". In: *International Journal of Networking and Virtual Organisations* 16.1 (2016), p. 103. ISSN: 1741-5225. DOI: 10.1504/ijnvo.2016.075129. URL: http://dx.doi.org/10.1504/IJNVO.2016.075129.

[42] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[43] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *ICML*. 2021.

[44] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. cite arxiv:1506.02640. 2015. URL: http://arxiv.org/abs/1506.02640.

[45] Manon Reusens, Amy Adams, and Bart Baesens. "Large Language Models to make museum archive collections more accessible". In: *AI amp; SOCIETY* (Feb. 2025). ISSN: 1435-5655. DOI: 10.1007/s00146-025-02227-8. URL: http://dx.doi.org/10.1007/s00146-025-02227-8.

[46] Joseph J. Rocchio. "Relevance Feedback in Information Retrieval". PhD thesis. Cambridge, MA, USA: Harvard University, 1971.

[47] Christoph Schuhmann et al. "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022. URL: https://openreview.net/forum?id=M3Y74vmsMcY.

[48] L Tarraga. "A randomised pilot study to assess the efficacy of an interactive, multimedia tool of cognitive stimulation in Alzheimer's disease". In: *Journal of Neurology, Neurosurgery amp; Psychiatry* 77.10 (June 2006), 1116–1121. ISSN: 0022-3050. DOI: 10.1136/jnnp.2005.086074. URL: http://dx.doi.org/10.1136/jnnp.2005.086074.

[49] Jörg Tiedemann and Santhosh Thottingal. "OPUS-MT — Building open translation services for the World". In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal, 2020.

[50] Tiffany Tong, Jonathan H. Chan, and Mark Chignell. "Serious Games for Dementia". In: *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*. WWW '17 Companion. ACM Press, 2017, 1111–1115. DOI: 10.1145/3041021.3054930. URL: http://dx.doi.org/10.1145/3041021.3054930.

[51] Vikhyat Korrapati. *moondream2*. 2024. DOI: 10.57967/hf/3219. URL: https://moondream.ai/.

[52] *Vision Language Models Explained — huggingface.co*. https://huggingface.co/blog/vlms. [Accessed 29-05-2025].

[53] JEFFREY DEAN WEBSTER. "Faith Gibson, The Past in the Present: Using Reminiscence in Health and Social Care, Health Professions Press, Baltimore, Maryland, 2004, 336 pp., pbk 32.95, *ISBN*1878812874.". In: *Ageing and Society* 25.5 (2005), 806–807. DOI: 10.1017/S0144686X05283964.

[54] S. Weisman. "Computer Games for the Frail Elderly". In: *The Gerontologist* 23.4 (Aug. 1983), 361–363. ISSN: 1758-5341. DOI: 10.1093/geront/23.4.361. URL: http://dx.doi.org/10.1093/geront/23.4.361.

[55] WikiArt. *huggan/wikiart · Datasets at Hugging Face — huggingface.co*. https://huggingface.co/datasets/huggan/wikiart. [Accessed 27-05-2025].

[56] WikiArt. *WikiArt.org - Visual Art Encyclopedia — wikiart.org*. https://www.wikiart.org/. [Accessed 27-05-2025].

[57] Jingyi Zhang et al. "Vision-Language Models for Vision Tasks: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.8 (Aug. 2024), 5625–5644. ISSN: 1939-3539. DOI: 10.1109/tpami.2024.3369699. URL: http://dx.doi.org/10.1109/TPAMI.2024.3369699.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**