Multimodal Similarity for Lost Art Retrieval

Maxime Moutet[*]
ENSAE
maxime.moutet@ensae.fr

Abstract

This project focuses on the development of an efficient system for identifying and comparing artworks across large-scale databases, specifically aiming at the retrieval of stolen art pieces from public and private collections. By leveraging multimodal similarity searches, we combine text-based metadata (such as descriptions, artist names, and historical information) with image-based representations to find matching or similar works of art. Using pre-trained models such as **BERT** for text embeddings and **DINO** for image embeddings, the system extracts meaningful features from both textual and visual data, ensuring robust cross-domain comparison. To address challenges related to the large size of image databases, the system is designed to download, process, and compare artworks dynamically, ensuring minimal memory usage.

1 Context and Objective

During the Second World War, the Nazi regime stole numerous works of art from museums, galleries, and private collections throughout Europe. At the end of the war, many of these stolen artworks were recovered and returned to their countries of origin, including France. However, despite efforts to return them, a significant number of owners did not find their asset. In response to this, owners or their representatives issued search notices in an attempt to trace and recover their lost art pieces. These notices were published to help reconnect artworks with their rightful owners. Over time, some of these works were exhibited in French museums, but many still remain without clear provenance.

This project seeks to develop a system capable of efficiently matching these artworks, based on search notices issued by the original owners, with those displayed in museums. Using advanced multimodal similarity search techniques, combining both textual descriptions and image-based features, the goal is to facilitate the identification and potential recovery of these artworks, helping to reunite them with their rightful owners.

2 State of the Art

This section reviews existing methods for similarity search, with a focus on techniques used for comparing text and images, and highlights advancements in multimodal approaches. Similarity search involves retrieving items from a database that are similar to a given query, and it has applications in fields such as information retrieval, recommendation systems, and content-based search.

Text-Based Similarity Search Text-based similarity search is traditionally performed by representing text data in a numerical format that allows for the calculation of similarity metrics such as cosine similarity. Early methods used bag-of-words (**BoW**) representations, which treat text as a collection of words, disregarding their order. While this method is simple, it fails to capture contextual information and semantic relationships between words.

[*] https://github.com/MoutetMaxime/multimodal-art-similarity

More sophisticated techniques, such as word embeddings, represent words in dense vector spaces where semantically similar words are closer together. These representations allow for more meaningful comparisons between texts. For example, **Word2Vec** [7] have been widely used for calculating word-level similarity.

Recent developments in transformer-based models have significantly advanced text similarity search. Models like **BERT** [3] and its multilingual versions, such as **mBERT**, have achieved state-of-the-art performance on a variety of NLP tasks, including similarity measurement. These models capture context-dependent word representations, making them more suitable for tasks involving long and complex text inputs. For instance, **sBERT** [10] modified **BERT** to represent entire sentences or paragraphs, facilitating the calculation of sentence-level similarity.

Image-Based Similarity Search Image-based similarity search relies on extracting features from images and comparing these features to measure similarity. The advent of deep learning has revolutionized image similarity search. Convolutional neural networks have been widely adopted for image feature extraction, where models such as **VGG** [11] or **ResNet** [5] are pre-trained on large image datasets like ImageNet and fine-tuned for specific tasks. CNN-based architectures allow for the automatic extraction of rich image features, making them highly effective for image comparison.

In recent years, transformer-based models have also been applied to image data. The Vision Transformer (ViT) [4] has shown promising results for image classification and similarity tasks by modeling image patches as sequences of tokens, similar to how transformers process text. Additionally, models like **DINO** [1] have introduced self-supervised learning methods to improve feature extraction, making these models particularly effective for unsupervised similarity tasks.

Multimodal Similarity Search Multimodal similarity search involves combining different types of data, such as text and images, to perform comparisons. In this context, the goal is to learn a common representation space where data from different modalities (e.g., textual descriptions and visual content) can be compared effectively.

Recent advances in multimodal learning have focused on learning joint embeddings for text and images. A prominent approach in this area is contrastive learning, where models are trained to align text and image representations in a shared space. For example, **CLIP** [9] is a model that learns joint representations of images and text by leveraging a contrastive objective. **CLIP** has been shown to perform exceptionally well on tasks such as zero-shot image classification and text-to-image retrieval.

3 Method

3.1 Text-based Representation

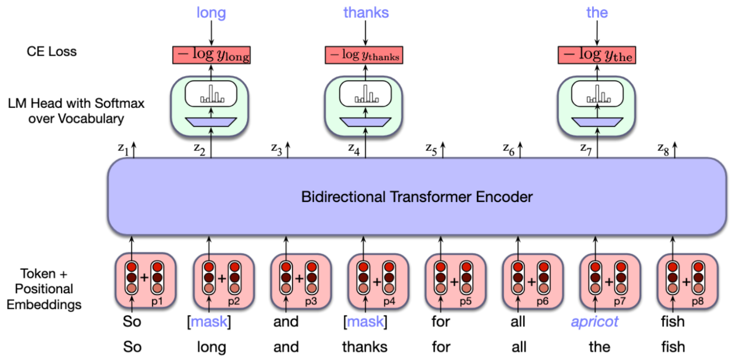

Figure 1: **BERT** Masked Token Training. *Image took from the "Machine Learning for NLP" course at ENSAE.*

Extracting meaningful information from textual data has been a challenge in previous years. The advent of transformer-based approaches was a huge breakthrough and allowed further progress.

BERT The model **BERT** [3] is a transformer-based bidirectional encoder with 100M parameters. It is trained by adding noise to the input sequence (mask, delete, insert tokens) and trying to recover the original sequence. It is equipped with a CLS token, which stores a learned representation of the entire sequence.

Thus, there are two ways of computing a dense representation for a sentence using a **BERT** model. The first is simply to use the embedding of the CLS token. The second is to compute a mean pooling on the entire sequence, using an attention mask to weight each token according to its importance in the sentence. Depending on the way the transformer has been trained, we will prefer one way or the other;

In our case, we need a representation for entire sentences. In fact, we have many informations (author name, title, description, history, etc.) that we would like to encode to represent the artwork. One way of doing so is to concatenate all these informations into a unique input sentence to feed to the model. However, many informations are noisy and just add confusion to the model (ids, links, references, etc.). We decided to focus only on a few major information: the author name, the title, and the description. The historical background of the artwork could also be interesting, but we found out that some data leakage could bias the search as it is sometime written when the piece of art has already been found. Thus, we focused on only a few features that we either concatenated to generate one common embedding, or generated an embedding per feature to compare with their corresponding feature in the search space.

In this project, the databases we are dealing with are in different languages. To compare them, we can either translate one or use a multilingual model to generate the embeddings. We chose this second option as we emancipate from the computational cost of translating everything. To perform our experiments, we compared different multilingual sentence **BERT** based models. A small compressed one, **MiniLM** [12], which maps sentences into a 384-dimensional embedding space. We also tried our experiments with **XLM-RoBERTa** [2], a larger model, based on **RoBERTa** [6] and specified in cross lingual comparisons. This one maps sentences into a 768-dimensional embedding space.

3.2 Image-based Representation

In recent years, Vision Transformers (ViTs) [4] have emerged as a powerful alternative to convolutional neural networks (CNNs) for image understanding tasks. Unlike CNNs, which rely on local receptive fields and hierarchical feature extraction, ViTs process images as sequences of patches and model global relationships directly using self-attention mechanisms. This allows them to capture long-range dependencies and contextual information more effectively, which is particularly valuable in tasks requiring a holistic understanding of the image. Although ViTs typically require more data and computational resources to train, they have demonstrated strong performance on various benchmarks once properly pretrained.

A Vision Transformer works by splitting an image into fixed-size patches (e.g., 16×16 pixels), which are then flattened and linearly projected into embedding vectors. Positional encodings are added to retain spatial information, and the resulting sequence is passed through a standard transformer encoder composed of alternating multi-head self-attention and feed-forward layers. The model processes the full image in parallel, learning global representations without relying on the inductive biases inherent to CNNs.

DINO In this project, we use the **DINO** (Self-Distillation with No Labels) model [1] to extract image embeddings in a self-supervised learning setting. **DINO** adopts a student-teacher framework in which both networks share the same architecture—typically a ViT—but receive different augmented views of the same image. The student is trained to match the output distribution of the teacher, whose weights are updated using an exponential moving average (EMA) of the student's weights, thus ensuring training stability. To avoid representation collapse (where the model outputs constant features), **DINO** uses normalization, centering, and sharpening techniques on the teacher outputs. A multi-crop training strategy is employed, with two global crops and several smaller local crops per image, encouraging the model to learn invariant and robust features.

More precisely, we use the **DINOv2-base** model [8] to generate image embeddings in a self-supervised setting. **DINOv2** improves over the original **DINO** by using better curated data and scaling both model size and training stability. It produces strong, general-purpose visual features that

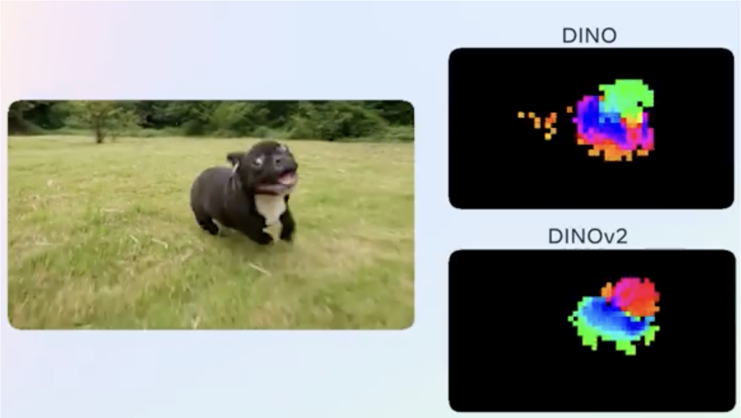

Figure 2: Segmentation example using **DINO** and **DINOv2**.

transfer well across tasks without finetuning. The version we use has around 22M parameters and was obtained by distilling a larger ViT trained on a diverse dataset of about 142M images. It maps images to a 768-dimensional feature space.

3.3 Similarity Score

We use a cosine similarity to compute the similarity score between two items. There are different ways of combining information from textual and image metadata. In fact, we can combine them after the computation of similarity scores based on text and image (respectively, s_{text} and s_{image}).

$$s = \beta s_{\text{text}} + (1 - \beta) s_{\text{image}} \tag{1}$$

Here β is the importance given to the text-based similarity score. Its value can be fixed, or learned. In our case, its value is fixed to 0.5, or 1 if the image comparison is not available. Indeed, we want to consider only the text and not penalize the similarity score in the latter.

Another way of computing the similarity would be using a common representation of the art work, combining both representations. This could be done by concatenating both representations or using a multimodal model, such as **CLIP** [9].

4 Experimental Setup

4.1 Data

The objective being to retrieve lost art works in French museums, we are provided with two different databases.

Lost Art The Lost Art Database tracks cultural property looted during Nazi persecution, mainly from Jewish owners, between 1933 and 1945, or items where such a loss is possible. It publishes Search Requests and Found-Object Reports to help reconnect former owners or their heirs with current owners and support fair resolutions. The database is entirely written in German and is composed of 38497 different artworks. There are 14 features in the database: Message Type, Record Type, Lost Art ID, Producer/Artist/Author, Title, Dating, Object Type, Inventory Number/Signature, Description, Provenance, Published Since, Contact, Link, Literature/Source. Some features are more important than others for the retrieval, like Title, Author and Description. However, there are many missing data, especially in descriptions (31% missing). The Link column allows for downloading the image of the artwork, when available.

This database will produce our queries for the similarity search.

Rose Valland (MNR) In 1997, the French Museum Service created the Rose Valland website (now the Rose-Valland Database, MNR-Jeu de Paume) to help search for and identify looted art. Museums that hold MNR objects are tasked with researching the history of the works and identifying their rightful owners. Most of the database, provided by the French Ministry of Culture, is written in French. There are too many features (62 columns), but a lot of them are useless for our task. We actually mainly focus on a few of them, such as title, author and description. Unfortunately, many descriptions are missing (55% missing), but many images are available (only 2% missing), which will be a great help for retrieval.

This database will be our search database with which we will compare our query.

Images On the website of the Rose Valland Database, we can easily find some images associated with each artwork (when available). They often have several images, but we are only interested in the main one, usually a photo of the front. However, you need to scrape each web page to access the image and then download it. The Lost Art database also has a link column, which takes you to a web page where you can find the image of the work of art. This image can also be retrieved to generate an embedding.

4.2 Evaluation

To evaluate our method, we rely on already found art works, provided by the French Ministry of Culture. We generate a query associated with an already found lost art and compare it with the MNR database. Then we can see the rank of the corresponding art, sorted by similarity. Our objective is to achieve a rank 0 - which means that the corresponding art in MNR is the most similar to the query - for every already found item in the Lost Art database. The length of our searching database being 2456, this is the worst rank a lost art can get, meaning we are not able to find it with our similarity-based approach. We are not provided with an extensive list of already found art and based our evaluation on only 9 samples.

4.3 Memory Usage

The export of each database is not very restrictive, as each is only a few MB. However, to get their corresponding image, we have to follow a link provided in the database, analyze the HTML source, and extract the image. This process and the final memory needed to store every image of the MNR database is very costly (more than 6GB of storage is needed). This could be done only once, compute every embedding and only store the embedding matrix, but in order to compare different vision models, we added a solution which streams the images in memory with BytesIO. The image is never written on the disk and is deleted as soon as we computed its representation. It solves the memory problem but increases the computing time.

4.4 Hardware Configuration

The experiments are carried out on a MacBook Pro with an Apple M1 chip, featuring 8 cores (4 performance cores and 4 efficiency cores) and 8 GB of unified memory, running macOS.

5 Results

Firstly, we tried to find lost art only using text-based methods. In fact, the access to images and their processing was not so clear, and we were not sure if we had access to all the images in the dataset. We used two different methods to generate a textual embedding of the art work:

- Concatenate all columns (title, author, description). Feed the model with this string and compare with the MNR database.
- Generate an embedding for each column. Then compare each feature separatly, and keep the mean of the similarity score. This reduces model's confusion and produces better results. We call this method "cross comparison".

The metric being evaluated is what we call the rank of the art work. This corresponds to its rank in the Rose Valland database, sorted by order of similarity. A rank 0 means that we have a matching, we found the lost art in the MNR database, while a high rank means that the model did not give a high similarity to the item we are looking for in the MNR database. We show in Figure 3 the distribution of ranks of our test set (of 9 lost art for which we have the correspondence).

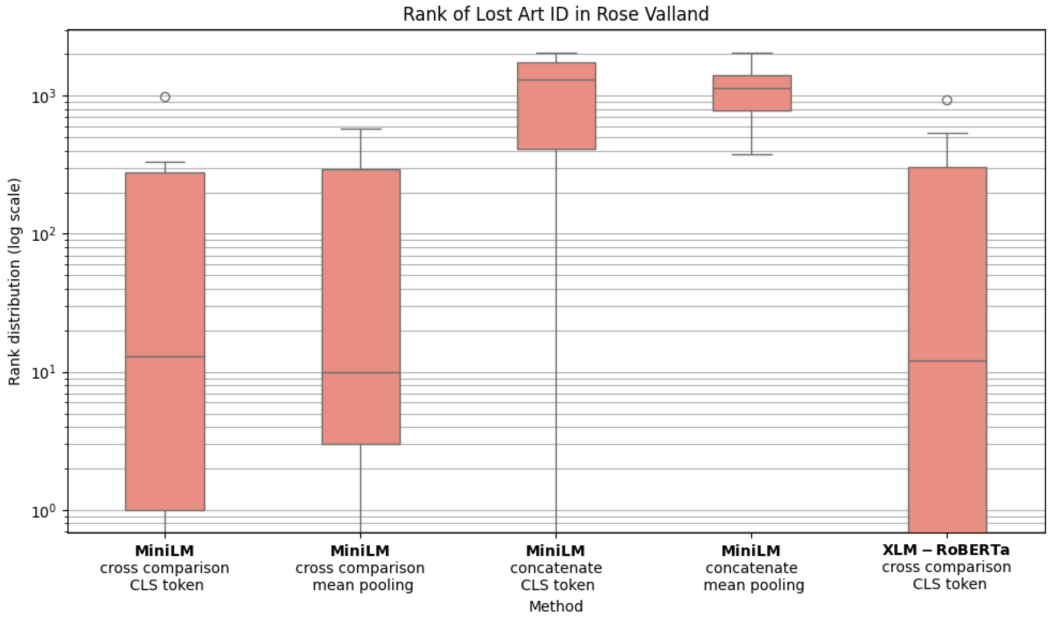

Figure 3: Comparison of performance for different text-based approaches. The y-axis is in log scale for better visibility.

We can see on Figure 3 that the performance of text-based method is pretty poor, especially when we concatenate every columns. For example, with **MiniLM**, cross comparison and the mean pooling calculation of embeddings, we have the following ranks of the test set, see Table 1.

Table 1: Ranks and similarities for Lost Art IDs in test set

Lost Art ID	Rank	Similarity
589707	3	0.579145
589708	295	0.400840
614072	0	0.843475
526702	0	0.515210
567247	10	0.458551
429210	3	0.579718
310418	213	0.401431
600027	354	0.460439
323038	574	0.362674

We decided to keep this text-based method, as it is on of the best in average, and do not have any outliers. **XLM-RoBERTa** could have been a good choice, but its computational cost is higher.

Then, we tried to add image similarities in the score. We first try to see what the rank would be with only images. The ranks appeared to be very low. The only problem is that for some lost arts, the corresponding art in the MNR database is not provided with an image. Thus, it is not possible to find it, and we penalize this case with a rank of 2456. Despite this drawback, the image-based similarity search is very powerful, and we derived a multimodal approach, with an averaged similarity score between text and image. We show the results of these experiments in Figure 4. We also show the rank of the test set, with $\beta = 0.1$, in Table 2.

Table 2: Ranks and similarities for Lost Art IDs in test set

Lost Art ID	Rank	Similarity
589707	0	0.678585
589708	0	0.836303
614072	0	0.846620
526702	0	0.794325
567247	1	0.725964
429210	0	0.847768
310418	213	0.401431
600027	0	0.789457
323038	310	0.362674

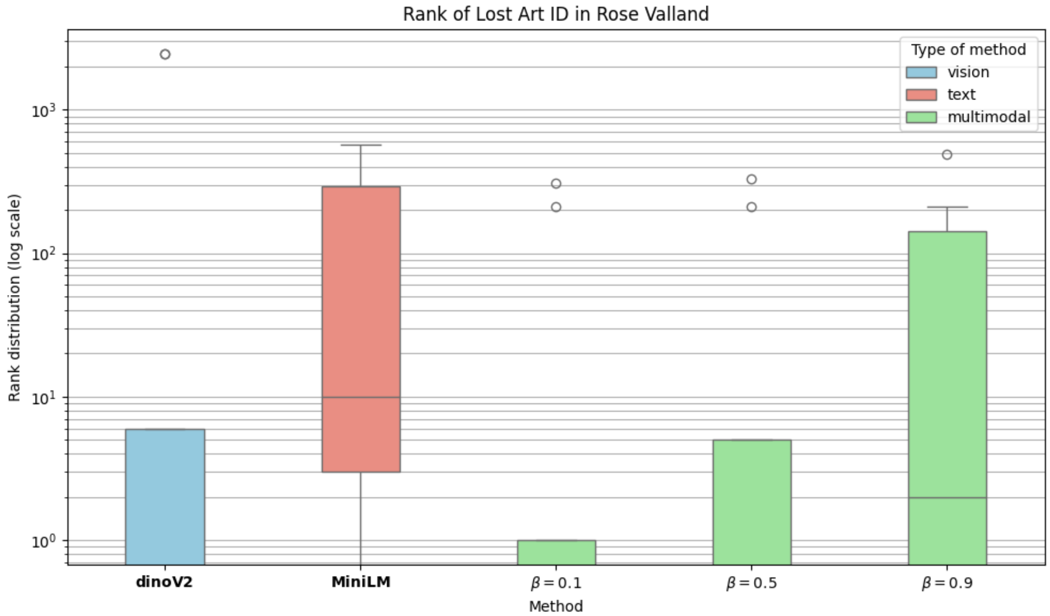

Figure 4: Comparison of performance for different multimodal approaches. The y-axis is in log scale for better visibility. The parameter β is the weight of the text similarity in the computation of the similarity score. The vision method being more reliable when images are available, we give a low weight to the text similarity. However, if the image is not available, we give full weight to the text similarity.

For both samples with a high rank, we discovered that there was no image in our MNR database when we launched the test, one description is missing, and both of them do not have an author name (Anonymous).

6 Conclusion

Our method, which relies on a language model and a vision model, is able to recover most of our lost artworks in the Rose Valland Database. Further work would have included more preprocessing of both databases, including allowing more columns to be taken into account. Moreover, we would have liked to develop other approaches, based on multimodal shared latent spaces such as **CLIP**, or contrastive learning techniques to train a model with our test set.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[7] Tomas Mikolov, Kai Chen, Geoffrey Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[9] Alec Radford, Jong Kim, Luan Xiao, and et al. Learning transferable visual models from natural language supervision. *Proceedings of the 2021 International Conference on Machine Learning (ICML)*, 2021.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[12] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.