UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA

SELF-SUPERVISED SKETCH-BASED DETECTION WITH APPLICATION IN HISTORICAL DOCUMENT SPOTTING

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIA DE DATOS

CHRISTOPHER ANDRÉS STEARS ROJAS

PROFESOR GUÍA:
JOSÉ SAAVEDRA RONDO

MIEMBROS DE LA COMISIÓN:
FELIPE BRAVO MÁRQUEZ
IVÁN SIPIRAN MENDOZA

SANTIAGO DE CHILE
2024

DETECCIÓN AUTOSUPERVISADA BASADA EN BOCETOS APLICADA A LA LOCALIZACIÓN EN DOCUMENTOS HISTÓRICOS

La digitalización es una herramienta fundamental para preservar y resguardar a la posteridad libros o documentos de patrimonio cultural, es por ello que se vuelve de vital importancia tener una herramienta capaz de buscar patrones y figuras a través de los distintos documentos. Las estrategias actuales se basan en la comparación de imágenes del mismo dominio (foto-foto) para detectar los distintos patrones en los documentos, pero su desempeño es limitado, alcanzando un *Mean Average Precision* (mAP) de 27,0% en la tarea de pattern spotting en el conjunto de datos DocExplore. Este trabajo propone una nueva aproximación que explora el uso de un dominio completamente diferente, específicamente bocetos, para detectar patrones en documentos de patrimonio cultural. Uno de los principales desafios al utilizar bocetos radica en la falta de pares foto-boceto para el entrenamiento, lo que dificulta el desarrollo de modelos generalizables. Para abordar esta limitación, se proponen dos modelos entrenados bajo un régimen auto-supervisado: S3BIR-CLIP y S3BIR-DINOv2 (donde S3BIR significa *Self-Supervised Sketch-based Image Retrieval*). Estos modelos son capaces de producir un espacio de características bimodal foto-boceto sin necesidad de datos emparejados explícitamente, demostrando un desempeño sobresaliente en tres conjuntos de datos públicos. Estos se integraron junto con un modelo de segmentación conocido como SAM (*Segment Anything Model*) para extraer regiones de interés dentro de los documentos y ser evaluados en el dataset DocExplore bajo la tarea de pattern spotting. Los resultados mostraron que esta propuesta es competitiva a la hora de detectar patrones dentro de los documentos, alcanzando un mAP del 21,0%. Este hallazgo ofrece nuevas oportunidades para los expertos dedicados a la preservación y análisis de documentos históricos, ya que permite el uso de bocetos a la hora de buscar información relevante, facilitando así la interacción con el patrimonio cultural digitalizado.

Abstract

Digitization is a fundamental tool for preserving and safeguarding books or cultural heritage documents for posterity, which is why it is of vital importance to have a tool capable of searching for patterns and figures through the different documents. Current strategies are based on the comparison of images from the same domain (photo-photo) to detect the different patterns in the documents, but their performance is limited, reaching a *Mean Average Precision* (mAP) of 27.0% on the pattern spotting task on a DocExplore dataset. This paper proposes a new approach that explores the use of a completely different domain, specifically sketches, to detect patterns in cultural heritage documents. One of the main challenges in using sketches lies in the lack of photo-sketch pairs for training, which hinders the development of generalizable models. To address this limitation, two models trained under a self-supervised regime are proposed: S3BIR-CLIP and S3BIR-DINOv2 (where S3BIR stands for Self-Supervised Sketch-based Image Retrieval). These models are capable of producing a bimodal photo-sketch feature space without the need for explicitly matched data, demonstrating outstanding performance on three public datasets. These were integrated together with a segmentation model known as SAM (*Segment Anything Model*) to extract regions of interest within documents and evaluated on the DocExplore dataset under the pattern spotting task. The results showed that this approach is competitive in detecting patterns within documents, achieving a mAP of 21.0%. This finding offers new opportunities for experts dedicated to the preservation and analysis of historical documents, as it allows the use of sketches when searching for relevant information, thus facilitating the interaction with the digitized cultural heritage.

A mis padres y hermana, gracias por todo.
A mis mascotas Chanel, Naksu, Haru y Miku.

Table of Content

1 Introduction **1**

1.1 The Problem 2

1.2 Objectives 3

1.2.1 Main Objectives 3

1.2.2 Specific Objectives 3

1.2.3 Thesis Structure 3

2 Preliminaries **5**

2.1 Theoretical Framework 5

2.1.1 Basic Definitions 5

2.1.2 Advances in Deep Learning Models 8

2.1.3 Convolutional Neural Networks 8

2.1.4 ResNet 10

2.1.5 Attention Models 11

2.1.6 Transformers 13

2.1.7 Vision Transformer (ViT) 14

2.1.8 Contrastive Learning 15

2.1.9 Self-Supervised Learning 16

2.1.10 CLIP 17

2.1.11 SAM 19

2.1.12 DINOv2 21

2.2 Literature Review 22

2.2.1 Sketch-based Image Retrieval 22

2.2.2 Self-Supervised Learning 23

2.2.3 Sketch-base Detection 24

2.2.4 Related work 25

3 Self-Supervised SBIR 27

3.1 Related 28

3.1.1 Self-Supervised SBIR (S3BIR) 28

3.2 S3BIR-(CLIP/DINOv2) 28

3.3 Experimental Setting 29

3.3.1 Datasets 29

3.3.2 S3BIR Models 31

3.3.3 Settings 32

3.4 Experimental Results 32

3.4.1 Quantitative Results in SBIR 32

3.4.2 SBIR Qualitative Results 33

4 Sketch-based One-shot Detection 37

4.1 Exploratory Data Analysis 37

4.2 Data Collection and Preprocessing 40

4.3 Proposals 40

4.3.1 Base Proposal 41

4.3.2 SAM Proposal 43

4.3.3 SAM + Clusters Proposal 43

4.4 Experimental Results 45

4.4.1 Quantitative Results 45

4.4.2 Qualitative Results 47

5 Discussion **51**

6 Conclusion **55**

6.1 Future Work 56

Bibliography **62**

ANNEXES Annex A **63**

A.1 Metrics 63

A.1.1 Precision 63

A.1.2 Recall 63

A.1.3 mAP (mean Average Precision) 64

A.1.4 Cosine Similarity 64

A.2 Average Precision of the best two models by class 65

A.3 Qualitative Examples of SAM + Proposed Clusters 69

A.4 Results achieved 71

List of Tables

3.1 Quantitative description of the used datasets for SBIR-based evaluation. . . 31

3.2 S3BIR-CLIP mAP with different loss functions. 33

3.3 Performance of the studied self-supervised SBIR models on diverse datasets. 33

4.1 The mAP of the image retrieval and pattern detection results of the S3BIR models in the different proposals is reported. 45

4.2 Performance (mAP) of the S3BIR-DINOv2 (QD-Extended) model in Image Retrieval and Pattern Spotting. 46

4.3 Performance (mAP) of the S3BIR-DINOv2 (Flickr25K) model in Image Retrieval and Pattern Spotting. 46

4.4 The table shows the number of points used in each of the proposed solutions, together with the corresponding number of segmentations performed and the memory occupied (in megabytes) by the embeddings. 46

5.1 Results of the Image Retrieval and Pattern Spotting task on the DocExplore dataset. Values refer to the mAP of the 1000 best candidates. 51

A.1 Results in Image Retrieval and Pattern Spotting by query of the S3BIR-DINOv2 (QD-Extended) model on the SAM proposal. 67

A.2 Results in Image Retrieval and Pattern Spotting by query of the S3BIR-DINOv2 (Flickr25K) model on the SAM proposal. 69

List of Figures

1.1 Example of cultural heritage documents. This excerpt was taken from the *dataset* DocExplore [38]. Yellow boxes indicate the location of patterns relevant to historians. 1

2.1 A sample of the sketches used as query. 6

2.2 Domain types for the use of *queries*. Image taken and modified from [64]. . . 6

2.3 The diagram illustrates the two main phases of the CBIR system: *offline* and *online*. Image taken from [64]. 7

2.4 Rankings using a sketch as query. Sorted from highest (left) to lowest (right) similarity. 7

2.5 A convolutional operation is performed between the input (on the left) and two kernels (in the center), resulting in two feature maps (on the right). . . . 9

2.6 The diagram illustrates an example of a CNN architecture. The feature extraction stage is depicted on the left, while the classification stage is shown on the right. 10

2.7 Residual blocks. Image taken from [27]. 11

2.8 The diagram illustrates a basic seq2seq architecture with a context vector. . 11

2.9 Alignment matrix between equivalent phrases in different languages, highlighting the importance given to each word during translation. Image taken from [1]. 12

2.10 Attention mechanism example. 12

2.11 Scaled dot product attention and multi-head attention graphs. Image taken from [59]. 13

2.12 Transformer model architecture. Image taken from [59]. 14

2.13 General description of Vision Transformer model. Image taken from [15]. . . 15

2.14 A framework used for contrastive learning in SimCLR. Image taken from [7]. 17

2.15 SimSiam architecture. Image taken from [8]. 17

2.16 CLIP Architecture. Image taken from [43]. 18

2.17 Segment Anything Model architecture. Image taken from [31]. 19

2.18 Image encoder architecture. 20

2.19 Schematic of the prompt encoding, in particular the point encoding for the image segmentation. 20

2.20 Mask decoder architecture. Image taken from [31]. 21

2.21 Architecture of self-distillation with no labels. Image taken from [5]. 21

3.1 The S3BIR-CLIP training methodology. The images with green borders represent the pseudo-sketch/image of the positive pair. Conversely, the red border will serve as the negative image to be utilized in conjunction with the triplet loss. 30

3.2 Recall-Precision curve on the eCommerce dataset. S3BIR-DINOv2 and S3BIR-SimCLR show similar behavior, superior to the performance of S3BIR-CLIP for the first recall values. 34

3.3 Recall-Precision curve of the Flickr15K dataset. Here, the superiority of S3BIR-DINOv2 is more visible. 34

3.4 Examples of sketch-based image retrieval using our proposals (S3BIR-DINOv2 and S3BIR-CLIP) in the eCommerce dataset. 35

3.5 Examples of sketch-based image retrieval using our proposals (S3BIR-DINOv2 and S3BIR-CLIP) in the Flickr15K dataset. 36

4.1 Number of occurrences of each category in the dataset. 38

4.2 The first boxplot shows the heights of the pages. The second shows the widths of the pages. Finally, the last plot shows the total area of the pages in logarithmic scale. 38

4.3 Pages of documents that are in poor condition or do not contain relevant information. 39

4.4 Green dots indicate sections of the document that do not contain information that is important to know. 39

4.5 On the left is a bar chart analyzing the frequency of queries by size. On the right is a boxplot analyzing the distribution by size based on area (in logarithmic scale). 39

4.6 Handmade sketches of the different classes presented in DocExplore. 40

4.7 Preprocessing followed to remove the background of the documents. 40

4.8 General diagram of the procedure followed by the proposals. 41

4.9 Process to obtain the image patches together with the classification token. . 42

4.10 Exemplification of the correlation between a sketch and a target. From left to right is presented: 1) Hand-drawn sketch representing a bateau class query. 2) Image containing the searched query. 3) Heat map generated by the query on the document patches, where red color indicates high correlation, followed by yellow and finally blue representing low correlation. 4) Selected areas with higher correlation and their respective black bounding boxes. 42

4.11 General scheme of the proposal using SAM. 43

4.12 Results achieved in the clustering process. From left to right is presented: 1) The original input document. 2) The clustering result on the similarity matrix between patches. 3) The points generated on the highest intra-variance cluster. 4) The regions proposed by SAM. 44

4.13 Examples of sketch-base image retrieval using S3BIR-DINOv2 (QD-Extended) using the SAM approach. 47

4.14 Examples of sketch-base image retrieval using S3BIR-DINOv2 (Flickr25K) using the SAM approach. 48

4.15 Examples of sketch-base detection using S3BIR-DINOv2 (QD-Extended) using the SAM approach. The green bounding box represents the detection based on the query. 49

4.16 Examples of sketch-base detection using S3BIR-DINOv2 (Flickr25K) using the SAM approach. The green bounding box represents the detection based on the query. 50

5.1 The points generated by the clusters approach are not sufficiently accurate to permit the selection of some regions on the right, and thus SAM is unable to segment them. 52

5.2 The generated points fit exactly on the patterns being searched for, allowing SAM to segment correctly. 53

5.3 Sketches of varying complexity and detail for the class “Marker”. 54

A.1 SAM + Cluster approach results with prompting. 69

A.2 SAM + Cluster approach results with prompting. 70

Chapter 1

Introduction

The preservation and study of cultural heritage is relevant to the history and identity of a country. In order to understand and preserve this heritage, it is essential to develop innovative approaches that allow the documentation and analysis of the complex visual designs present in cultural heritage objects and documents.

Although several studies have focused on the detection of archaeological objects and historical documents for the purpose of identifying specific figures and/or shapes in images or documents [20, 61, 9, 65], it has not yet been possible to train an object detection model capable of recognizing and locating all these complex shapes without having seen them before, regardless of the cultural heritage under analysis. This is due to the large variability in pattern sizes and the inherent complexity of these objects.

Such complex patterns and designs are present in archaeology, books and cultural heritage documents, which once extracted rescue all possible information, and then make them available to conservation institutions, and in many cases digitized on the Internet. However, a not insignificant work lies in the categorization of these objects, that is, in obtaining the characteristics of experts to have more information about the patterns and designs. An example of this is shown in the figure 1.1.

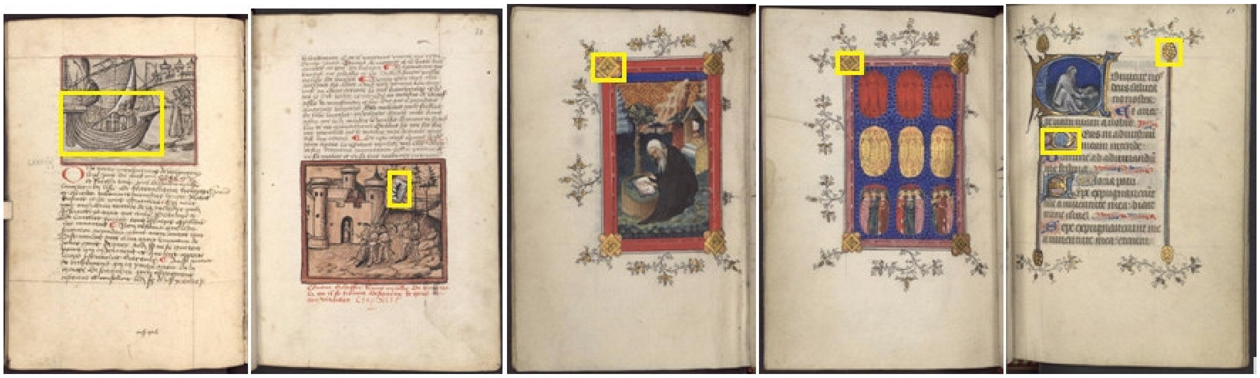


Figure 1.1: Example of cultural heritage documents. This excerpt was taken from the *dataset* DocExplore [38]. Yellow boxes indicate the location of patterns relevant to historians.

The field of computer vision has made significant progress in recent years, making it possible to analyze large sets of images in a matter of seconds and obtain very good representations of them, thanks to architectures such as Transformers, as demonstrated in [59, 31, 37, 56, 58]. These techniques have exhibited exceptional performance in object detection and classification tasks, even when applied to classes and objects that were not included in the training phase (zero-shot). This performance is supported by the large datasets used in their training, an example of which is SAM [31], a model that will be explained in more detail in the section 2.1.11. In this sense, Sketch-Based Image Retrieval (SBIR) models are no exception to the good results. Indeed, these models are now capable of recognizing a wide variety of objects from sketches, as documented in [48, 6]. However, despite these advances, SBIR faces a significant challenge: the limited availability of paired sketch-image datasets, which limits the potential for further improvements in this area.

Therefore, the primary objective of this research is to develop a deep learning-based model that is capable of searching within an image for the sketch created by an expert. For this purpose, a self-supervised approach is used to mitigate both the existing domain gap and the scarcity of image-sketch data pairs. A segmentation model (SAM) is also incorporated to locate the different shapes and patterns within an image. This combination of techniques is contrasted with the various existing proposals through the same program provided by DocExplore [38].

1.1 The Problem

Currently, most object detection models are based on datasets that share the same domain as the recognition objective, which makes sense and is a well-supported approach [24, 44, 27]. However, what would happen if we want to use a totally heterogeneous domain to detect objects, such as sketches?

Sketches, by their graphic nature, possess the ability to capture and transmit the visual richness of objects, preserving essential elements that reflect the distinctive characteristics of a particular object [57]. Sketch-based understanding is a key process in visual perception. During the dawn of artificial intelligence, Hubel and Wiesel [30] showed how the animal visual cortex highly responds to edge stimuli. In our century, Walther et al. [60] also showed the semantic power of image contour information using functional MRI.

Sketching is a natural and primitive means of communication; it is how our ancestors transmitted ideas, stories, and activities that have allowed us to learn from our past. Indeed, much of what we know about our history is due to the drawings humans engraved in caves or on rocks and pottery [28]. Sketch understanding is also profoundly connected to cognition development [23]; it is how an infant starts to understand the world and enables people to understand complex structures and unfold complex processes.

The incorporation of a domain that presents a significant gap compared to natural images poses a considerable challenge for object detection, especially when dealing with images and sketches related to cultural heritage, which are often scarce. For this reason, various datasets are used with the aim of addressing this problem and achieving the detection of the diverse patterns and designs present in cultural heritage objects/documents, even when they have not been seen previously (one-shot detection).

1.2 Objectives

1.2.1 Main Objectives

The main objective is to develop a deep learning model that, based on a sketch, can identify and locate the desired patterns within the cultural heritage objects/documents, with the aim of contributing to the automation of the labeling and documentation process.

1.2.2 Specific Objectives

- Define a deep learning architecture that employs both natural images and sketches to detect cultural heritage objects. This architecture will primarily be based on the SAM approach, which will be used to obtain an initial segmentation of all patterns and objects seen in the documents and/or images.
- Evaluate the model defined in the DocExplore dataset. The dataset will provide a variety of relevant test cases, allowing for the measurement of the models ability to accurately detect cultural inheritance objects in different documents and scenarios.
- Compare the proposed models with the state of the art in pattern detection and image retrieval in historical documents.

1.2.3 Thesis Structure

The second chapter presents the theoretical framework. First, it defines the terms that will be used throughout the research. Subsequently, it discusses the concepts of deep learning, specifically the techniques and models used in sketch-based detection and sketch-based image retrieval. Finally, it reviews the state of the art in the area and presents a summary and conclusion.

The third chapter provides a comprehensive analysis of self-supervised learning. First, it highlights the importance of having models capable of generating a bimodal feature space (image sketch) to achieve optimal performance in SBIR. Subsequently, the architectural used to address this challenge and the datasets used to train the models are elucidated. Finally, the results of the proposed models are presented.

The fourth chapter is dedicated to the topic of sketch-base one-shot detection. The chapter begins with a brief overview of the paradigm, followed by an exploratory data analysis of the Docexplore data. Next, the three proposed solutions to the problem are presented. Finally, the results of the three proposed solutions are presented.

The fifth chapter discusses and analyzes the results. Finally, the sixth chapter presents the conclusions derived from the research, suggests improvements and discusses future work in this area.

Chapter 2

Preliminaries

This chapter discusses the fundamental concepts and techniques related to object detection, divided into two main sections: the first section (2.1) addresses the state of the art in the field, providing an overview of recent advances. Then, in the second section (2.2), the literature review will be conducted, delving into the most significant contributions to this research.

2.1 Theoretical Framework

Throughout this thesis, we will see various terms that will be used in future chapters. To avoid confusion, we will define and specify them.

2.1.1 Basic Definitions

Target Image

In this research, the search for the desired object or pattern is performed on what is called the “target image”, also known as the natural image. This image represents a color and digital visual version of reality, containing a variety of patterns in different positions and sizes, as shown in figure 1.1. It is in this context that the task of searching for the specific object or pattern that the experts want to locate is performed.

Query Sketch

In the context of this research, the query sketch, also known as the query, is presented as a sketch containing essential features about the shape or pattern of the object or figure being searched within the target image. Figure 2.1 illustrates some examples of sketches.

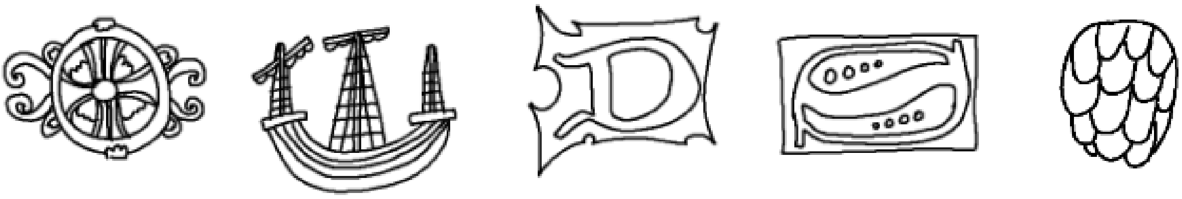


Figure 2.1: A sample of the sketches used as query.

Image Retrieval

The field of image retrieval is a subcategory of information retrieval. In this process, the retrieval system receives as input a query that specifies certain criteria, characteristics, or references of the desired content. The query is then compared with the database using a similarity criterion, with the objective of classifying them according to an order. In this way, it attempts to identify and present in an orderly fashion those elements of the database that best meet the criteria established by the query.

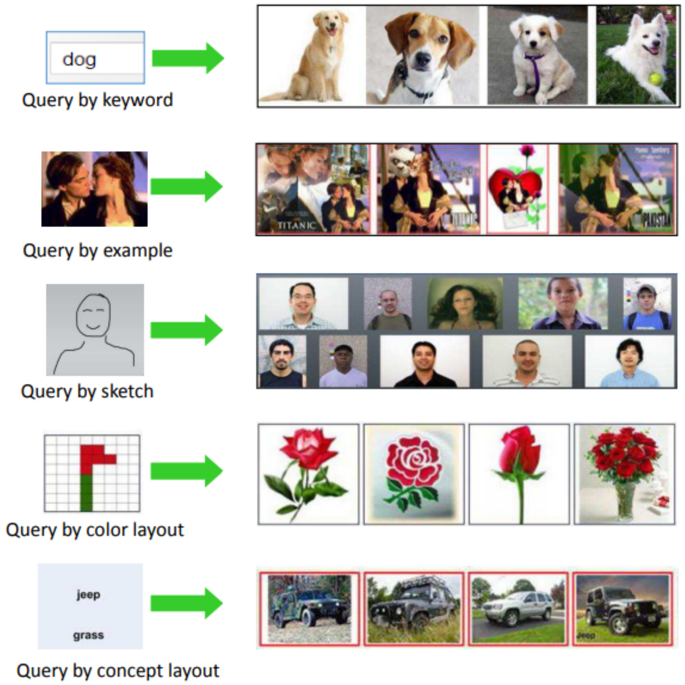


Figure 2.2: Domain types for the use of *queries*. Image taken and modified from [64].

Figure 2.2 illustrates the different domains of image retrieval, showing how this process can be applied in different contexts. One of them is sketch-based image retrieval, where the query is a sketch. Typically, the ranking of results is defined using similarity-based techniques, such as cosine similarity .8, to rank the retrieval according to the degree of similarity between the query sketch and the images in the database.

The motivation for retrieving images rather than other types of data, such as text, is that images do not require tags or metadata information; the visual content of the image is sufficient for retrieval. We refer to this problem as content-based image retrieval (CBIR), a field that has been the subject of research for decades [52, 33].

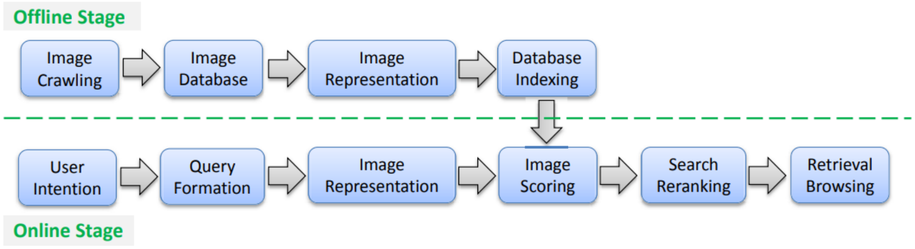


Figure 2.3: The diagram illustrates the two main phases of the CBIR system: *offline* and *online*. Image taken from [64].

The figure 2.3 shows a typical CBIR flowchart, illustrating the steps involved in it. In the offline phase, a database is built by systematically collecting images. For each image in this database, features are extracted to obtain its representation (descriptive vectors). It is important to note that the vector representation can vary according to the methodology implemented. Finally, these vectors are used to index the data.

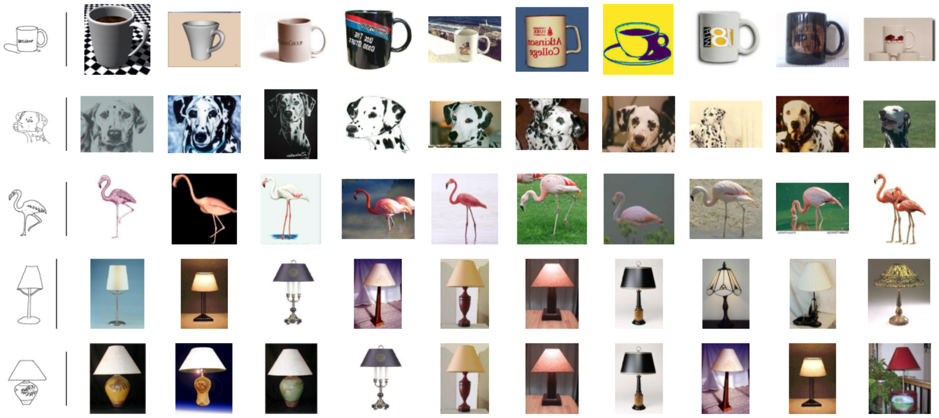


Figure 2.4: Rankings using a sketch as query. Sorted from highest (left) to lowest (right) similarity.

On the other hand, in the online phase, the system receives a query from which its vector representation is generated. Then, a similarity score is calculated between this representation and the previously indexed database. Based on these results, a similarity ranking is established to finally return the results. See figure 2.4, which shows the result by applying the 2.3 diagram using a sketch as a query.

Sketch-based Image Retrieval (SBIR)

As mentioned above, it is possible to receive a sketch as query input, see figure 2.2. Like CBIR, the process for Sketch-based Image Retrieval follows the same logic, where it is crucial to obtain a good representation of the sketch for later comparison with the indexed natural images.

The proposed solution makes use of SBIR, which is a fundamental component. Therefore, an extensive section in chapter 3 is dedicated to the explanation of this task.

2.1.2 Advances in Deep Learning Models

This section sets forth the theoretical foundations necessary for a proper understanding of the proposed methodology. However, it is assumed that the reader has a basic knowledge of feedforward neural networks. This knowledge will serve as a basis for more advanced concepts to be presented in later sections.

2.1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) emerged as a solution to a problem that Multilayer Perceptrons (MLP) could not solve efficiently. The concept was introduced in 1989 in [32] with the use of backpropagation. Convolutional neural networks are a more structured version of MLPs, characterized by fewer synaptic connections and specifically designed to process information in matrix form, such as images.

Convolutional neural networks employ the mathematical operation of convolution, represented by the equation 2.1, which describes the interaction between two features, for example, between x and w. In this expression, x is referred to as the input, w corresponds to the kernel (or weights), and the output s is known as the feature map.

$$s(t) = (x * w)(t) = \int_{-\infty}^{\infty} x(a)w(t-a)da \tag{2.1}$$

The formalization of the discrete domain is expressed as follows:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.2}$$

When working in the context of images, i.e., on multidimensional arrays that we will call tensors, the convolution is performed on more than one axis simultaneously. Therefore, when using an image I as input, a two-dimensional K kernel must be applied, as shown in the following expression:

$$s(i,j) = \sum_{m}\sum_{n} I(i+m, j+n)K(m,n) \tag{2.3}$$

The expression "cross-correlation" is thus defined as the calculation of a feature map at position (i, j) by applying the Hadamard product between K and a section of the image I.

This process is repeated for all possible positions in the spatial dimensions of the image. This property, known as parameter sharing, implies that the same kernel weights are used for each spatial dimension of I.

It is of significant importance to note that the kernel serves the function of learning the weights that allow it to extract the most relevant features from the image I. This process contributes to the convolutional neural network being able to identify different patterns and representations within the images in its training process.

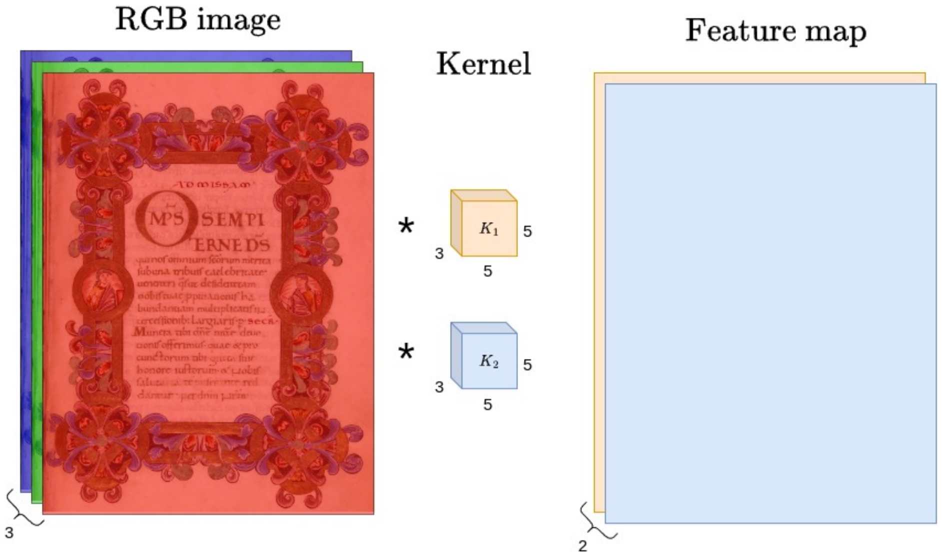


Figure 2.5: A convolutional operation is performed between the input (on the left) and two kernels (in the center), resulting in two feature maps (on the right).

In practice, a convolutional layer is formed by a series of convolution operations, each applying a different kernel to the same input in parallel. This process is illustrated in figure 2.5. In this example, the input is an image with three color channels (RGB), thus necessitating a kernel with three channels to apply the convolution.

Following each convolutional operation, a nonlinear activation function is applied, such as the hyperbolic tangent function (tanh), sigmoid, rectified linear unit (ReLU), and so forth, with the objective of introducing nonlinearity into the model.

Another common layer utilized in CNNs is the pooling layer. The objective of these layers is to reduce the spatial dimensions of the feature map, thereby reducing the number of parameters and preventing overfitting. Two types of pooling are distinguished: max pooling and average pooling. The former selects the maximum value, whereas the latter selects the average value. Both types of pooling maintain the depth of the feature map.

On the other hand, the padding comes to solve possible problems with the kernel and the dimensions of the image I, since it allows to add additional pixels to the edge of the image, thus avoiding the loss of information at the edges and allowing the kernel to “slide” over the entire image.

After repeating the above layers, the generated feature maps become progressively smaller,

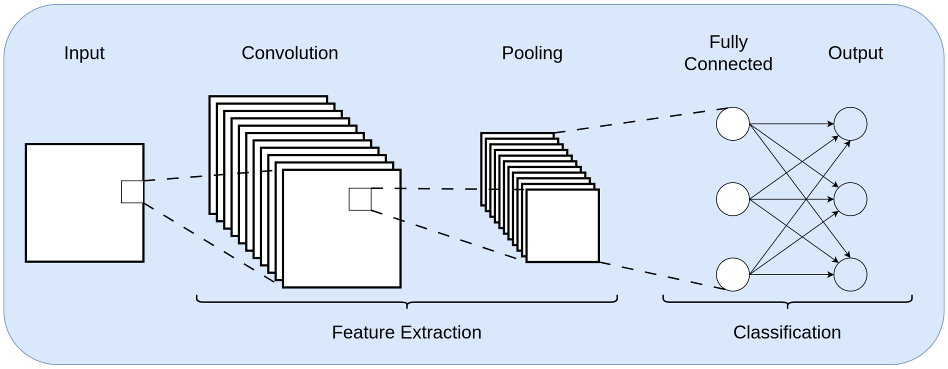


Figure 2.6: The diagram illustrates an example of a CNN architecture. The feature extraction stage is depicted on the left, while the classification stage is shown on the right.

resulting in kernels that tend to cover more of the image to capture features with higher semantics. In addition, at the end of these layers, it is possible to add a fully connected layer followed by a softmax layer to perform image classification, as shown in figure 2.6. The whole pre-classification process is then called feature learning or backbone, where the neural network learns to extract relevant and hierarchical features from the images.

The feature learning phase plays a foundational role in the transfer learning process. It allows the network to acquire prior knowledge in one domain and then adapt it to a completely different one. This process is performed exclusively by adjusting the parameters of the classification head, known as fine-tuning. This approach will be utilized in this research.

2.1.4 ResNet

With the increasing popularity of convolutional neural networks, they became the norm when working with images. Over time, architectures were developed that concatenated large numbers of convolutional blocks, where He et al. [27] proposed a solution to the problem faced when working with very deep networks.

Specifically, the authors found that working with a large number of convolutional blocks increased the error in the training phase. This problem was due to gradient vanishing. In the paper entitled “Deep Residual Learning for Image Recognition” [27], the authors propose the use of residual blocks as a solution, as shown in the figure 2.7. In detail, by having an output of the form $\mathbb{F}(x) + x$, where $\mathbb{F}(x)$ represents a nonlinearity and x the input, this sum allows the gradient to flow directly to the input (shortcut connections), thus avoiding gradient vanishing and solving one of the main problems encountered when working with deep convolutional neural networks. This network has different versions depending on the total number of layers in its architecture, whose name varies according to the number, such as ResNet-34, ResNet-50, ResNet-101, and ResNet-152.

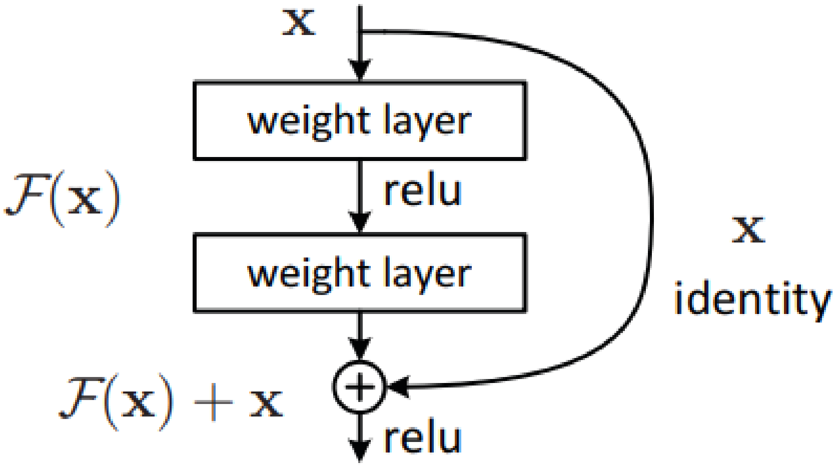

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.7: Residual blocks. Image taken from [27].

2.1.5 Attention Models

Sequence-to-sequence (Seq2Seq) learning models are those that consist of two main components: an encoder and a decoder. The encoder receives an input sequence from one domain, such as text, and converts it into an internal representation, usually called a context vector. The decoder then takes this context vector to generate an output sequence in another domain, such as its translation [54].

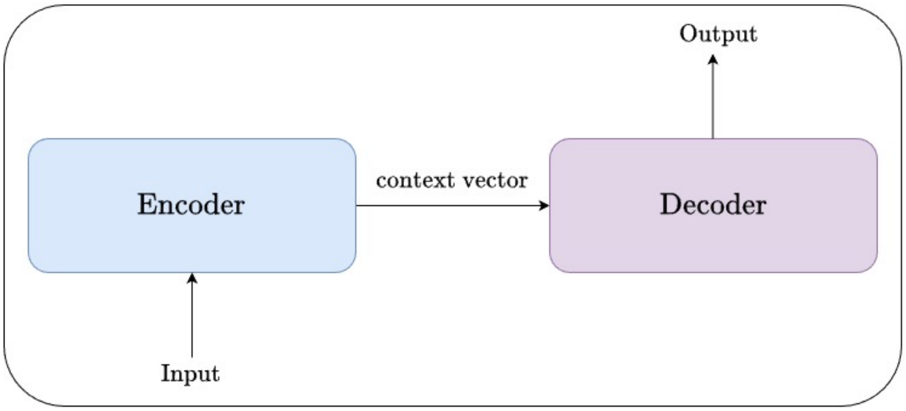


Figure 2.8: The diagram illustrates a basic seq2seq architecture with a context vector.

When dealing with long and complex sentences, the context vector between the encoder and the decoder becomes insufficient to capture all the relevant information needed [1]. To address these problems, attention mechanisms have been introduced that allow the model to dynamically select relevant information from the input stream based on the current context. This allows the model to handle sentences of different lengths and to capture the relationships between words in a sentence, see figure 2.9.

The attention mechanism presented in the paper entitled "Attention is All You Need" by Vaswani et al. [59] (2017) introduces the Transformer architecture, which employs attention as its primary mechanism to model the relationships between words in a sentence. This

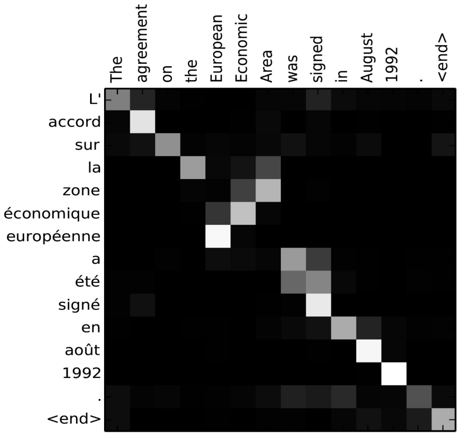


Figure 2.9: Alignment matrix between equivalent phrases in different languages, highlighting the importance given to each word during translation. Image taken from [1].

approach has proven to be efficient in both Natural Language Processing (NLP) tasks and vision, as will be detailed below.

The attention used in [59] is represented by a query and a key-value set, where both the query and the key-value set are represented as vectors. The output is calculated as a weighted sum of the values, with the weight assigned to each value determined by a score function with the corresponding key. This is illustrated in figure 2.10.

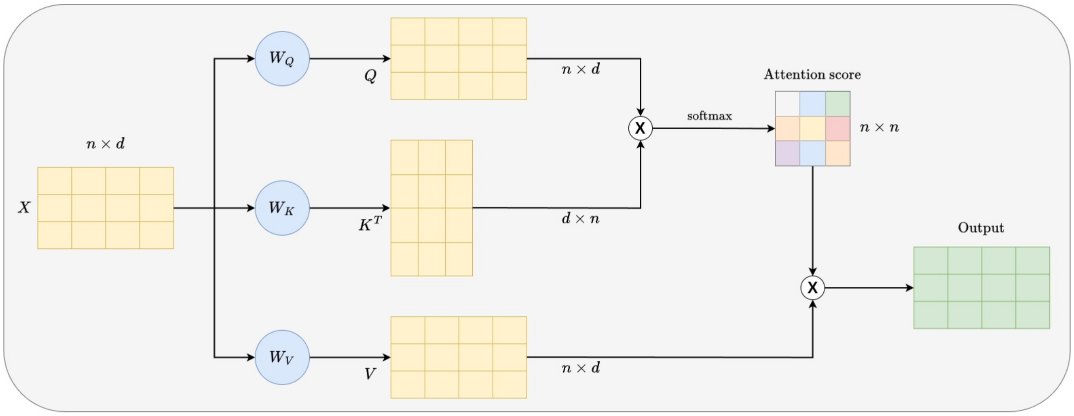


Figure 2.10: Attention mechanism example.

The general formula for calculating the output of the attention mechanism, designated as “Scaled Dot-Product Attention”, is expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{2.4}$$

where Q, K, and V are linear transformations over the input X, as shown in figure 2.10. That is, $Q = XW_Q$, $K = XW_K$, and $V = XW_V$, where the matrices W_Q, W_K, and W_V contain the values that the model will learn. The product between Q y K^T relates these transformations through the dot product, a result that represents the similarity between vectors. This product generates a matrix of “n” representing the weight (relevance) of each representation with respect to the others. These weights are transformed by applying the softmax function to obtain a probability distribution. The term $\sqrt{d}$ is the normalization factor with respect to the dimensionality of the data. Finally, the result of softmax is multiplied by V, generating a matrix of $n \times d$, which will be the new representation generated by the attention mechanism.

The multihead attention is a concept used in [59] where multiple attention mechanisms or heads are integrated into a single block. This approach aims to avoid the use of higher dimensional matrix projections such as W_Q and W_K. Each head operates independently and processes the data set in parallel, as shown in figure 2.11. The results of all heads are concatenated into a single vector and weighted by the W^O matrix, which contains adjustable parameters and allows the assignment of weights to each resulting block. This operation is expressed as:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, ..., \text{head}_h\right) W^O \\ \text{where } \text{head}_i &= \text{Attention}(Q, K, V) \end{aligned} \tag{2.5}$$

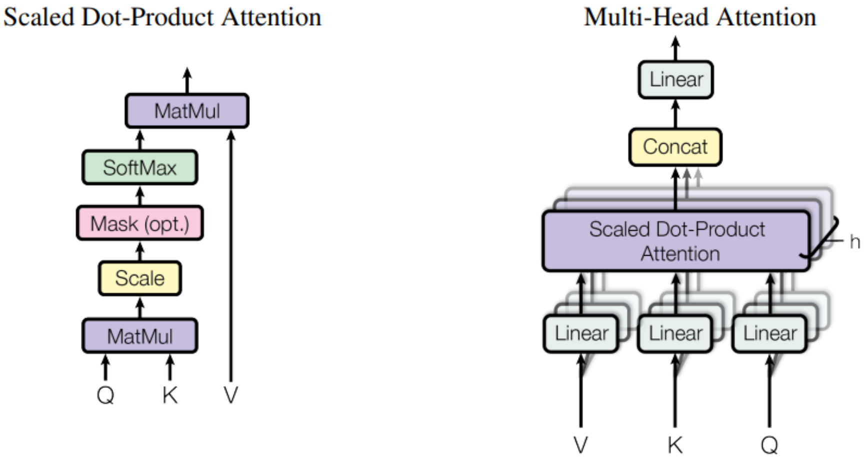


Figure 2.11: Scaled dot product attention and multi-head attention graphs. Image taken from [59].

2.1.6 Transformers

The transformer architecture proposed by Vaswani et al. [59] was originally intended for processing word sequences, specifically for automatic text translation. As discussed in the “Attention” 2.1.5 section, it consists of the “Scaled Dot-Product Attention” and “Multi-Head Attention” methods. Figure 2.12 shows the architecture in detail, where it can be seen that it is of the “Seq2Seq” type, divided into two parts: encoder and decoder.

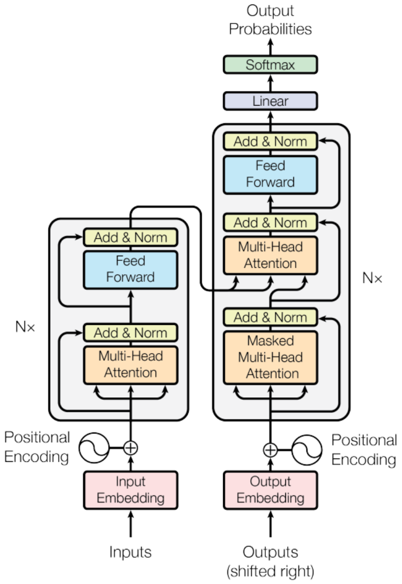


Figure 2.12: Transformer model architecture. Image taken from [59].

The encoder is designed to process the entire input sequence. Initially, the multi-head self-attention mechanism is applied, followed by a two-layer MLP with a ReLU nonlinearity. Between these, there is a residual connection, followed by normalization. Finally, the encoder output is transformed into two attention vectors, K and V, which are used in the decoder.

In contrast, the decoder objective is to learn to focus on tokens that are present at a specific point in time. This is because the inference process does not have access to future information. The masked multi-head attention module addresses this issue by conditioning the network on future positions, thereby generating attention exclusively based on its past. Subsequently, all the information passes through the multi-head attention module, where the encoder generates the keys and values as input, and the decoder generates the queries. It should be noted that each attention block contains residual connections. Finally, the feed-forward module (comprising two linear layers and a ReLU activation function) is employed, which is then connected to a multilayer perceptron (MLP) in order to obtain the respective probability distribution, in this case, of the words to be translated.

It is crucial to note that, in contrast to recurrent models, the temporal order intrinsic to these networks is not preserved. For this reason, the “Positional Encoding” is employed, which incorporates positional information to ascertain the location within the sequence where the attention is focused. The authors proposal [59] is based on the use of sine and cosine functions of different frequencies.

2.1.7 Vision Transformer (ViT)

The Vision Transformer (ViT) represents a paradigm shift in computer vision. Introduced by Dosovitskiy et al. [15], its architecture is based on the original Transformer described above 2.1.6. Figure 2.13 shows a general schematic of ViT.

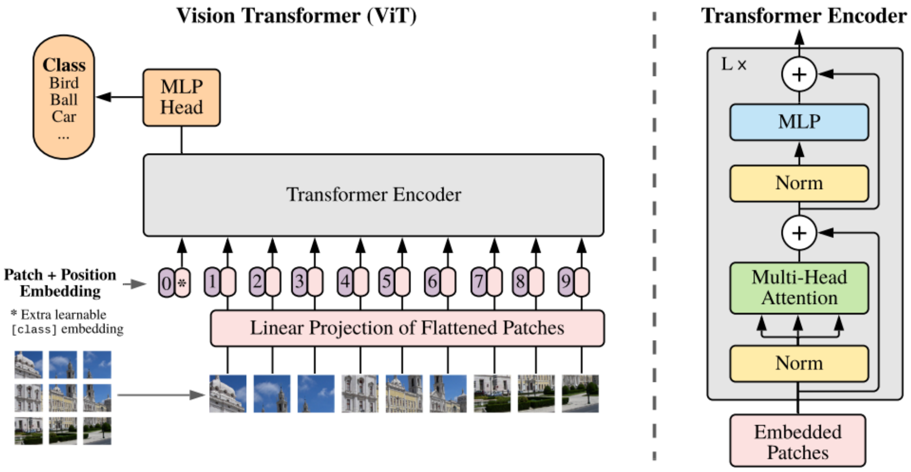


Figure 2.13: General description of Vision Transformer model. Image taken from [15].

The fundamental concept is to process images in a manner analogous to the manner in which text is processed. To achieve this, the input image is divided into patches of a fixed size, which may vary according to the implementation. These patches are then flattened and projected into an embedding space using a linear transformation. The resulting embeddings are then passed to the "Transformer Encoder", where a learnable embedding known as a "class token" (CLS) is added, along with information about the relative position of each patch (positional embedding). All this information is used as input to the transformer, whose number of layers depends directly on the implementation.

Finally, the output of the encoder transformer generates the feature maps, which then pass through a two-layer MLP with a nonlinear GELU (Gaussian Error Linear Unit) activation function. This process culminates in the respective image classification. It is noteworthy that, although the ViT produces a classification (directly linked to the CLS token), the patches can be retrieved for use in conjunction with the classification.

2.1.8 Contrastive Learning

Contrastive learning is a technique within the field of deep learning that seeks to learn an embedding space in which positive (similar) pairs are placed in close proximity to each other, while negative (dissimilar) pairs are placed at a distance. There are different types of contrastive loss functions, which utilize either a single positive and negative example or multiple examples. Some of these functions, which were addressed in later chapters, are:

- Triplet loss: was proposed by Schroff et al. [51] to address the challenge of recognizing the same person under varying angles and illumination conditions. The triplet loss function comprises three images: an anchor image, a positive image, and a negative image. The objective is to reduce the distance between the anchor and the positive image, while increasing the distance between the anchor and the negative image. The equation 3.1 provides a precise definition of the triplet loss function, employing a cosine distance metric.

- NT-Xent: also known as "Normalized Temperature-Scaled Cross Entropy", is a loss function introduced by Chen et al. [7] that aims to learn meaningful representations by maximizing similarity between positive pairs (augmented views of the same instance) and minimizing the similarity between negative pairs (augmented views of the same instance). This loss generates pairs of augmented images for each batch image N, resulting in a total of $2N$ pairs. Subsequently, the loss function seeks to maximize the similarity between the generated pairs of each image and minimize the remaining $2(N-1)$. In other words, the loss function for a positive pair (i, j) is defined as:

$$l_{i,j} = -log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{2.6}$$

  where $\text{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\|_2 \cdot \|z_j\|_2}$ y τ denotes the temperature parameter.

- InfoNCE: this loss function, introduced by He et al. [26], takes away the basic idea of NCE (Noise Contrastive Estimation) [40], whose measure of similarity comes from the product point. This is defined as:

$$L_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)} \tag{2.7}$$

  where q is a set of examples, tau is the temperature hyperparameter, and the sum of the denominators is a one positive and K negatives.

2.1.9 Self-Supervised Learning

Self-supervised learning (SSL) has emerged as a powerful alternative to supervised learning. Unlike supervised learning, which is often limited by the availability of labeled data, self-supervised approaches have the ability to learn from unlabeled data. This approach has been successfully used in areas of natural language processing (NLP) and computer vision. In NLP, models such as BERT [12] have been employed, where text tokens have been replaced with learning masks, thereby teaching the model to retrieve the original text. Similarly, computer vision has techniques inspired by BERT. One such technique, is currently used as pre-training of the ViT of SAM, is MAE (Masked Autoencoders). MAE has the task of directly reconstructing those masked patches.

The following are three models that will be used as part of the research to compare proposals in the context of Self-Supervised SBIR:

- SimCLR [7]: this framework employs a loss function defined in 2.6 to learn representations by maximizing the concordance between different magnified views of the same data example. This transformation process applies three random transformations to the data: random cropping followed by resizing to the original size, random color distortion, and random Gaussian blur. Figure 2.14 provides a general overview of the framework. Here, $\widetilde{x}_i$ and $\widetilde{x}_j$ are considered positive pairs, $f(\cdot)$ represents a base encoder network, and $g(\cdot)$ denotes the head projection.

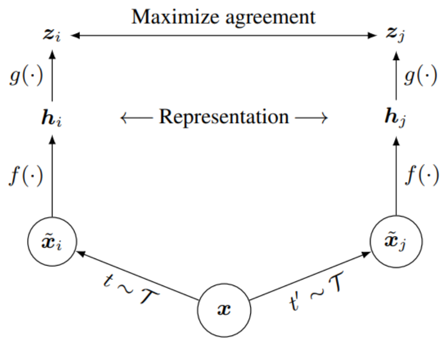


Figure 2.14: A framework used for contrastive learning in SimCLR. Image taken from [7].

- BYOL [25]: also known as "Bootstrap Your Own Latent", this framework belongs to the "self-destillation" family. It employs two neural networks, the so-called online and target networks, to learn. The online network comprises an encoder, a projector, and a predictor. The target neural network employs the same architectural framework as the online network, with the exception that it utilizes a distinct set of weights and a gradient stop. It is crucial to note that the target network updates the network weights through an exponentially moving average of itself and the online network.
- SimSiam [8]: also known as the "Simple Siamese Network", is a member of the "self-destillation" family of networks. In this type of network, two distinct views of an image are input to two encoders, which are then mapped to each other by a predictor. In detail, the architecture receives two augmented entries of the same image. These are then passed by a backbone (ResNet [27]) and an MLP project. The encoder f employs a shared weighting scheme between the two views. Ultimately, the MLP, designated as h, transforms a single output and adjusts it to align with the other view, with the objective of minimizing the negative cosine similarity. Figure 2.15 depicts the general scheme of SimSiam.

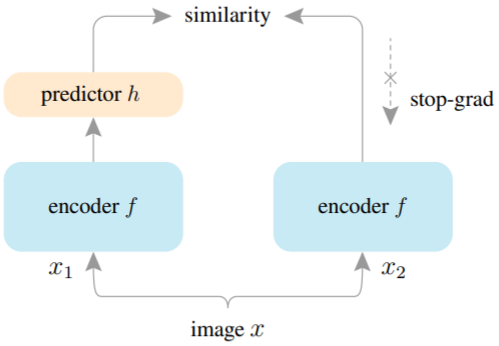
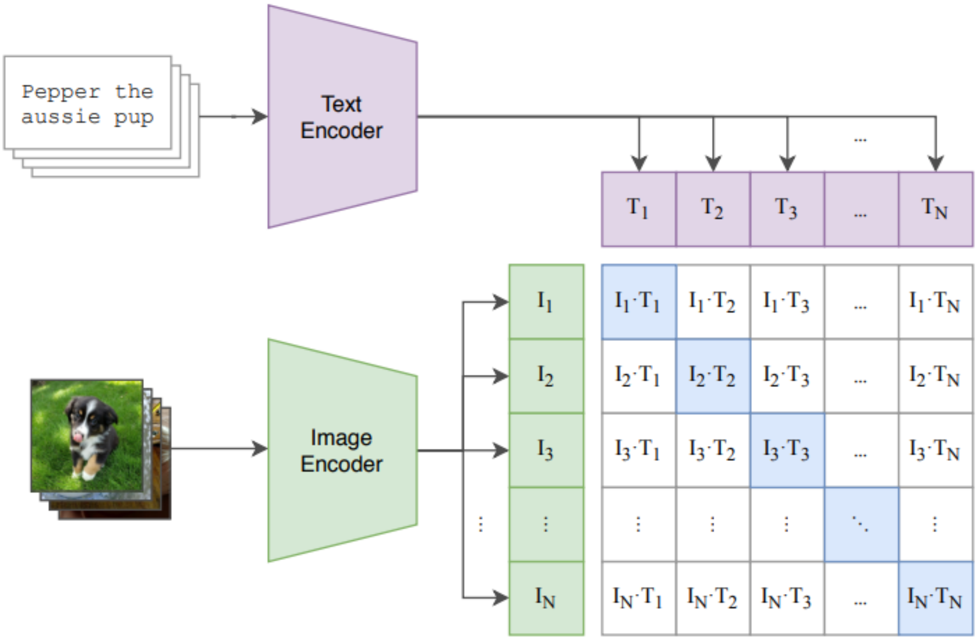


Figure 2.15: SimSiam architecture. Image taken from [8].

2.1.10 CLIP

Contrastive Language and Imagery Pretraining, also known as CLIP, is a model developed by OpenAI that introduced new concepts in the field of computer vision. The model was trained with data pairs image-text in order to learn how to associate concepts in images with natural language. In other words, the model learns to associate words (or complex phrases) with images containing those concepts. To illustrate, if we present the model with an image of a horse accompanied by the text “This is an image of a horse” the model should be able to associate the two concepts.

The principal advantage of CLIP in comparison to previous models is that it was not trained with a fixed set of categories. This contrasts with the models trained in ImageNet, which are limited to recognizing only 1000 categories. CLIP is trained to recognize any concept presented to it, thus enabling the model to recognize concepts that were not present during training. This is made possible by the models capacity to associate concepts with words, rather than with predefined categories.



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Figure 2.16: CLIP Architecture. Image taken from [43].

As illustrated in figure 2.16, both the text and the image have encoders. The text encoder is a transformer with 63 million parameters, 12 layers with 512 embedding dimensions, and 8 attentional heads. Specifically, an “encoding byte pair encoding” was employed to represent the text, with a vocabulary size of $49,152$. In contrast, two architectures were considered for the image encoder: the first based on a ResNet-50 and the second on a ViT. Both encoders generate an embedding of 512 dimensions.

Once the text and image embeddings have been generated, we proceed to contrastive training. In detail, contrastive training is intended to bring the vector spaces (embeddings of the text and image) as close as possible to the positive pairs. To achieve this, we obtain an array of similarity between the embeddings and the image. This similarity matrix is obtained by applying the product point between the embeddings of the image and text. This process allows us to identify which values are positive (the diagonal) and which are negative. The following pseudo-code in Python illustrates this process:

```
# image_encoder - ResNet or Vision Transformer
# text_encoder - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l] - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t - learned temperature parameter
# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]
# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)
# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)
# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

In summary, CLIP incorporates a contrastive approach supported by research, as seen in [43], which demonstrates its ability to learn representations that are more effective than equivalent predictive approaches. This highlights the effectiveness of this model in obtaining meaningful representations in multimodal spaces.

2.1.11 SAM

The Segment Anything Model (SAM) is a neural network that has been trained with more than 11 million images and more than a billion masks. These quantities significantly exceed current data sets, which allows the model to generalize well when segmenting, even in objects that it has never seen before.

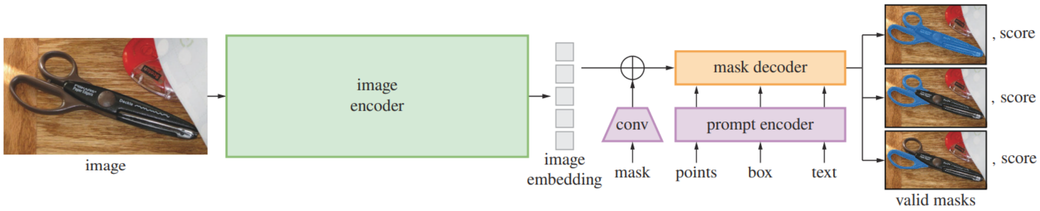


Figure 2.17: Segment Anything Model architecture. Image taken from [31].

The network comprises a Vision Transformer as encoder for images, a “prompt encoder“ for segmentation based on different prompts, and a mask decoder for the efficient generation of corresponding masks. The architectural design is depicted in Figure 2.17.

The initial step of SAM is of great importance for the subsequent parts of its architecture,

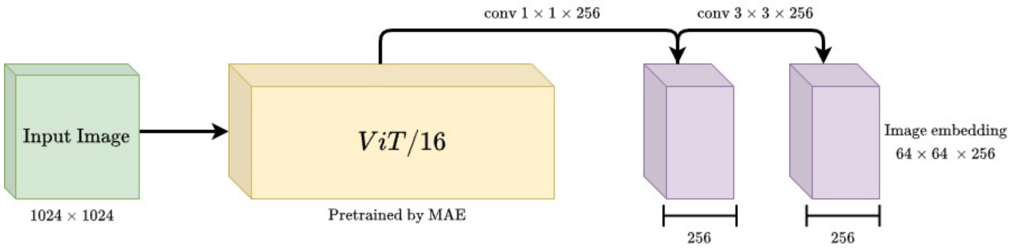


Figure 2.18: Image encoder architecture.

as demonstrated in 2.1.6, where it can be seen that this encoder extracts the essential features from the input image. The vision transformer (ViT-H/16) was trained using self-supervised learning with MAE [34], which uses an input resolution of 1024 × 1024. The output of the image encoder must pass through a series of convolutions to modify its dimensions and generate a feature map of dimensions $\mathbb{R}^{64\times64\times256}$, which will serve as a standard for the different encoders. This process is illustrated in figure 2.18.

One of the principal characteristics of SAM is its prompt encoding. It is capable of receiving as input a point within the image to be segmented, a bounding box, and even text (although this functionality has not yet been implemented by the authors). This research will focus exclusively on point encoding, as one of the proposed solutions makes use of this functionality.

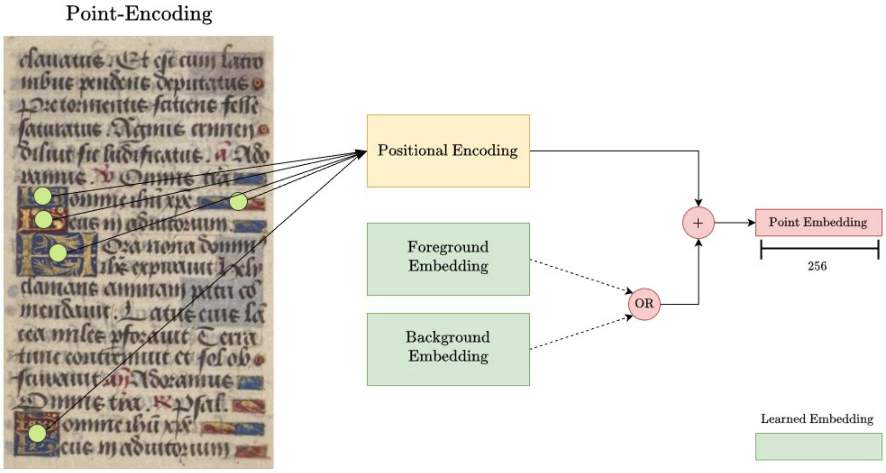


Figure 2.19: Schematic of the prompt encoding, in particular the point encoding for the image segmentation.

The point encoding refers to the capacity of the model to receive points that delineate a specific region of the image to be segmented. These points are encoded to incorporate their relative position within the image (positional encoding), along with information indicating whether or not they contain an object (foreground or background), as illustrated in figure 2.19.

Subsequently, both the image embedding and the point embeddings are subjected to a series of layers of attention and MLPs that facilitate the interrelation of the two, thereby obtaining novel representations for the image and the points, as illustrated in figure 2.20. Finally, these new embeddings are received by the decoder mask, which generates two outputs. The first of these relates both embeddings through a point product to obtain three different masks. The second approach employs point embeddings to generate the respective score of (IoU) masks generated through an MLP.

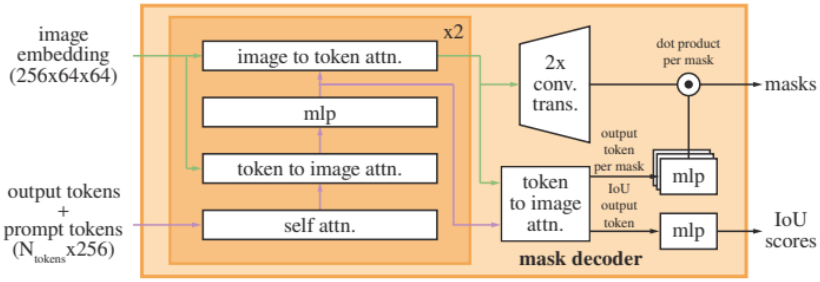


Figure 2.20: Mask decoder architecture. Image taken from [31].

2.1.12 DINOv2

DINOv2, developed by Meta, builds upon the concept of learning features at both the image and patch level, in a manner analogous to iBOT [62], and modifies certain design elements to accommodate large-scale data sets. The authors highlight that the majority of the advancements can be observed in terms of processing speed (twofold increase) and efficiency (threefold reduction in memory usage). The DINOv2 process draws inspiration from natural language processing (NLP) techniques, employing data similarity in place of external metadata.

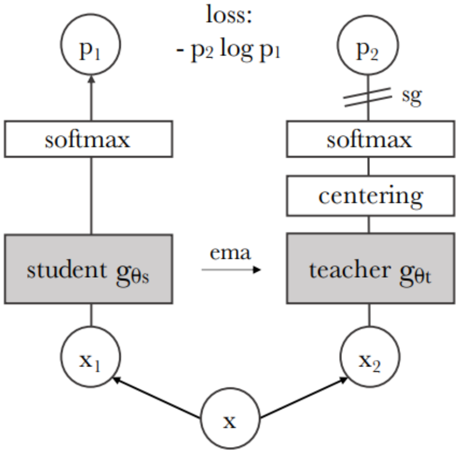


Figure 2.21: Architecture of self-distillation with no labels. Image taken from [5].

In order to understand DINOv2, it is necessary to first address its predecessor, DINO [5], since the second version maintains basic elements of the first. As the name suggests, “self-DIstillation with NO labels” refers to knowledge distillation, whose design aims for a small network to mimic the output of a large network (teacher-student). As illustrated in figure 2.21, both the student network and the teacher network exhibit identical architectural characteristics, yet differ in their respective parameter values (θ_s and θ_t, respectively).

The model receives as input two views of the same image, generated from random cuts, with the following notation: x_1 and x_2. The cutouts can be either global (representing a factor between 0.32 and 1.0 of the original image area) or local (representing a factor between 0.05 and 0.32 times the original area). The aforementioned cuts are processed through the backbone in both branches, where each output is normalized with a softmax function with temperature over its dimensions, denoted by P_s and P_t. Finally, the degree of similarity between this approach and a cross-entropy loss is evaluated. In this approach, the teacher network transmits gradients solely through the student, and the teacher parameters are updated with an exponential moving average (EMA) of the student parameters, as indicated in [5]. The equation 2.9 provides a detailed account of the loss incurred by the model.

$$\min H(P_t(x), P_s(x)), \tag{2.8}$$

$$\text{with } H(a, b) = -a \log b \tag{2.9}$$

As with DINO, iBOT receives two magnified views of the same image, designated as u and v. Prior to being processed by the student network, these views are masked, whereas for the teacher branch, they are not. Subsequently, both the student branch and the teacher calculate their characteristics through a vision transformer (ViT). The objective of iBOT is to minimize the discrepancy between the patches of both branches, given:

$$L_{iBOT} = -\sum_i p_{ti} \log p_{si} \tag{2.10}$$

where i are the patch indices for the masked tokens.

Finally, DINOv2 is proposed as an improvement of DINO, where the latter learns using a self-supervised discriminative method that can be seen as a combination of the DINO and iBOT losses, using a teacher-student architecture. It is crucial to acknowledge that there are numerous pre-trained DINOv2 models. In this thesis, the distilled ViT-B/14 model without registers is employed as a specific case.

2.2 Literature Review

2.2.1 Sketch-based Image Retrieval

Sketch-based image retrieval is the most popular and studied task in sketch-based understanding. Before the deep-learning era, researchers were focused on designing appropriate feature extractors to represent sketches and images (a.k.a. feature engineering) [10, 46, 29]. In addition, the first approaches had to deal with the scarcity of data [47, 17, 29].

As in other vision-related tasks, after the bloom of deep learning, the entire research community moved to work with convolutional neural networks and, more recently, attention models. Regarding the bimodal nature of SBIR, siamese networks have been the core architecture of the diverse proposals from the early to modern models on SBIR [50, 4, 48, 41].

In this vein, Bui et al. [4] presented a hybrid multi-stage training methodology for SBIR, exploiting contrastive learning and triplet loss. Their results showed good performance, achieving a mAP of 53.26% in the Flickr15K dataset and 65.99% in the Saavedra's dataset. More recently, Chaudhuri et al. [6] presented a data-free sketch-based image retrieval requiring only an unimodal classifier, which is used to train generative models to produce photos and sketches that are then used to train a metric learning model. They showed competitive results with respect to data-dependent approaches. Sain et al. [48] leveraged CLIP [43] for zero-shot sketch-based image retrieval. Their core contribution is a prompt-based learning mechanism added to the patch embeddings. The authors evaluated their proposal in terms of generalization in three different datasets, showing outstanding results.

Even though we have seen tremendous advances in SBIR, a critical problem still exists. Most of the approaches work under a supervised regimen and require having sketch-image pairs during training, which could limit their applicability. Zero-shot learning is a way to deal with the lack of labeling. Still, this strategy should be trained under a supervised regimen and does not consider information from the target application (e.g., eCommerce), which does not guarantee the best performance in a specific domain. For instance, in a recent work, Torres et al. [55] showed the poor generalization of SOTA models in eCommerce. A model achieving high performance in Flickr15K (56%) dataset shows poor performance on an eCommerce dataset 21%. This situation could be why we do not see a massive use of the sketch-query modality for image retrieval in the industry.

2.2.2 Self-Supervised Learning

Self-supervised learning (SSL) is one of the most promising methods to learn data representation with a high level of generalization that can be applied across diverse downstream tasks [2] and with high potential to be used in real scenarios.

In recent years, we have seen diverse self-supervision approaches, particularly for representation learning. A semantic representation (a.k.a. embedding) is the key to having an effective model. The current SSL approaches assume pseudo labels created by the intrinsic structure of the data. Thus, given a dataset X, we must define a process P that produces $[x_i, y_i]$, where x_i is the input and y_i is the corresponding pseudo label produced by P.

Ericsson et al. [21] divide the self–supervision methods according to the strategy to infer the corresponding pseudo labels. The intuition is that if the pseudo labels Y are created from some intrinsic structures in the data, a model learning to predict those labels must recognize and exploit this structure to solve Y successfully. Among the self-supervision typology defined by Ericsson, the instance discrimination regimen can be applied to the bimodal context of SBIR as they are based on siamese architectures.

In the **instance discrimination** regimen, each instance is treated as a unique element, and the model is trained to discriminate between different instances. This is the most successful approach, where contrastive and regularization-based are the most representative learning strategies.

- **Contrastive learning**: it is based on the siamese networks using triplet loss. Here, two transformations are applied to the input image to produce the anchor input and the positive one. The negative sample is obtained by transforming any other image. The objective of a model is to produce representations (a.k.a. embeddings or feature vectors) in such a way that the anchor and positive embeddings fall near each other (small distance), and the negative and anchor embeddings fall apart (large distance). MoCo [26], SimCLR [7], and CLIP [43] are good examples of this kind of strategy.
- **Non-Constrative learning**: a critical drawback of contrastive methods is the inherent difficulty of forming negative pairs. Alternatively, some methods use a **regularization strategy** to take only positive pairs. For instance, BYOL [25] is a non-contrastive model that trains a teacher-student architecture by adding a stop-gradient mechanism in the teacher branch. Thus, only the student parameters are modified during training concerning the produced loss. In addition, the student's knowledge is smoothly transferred to the teacher branch through an exponential moving average approach (EMA) as in MoCo. More recently, **DINO** (self-DIstillation with NO labels) [5] emerges as the best self-supervised strategy for the image domain, especially when it is combined with ViT architectures. DINO is based on a teacher-student architecture, similar to BYOL, but with a different similarity-matching loss. The representation is a probability distribution estimated by a *softmax* function over a centered representation. Caron et al. [5, 42] showed that centering and sharpening by *softmax* allow the model to avoid collapse. Finally, the loss encourages the student network to return the same distribution as the teacher does for the same input.

2.2.3 Sketch-base Detection

Sketch-based Detection is a relatively new and under-explored field of research within computer vision. Unlike object detection based on natural images, which has seen significant progress in recent years thanks to advances in deep learning [45, 37, 22].

One of the earliest works to address the topic of Sketch-based Detection was proposed by Tripathi et al. [56] (2020). In this work, the authors propose a model based on cross-modal attention that guides an Region Proposal Network (RPN) to the object being searched for based on the sketch. The authors idea was to add the query information to the image feature representation before generating the region proposals.

In a later work, the same author presented improvements on his original work. In this work, Tripathi et al. (2023) proposed a novel "sketch-guided vision transformer encoder" [58]. The primary objective of this architectural design is to condition the learning of the model based on the input sketch. The authors posit that this approach serves to mitigate the discrepancy between natural images and sketches, which represents a significant challenge in this field of research. Moreover, they introduced a decoder that receives both object features and sketch features, allowing for the refinement of the relevant object features (ground truth) to bring them closer to the query sketch.

A particularly innovative approach is that proposed by Chowdhurt et al. [6], who utilize SBIR as a component of a solution in sketch-based object detection. Their work is notable for not employing bounding boxes in model training; instead, they have opted for an SBIR approach to replace the labels of the dataset. The proposed architecture is based on an adaptation of CLIP, in which a vector of learnable parameters (known as prompt learning, as described in Zhou et al. [63]) is injected into the ViT in order to reduce the domain gap between natural images and sketches. The objective is to implement two distinct prompt learning techniques, one for natural images and another for sketches. This approach enables the model to be conditioned to reduce the domain gap between the two types of input.

This proposal is comprised of several key elements. Firstly, a feature extractor transforms the input image into a vector space. Subsequently, a pretrained RPN generates $1,000$ regions of interest. Subsequently, intermediate characteristics are calculated using the RoI pool and a fully-connected layer. Concurrently, each proposed region is processed through the CLIP model in conjunction with its respective prompt learning. Conversely, the sketches are fed directly into the CLIP model, along with their respective prompt learning. Finally, the sketch embedding is compared with the $1,000$ generated embeddings to identify the one with the greatest similarity.

2.2.4 Related work

In the field of pattern spotting and image retrieval in historical documents, several approaches have been proposed in recent years, each contributing innovations to the field. The approach proposed by En et al. [19] focuses on the analysis of informative areas, specifically figures and patterns, through the use of sliding windows. This method begins with the pre-processing of the documents, which involves the removal of the background (non-informative areas) based on the premise that the areas of greatest interest would be those that appear less frequently in the set of documents. This is followed by the extraction of features from the sub-windows, which is carried out using one of the BoVW (Bag of visual words), VLAD (Vector of locally aggregated descriptor) or FV (Fisher vector) representation. Finally, after applying principal component analysis (PCA) to the features extracted from the sliding windows, the similarity between the query vector and the vectors of the entire data set is calculated using the dot product. This allows for the identification of both the documents and the specific sections that are most relevant to the given query. Although the proposal demonstrates satisfactory performance in information retrieval, it exhibits limitations in identifying sought-after patterns.

In a more sophisticated approach, Úbeda et al. [65] proposed a CNN-based solution to address this problem. The method commences with a preprocessing phase analogous to that proposed by En et al. [19], wherein the background and text are eliminated from documents through the application of a classifier specifically trained for this task. Subsequently, the authors utilize the RetinaNet model [35] previously trained on the COCO dataset [36], to extract features from both documents and queries. The final stage of the process involves a similarity search based on cosine similarity, which allows the tasks of image retrieval and pattern spotting to be solved. This approach is notable for its utilization of the pretraining of a convolutional neural network (CNN) and the multi-scale representations of the feature pyramid network (FPN), which is a type of neural network. The proposal demonstrates an improvement in performance in the Patter Spotting (PS) task when compared to the results presented in [19]. However, in the Image Retrieval (IR) task, the performance is found to be inferior.

In a similar line of research, Wiggers et al. [61] presented a solution based on siamese neural networks, utilizing the successes obtained with these architectures in fields such as facial recognition and the identification of signatures or symbols as a reference. The method is based on the use of a CNN AlexNet architecture, which has been trained as a siamese network. A distinctive feature of this approach is the proposal of regions of interest within the document using the Selective Search algorithm, which is capable of identifying more than 36 million potential regions in the $1,500$ pages of DocExplore. The presented proposal achieves comparable results to [65], where the notable aspect is the extensive number of proposed regions, a significant proportion of which may be considered noise.

In a recent study, Dias et al. [9] proposed a significant improvement over the work of Wiggers et al. [61]. Their approach maintains the use of the Selective Search (SS) algorithm but introduces an additional post-processing step to filter out regions that do not contain relevant information. To extract features from these regions, the authors employ a siamese neural network (SNN) with two distinct architectures: one based on VGG19 and the other on ResNet50V2. The innovation of this approach lies in the incorporation of a deep hashing layer, whose objective is to convert the original high-dimensional features into low-dimensional hash codes. This process not only reduces the computational complexity, but also improves the efficiency of the similarity search. Finally, for the similarity search phase, the authors propose the use of XOR to leverage the binary nature of the generated codes. This proposal was able to reduce the number of proposed regions by 46 times compared to the work of Wiggers et al. [61], thereby demonstrating a significant improvement in the efficiency of the process. Moreover, in the context of image retrieval, this method demonstrates superior performance compared to previous proposals, approaching the performance of the approach presented in En et al. [19]. In contrast, in the context of pattern spotting, the method in question outperforms all previous proposals.

It is important to note that, while the works presented above present innovative proposals, none of them focuses specifically on the use of sketches. Nevertheless, these studies have made clear two challenges when working with DocExplore. The first challenge is to propose regions with relevant information. Most approaches focus on data preprocessing to eliminate areas without information. The second challenge is to search by similarity. This is very relevant, since the gap between sketches and natural images must be reduced. In response to these challenges, it is proposed that SAM be used for the first point. Regarding the second point, the following chapter is dedicated to a detailed examination of the problem of the gap between sketches and natural images, with a particular focus on the use of self-supervised approaches.

Chapter 3

Self-Supervised SBIR

Sketch-Based Image Retrieval (SBIR) is the most popular task in the sketch-based understanding domain. It is a bimodal problem that combines two different image modalities: sketches and regular images, where the last are generally photographs. SBIR models aim to learn meaningful representations from sketches and photos in a shared semantic space. Beyond image retrieval, we can leverage learned SBIR representations to solve other sketch-based understanding problems. For instance, we can use them for sketch-based object localization [6] or sketch-to-photo translation [14]. In both cases, having discriminative representations in a bimodal feature space is the key to achieving high performance.

However, a critical problem when we train an SBIR model is the need for paired data available to produce a bimodal sketch-photo feature space. That is, we must have access to real sketch-photo pairs for training, which can be unfeasible. In fact, there are still small paired datasets like Sketchy [49] used for research, which does not allow us to have a general SBIR model; the models trained on those datasets suffer from low generalization.

Torres et al. [55] showed a high-performance degradation of traditional SBIR models when evaluated on an eCommerce dataset. To address this problem, Morales et al. [41] proposed SBIR-BYOL, a self-supervised model extended from BYOL for the bimodal sketch-based image retrieval task. They showed better results with this self-supervised learning approach, which allowed a model to be adjusted to unpaired data by looking at the target image catalog (i.e., unpaired data). The self-supervised strategies are more appropriate for industry or for problems whose context is very specific (cultural heritage), as they do not need intensive labeling and produce better generalization.

Therefore, this work proposes self-supervised deep-learning models for sketch-based image retrieval S3BIR. The approaches can produce a bimodal sketch-photo feature space from unpaired data, requiring no explicit sketch-photo pairs. Our proposals are inspired by the recent work of Sain et al. [48] proposing a prompt-base CLIP for SBIR under a supervised regimen. In contrast, we show that a simpler architecture is enough for self-supervision in the bimodal context. The approaches rest on semantic visual encoders that can be easily exchanged. Here, we evaluate CLIP and DINOv2, which are adjusted to produce semantic representations from sketches and regular images. The approaches show outstanding performance in diverse public datasets under a self-supervised regimen. To demonstrate the proposed model's performance, we present a benchmark for self-supervised SBIR using diverse methodologies, including contrastive and non-contrastive models adapted to our context.

3.1 Related

3.1.1 Self-Supervised SBIR (S3BIR)

S3BIR (self-supervised sketch-based image retrieval) consists of training two encoders, one for sketches and the other for regular images (photos), aiming to achieve the maximum performance on image retrieval querying by sketches. Here, the term "self-supervised" consists of training without paired data. We do not have access to sketch-photo pairs. Generally, we only have access to a collection of regular images for querying that we regard as an unpaired dataset.

We still find a few works dealing with self-supervised sketch-based image retrieval. For instance, Morales et al. [41] proposed SBIR-BYOL as an extension of BYOL for sketch-image representations. The authors reported a mAP of 25.3% in the eCommerce dataset. However, self-supervised SBIR must be studied in depth.

Self-supervision on SBIR is addressed by presenting two models under the same self-supervision scheme: S3BIR-CLIP and S3BIR-DINOv2. Both proposals leverage features learned by visual foundational models like CLIP [43] or DINO[5, 42]. The prompt-based mechanisms proposed by Sain et al. [48] are also incorporated. The proposals work under a complete self-supervised regimen, creating pseudo-sketches by a semantic contour detection model [53].

3.2 S3BIR-(CLIP/DINOv2)

CLIP (Contrastive Language-Image Pre-training)[43] is a contrastive bimodal model trained on approximately 400 million of image-text pairs. This approach enables CLIP to establish a common embedding space for images paired with texts, which allows generalization in subsequent zero-shot tasks.

Sain et al. [48] adapted CLIP for zero-shot SBIR. Their proposal uses a pretrained CLIP, keeping all the layers frozen except for the normalization layers to avoid catastrophic forgetting. Additionally, the authors incorporated learnable vectors, known as *prompt learning*, which are injected as input to the ViT backbone of CLIP. The learned prompts are critical because they allow the model to adjust the resulting embedding depending on whether the input is a sketch or a photo. Even though Sain's proposal deals with a zero-shot task, it is trained under a supervised regimen requiring the availability of sketch-photo pairs.

Although the proposal is based on Sain's approach, it incorporates modifications to allow self-supervised training for sketch-based image retrieval using unpaired data. Figure 3.1

illustrates the general scheme of our approach named S3BIR-CLIP/DINO (self-supervised SBIR). Following is a description of each component:

- **A bimodal input**: as we deal with a bimodal context, we still require sketch-photo pairs to be formed. To allow our model to be trained under a self-supervision regimen, we use pseudo-sketches computed directly from each photo of the training dataset. To this end, we use PidiNet [53], a semantic contour detection model.
- **A visual encoder**: we use a visual encoder for pseudo-sketches and photos. However, to deal with these two modalities, we follow the proposal of Sain et al. [48] that incorporates learned prompts to differentiate a sketch from a photo, allowing us to keep the same encoder for both modalities.

  An input image (photo or sketch) is first resized to 224×224 and then split into 16×16-size patches, where a vector of 196 values represents each patch. In this manner, we can represent an input image by a matrix $E_{N \times 768}$, where N is the number of patches (in our case $N = 196$). E is then augmented with a learned positional encoding. In addition, the model defines two learned prompts, $V_s, V_p \in \mathbb{R}^{3 \times 768}$, that are added to E, depending on the input type. Furthermore, we add a class token to E to be used as the global representation. E is then passed through a ViT encoder that produces the boosted patch embeddings, where the boosted cls-token is used as the final representation after it is projected to $\mathbb{R}^{512}$.

  Our proposal allows us to exchange the visual encoder for any state-of-the-art one. In this proposal, we use CLIP and DINOv2, which are evaluated separately. In the case of S3BIR-CLIP, unlike Sain's proposal, we removed the text encoder. Consequently, the loss function is derived directly from the visual CLIP encoder.
- **A contrastive loss**: once the embeddings for both modalities have been acquired, the cosine distance is employed as the loss function for triplet loss. In particular, the triplet loss function is defined as follows:

  $$L_{tp} = \max\{0, D(S, P^{+}) - D(S, P^{-}) + \lambda\} \tag{3.1}$$

  where $D(x, y) = (1 - \cos(x, y))$ represents the cosine distance between two vectors, and S, P^{+}, and P^{-} are the embeddings of the pseudo-sketch, positive image, and negative image respectively.

3.3 Experimental Setting

Here, the focus on assessing the proposals in the context of sketch-based image retrieval. Following this, the involved datasets and evaluated models are described.

3.3.1 Datasets

For the evaluation on the sketch-based image retrieval task, the use the following datasets:

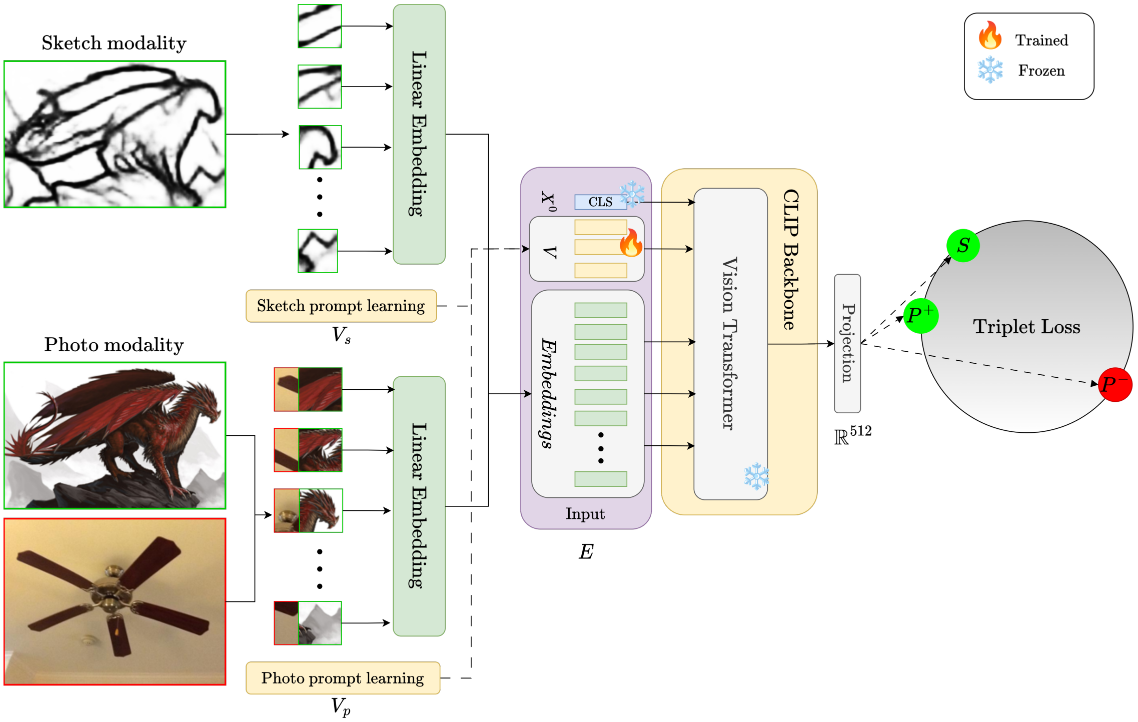


Figure 3.1: The S3BIR-CLIP training methodology. The images with green borders represent the pseudo-sketch/image of the positive pair. Conversely, the red border will serve as the negative image to be utilized in conjunction with the triplet loss.

- **Flickr15K** [29] (testing): a dataset containing 15,000 images, mostly of outdoor scenes, divided into 33 different classes. This dataset also contains 330 sketches, equally distributed over the 33 classes. This dataset was only used to evaluate the retrieval precision of the models.

- **Flickr25K** [16] (training): this dataset is an extension of the sketch dataset proposed by Eitz et al. [16], which consists of 20,000 different sketches distributed into 250 categories. Flick25K adds 25,000 photos distributed in the same 250 categories as Eitz's. It is important to note that our training protocol does not utilize the sketches provided; instead, we only use the photo collection. Flickr25K is used along with Flickr15K, where the former is used for training and the last for evaluation.

- **eCommerce** [55] (training - testing): this dataset was recently proposed by Torres et al. [55], aiming to have a closer approximation of the performance of SBIR models in real environments. This eCommerce dataset consists of two non-overlapping sets. The first one is a collection of 50,701 photos of diverse products representing what an eCommerce sells. This collection is used for training only. It is important to note that this set does not contain any query sketch. The second set consists of 5,665 photos and 665 real query sketches. This second set is used for evaluation, where the collection of photos is the target catalog for searching. Both sets contain images distributed among 141 product categories.

- **QuickDraw Extended** [13] (training - testing): this dataset was introduced by Dey et al. [13] to evaluate Zero-Shot SBIR systems. This dataset consists of 330,000 sketches and 204,000 photos spanning across 110 categories.

Before delving into the models used, the partitions of the data sets used should be detailed. Flickr15K and Flickr25K do not follow any particular partition since, as mentioned above, Flickr25K is used to fit the model, and Flickr15K is used to evaluate it. The eCommerce dataset has already been divided into a training catalog and a validation set. The validation set includes a search catalog and a set of real sketches for querying. Finally, for the QD-Extended dataset, the classes were divided into two groups: 80% in the first group and 20% in the second. The initial 80% of the data set is then divided again, with 80% allocated for training and 20% for validation. The second group is used to test the generalization capability of the model on unknown classes.

Table 3.1 quantitatively describes the used datasets. The number of photos or sketches used for training, validation, and testing is reported. Here, a validation set is composed of a target catalog with a similar distribution to the training target catalog. A testing set contains a target catalog with a different distribution from the training dataset. Generally, unseen classes are used to form the testing dataset.

Datasets	Training		Validation		Testing (unseen classes)	
	sketches	photos	sketches	photos	sketches	photos
$Flickr15K$	-	-	-	-	329	14,501
$Flickr25K$	-	25,000	-	-	-	-
$eCommerce$	-	50,701	665	5,665	-	-
$QD-Extended$	-	129,810	266,350	32,498	66,022	40,476

Table 3.1: Quantitative description of the used datasets for SBIR-based evaluation.

3.3.2 S3BIR Models

We compare our two proposals S3BIR-CLIP and S3BIR-DINOv2 with the following self-supervised models trained in the context of S3BIR.

- **S3BIR-BYOL**: we follow the original SBIR-BYOL [41] implementation and use the teacher network to process sketches and the student network to process photos. The exponential moving average (EMA) rate starts at 0.99 and is decayed up to 1 with a cosine schedule throughout the entire training procedure. Both the projection head and the prediction head are 2-layer MLPs with batch normalization and RELU activation only after the hidden layer; the hidden layers' size is 4,096, and the output layers' size is 256.
- **S3BIR-SimSiam**: this is an extension of the SimSiam model [8]. Here, we follow the same reasoning as in SBIR-BYOL and have the prediction head process the output of the branch that processes photos. The projection head is a 3-layer MLP with batch normalization in every layer and RELU activation in the hidden layers; every layer's output size is 2,048. The prediction head is a 2-layer bottleneck MLP with batch normalization and RELU activation in the hidden layer; the hidden layer's size is 512, and the output layer's size is 2,048.

- **S3BIR-SimCLR**: this follows the proposal of Chen et al. [7]. The projection head is a 2-layer MLP with batch normalization and RELU activation in the hidden layer, and both the hidden layer's and output layer's size are 2,048. The temperature used in the loss function is 0.5.

3.3.3 Settings

The proposals were implemented and run under PyTorch. Adam optimizer was utilized with its default parameters with the exception of the learning rate, which was adjusted for the various components of the model following the same methodology of Sain et al. [48]. Thus, we used a learning rate of 1×10^{-6} for the normalization layers and 1×10^{-4} for the prompt learning. The batch size in all experiments was 32. The models were trained for a maximum of 50 epochs. For the triplet loss function, we used $\lambda = 0.2$. The S3BIR-BYOL, S3BIR-SimSiam and S3BIR-SimCLR models utilize ResNet-50 [27] as the backbone, whereas the S3BIR-CLIP and S3BIR-DINOv2 models employ the ViT-B/16 architecture pre-trained on ImageNet [11]. A similar configuration was employed for SBIR-DINOv2.

3.4 Experimental Results

3.4.1 Quantitative Results in SBIR

Table 3.2 presents the results obtained by our S3BIR proposals using CLIP and DINOv2 for different contrastive loss functions for sketch-based image retrieval. We observe that the traditional triplet loss (distance-based) achieves the highest mean average precision (mAP) for both datasets. In the eCommerce dataset, the results between the three loss functions are highly comparable, with the triplet loss function demonstrating the most favorable outcomes. In this vein, using triplet loss, S3BIR-CLIP and S3BIR-DINOv2 achieve similar mAP values, 45.38% for the former and 44.30% for the latter. However, the difference is greater in Flickr15K, where S3BIR-CLIP achieves 54.03% and S3BIR-DINOv2 61.09%.

In addition, a more pronounced discrepancy between the loss functions is observed in the Flickr15K dataset. The traditional triplet loss function keeps the superiority condition. However, InfoNCE performed particularly well for Flickr15K, achieving the highest value of mAP@200.

Table 3.3 presents the performance of the proposed S3BIR models evaluated on the datasets presented in section 3.3.1. Here, all our proposals are trained using the traditional triplet loss because of the results shown in Table 3.2. The models are evaluated in terms of mAP@all and mAP@200. In addition, we will later present the models' performance regarding recall and precision to have a finest analysis.

Thus, Table 3.3 indicates the superiority of S3BIR-DINOv2 achieving a mAP of 61.09% in Flickr15K and 17.57% in QD-extended, the best results achieved under a self-supervision regimen. Furthermore, our results on Flickr15K are widely superior to previous SOTA models

Loss Functions	Encoder	Flickr15k		eCommerce	
		mAP@all	mAP@200	mAP@all	mAP@200
Triplet loss w/ cosine distance	CLIP	0.5403	0.6267	**0.4538**	0.4644
	DINOv2	**0.6109**	0.6551	0.4430	**0.4897**
NT-Xent	CLIP	0.4194	0.4808	0.3811	0.4509
	DINOv2	0.5262	0.6354	0.4381	0.4896
InfoNCE	CLIP	0.3348	0.4449	0.3723	0.4320
	DINOv2	0.4741	**0.6732**	0.4310	0.4818

Table 3.2: S3BIR-CLIP mAP with different loss functions.

like the proposal of Bui et al. [4] that achieves 53.26%. S3BIR-CLIP remains the second-best model, exhibiting slightly superior performance in the eCommerce dataset, achieving a 45.38% in mAP@all.

For further analysis, Figures 3.2 and 3.3 show the precision of the proposals for different values of recall for the eCommerce and Flickr15K datasets, respectively. In the case of eCommerce, S3BIR-SimCLR shows a similar behavior as S3BIR-DINOv2, being superior to S3BIR-CLIP for the first recall values. For Flickr15K, S3BIR-CLIP and S3BIR-DINOv2 present a similar high performance and superior to the others.

Models	Flickr15K		eCommerce		QD-Extended	
	mAP@all	mAP@200	mAP@all	mAP@200	mAP@all	mAP@200
S3BIR-BYOL	0.1176	0.2552	0.2606	0.3876	0.0589	0.1231
S3BIR-SimSiam	0.1086	0.1899	0.2134	0.2915	0.0563	0.1146
S3BIR-SimCLR	0.3424	0.4222	0.4180	0.4762	0.0749	0.1501
SBIR-CLIP [48]	-	-	-	-	*0.2020**	-
S3BIR-CLIP	0.5403	0.6267	**0.4538**	0.4644	0.1380	0.2637
S3BIR-DINOv2	**0.6109**	0.6551	0.4430	0.4897	**0.1757**	0.3102

* it is not a self-supervised model as it has access to sketch-photo pairs for training.

Table 3.3: Performance of the studied self-supervised SBIR models on diverse datasets.

3.4.2 SBIR Qualitative Results

This section presents examples of qualitative results of our proposals, S3BIR-CLIP and S3BIR-DINOv2. Figures 3.4 and 3.5 illustrate the outcomes on the eCommerce and Flickr15K datasets, respectively. Although both models produce a similar performance, there are cases where DINOv2 shows a better behavior. For instance, the first example of Figure 3.5 shows the superiority of S3BIR-DINOv2 in the retrieval task.

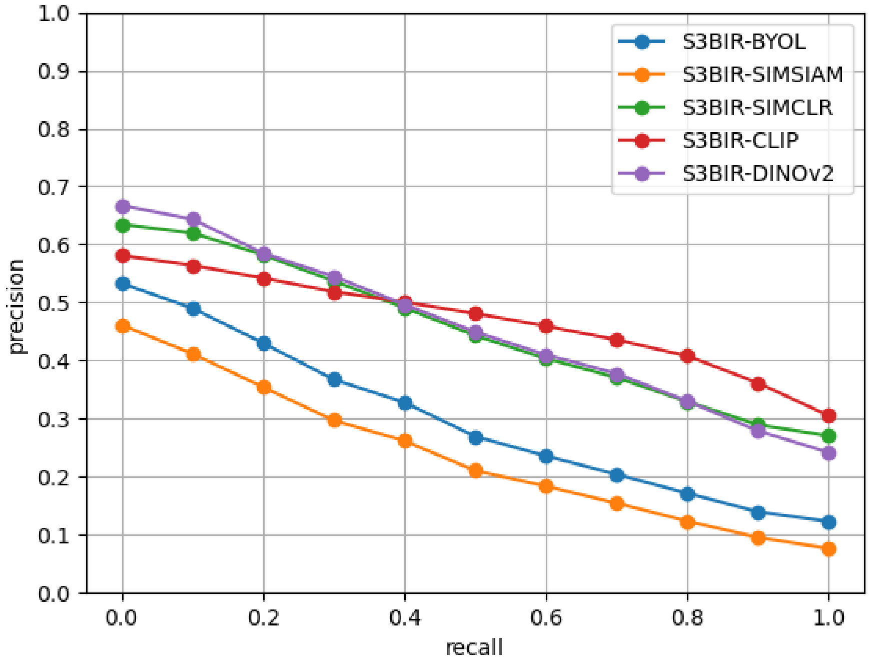


Figure 3.2: Recall-Precision curve on the eCommerce dataset. S3BIR-DINOv2 and S3BIR-SimCLR show similar behavior, superior to the performance of S3BIR-CLIP for the first recall values.

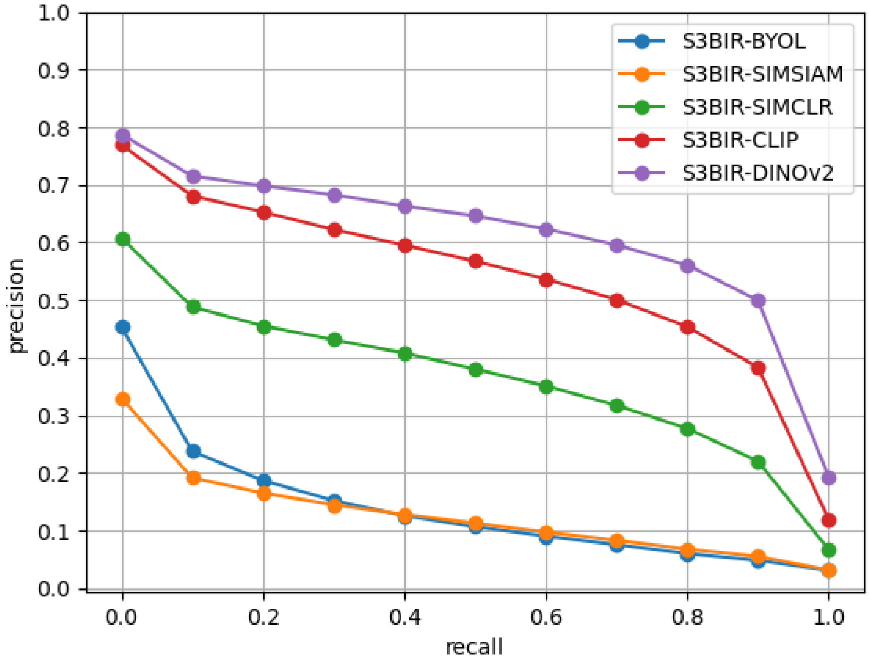


Figure 3.3: Recall-Precision curve of the Flickr15K dataset. Here, the superiority of S3BIR-DINOv2 is more visible.

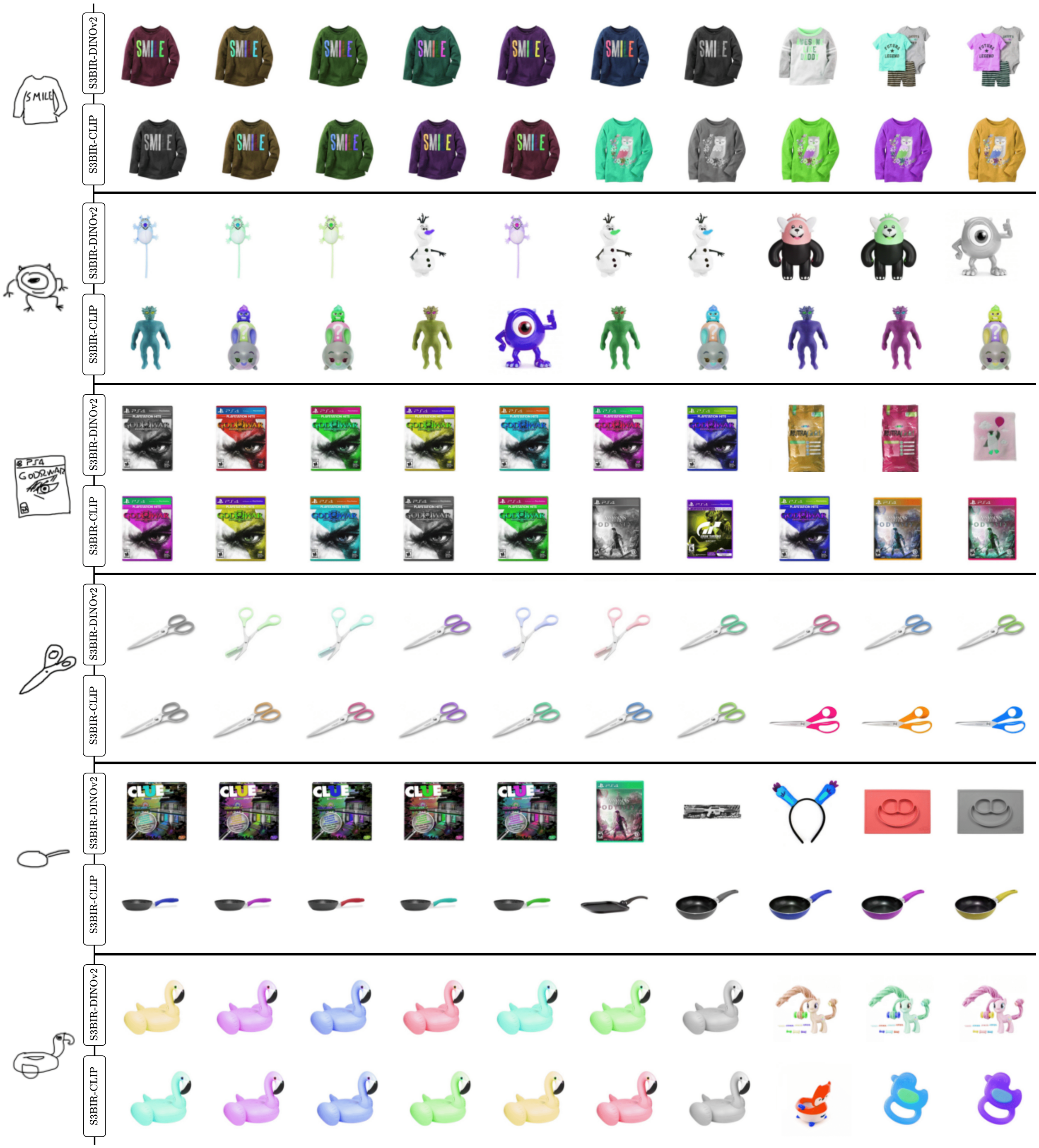


Figure 3.4: Examples of sketch-based image retrieval using our proposals (S3BIR-DINOv2 and S3BIR-CLIP) in the eCommerce dataset.

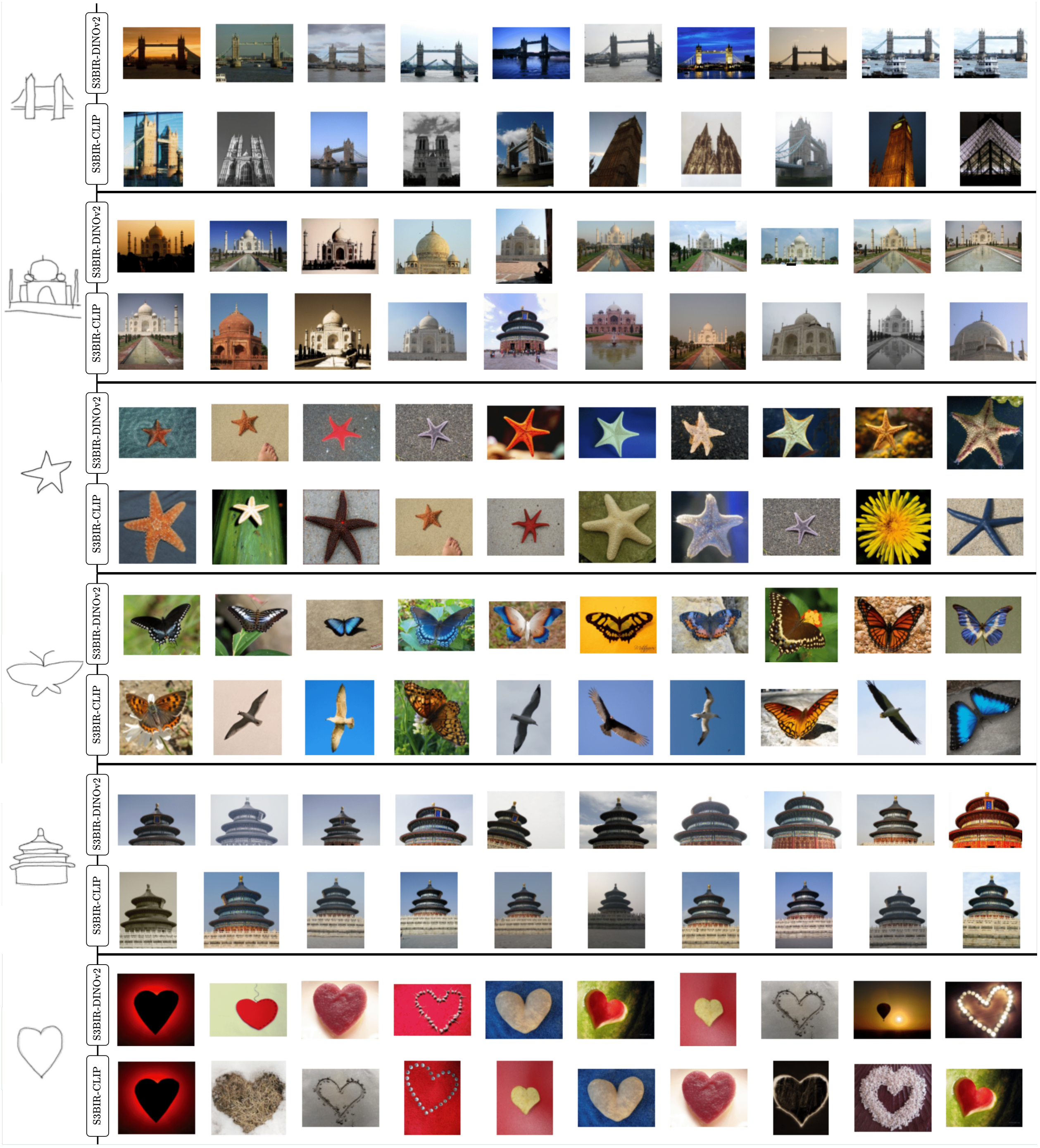


Figure 3.5: Examples of sketch-based image retrieval using our proposals (S3BIR-DINOv2 and S3BIR-CLIP) in the Flickr15K dataset.

Chapter 4

Sketch-based One-shot Detection

The domain of object detection encompasses a challenging task known as sketch-based one-shot detection. In this challenge, models must be able to detect and localize all instances of a specific object within the natural image (target), using only the reference sketch provided by the user (one-shot detection). Consequently, models are designed to learn meaningful representations from sketches and photos in a shared semantic space.

This task is closely related to the field of sketch-based image retrieval. As previously outlined in chapter 2.1.1, SBIR is designed to facilitate the retrieval of images from a database (catalog) based on a sketch. Similarly, sketch-based one-shot detection aims to identify each of these objects within the image based solely on the sketch. This thesis proposes the use of SAM to perform an initial segmentation of the objects present in the image, which is then transformed into a task of SBIR.

In order to effectively approach the detection of cultural heritage objects based on sketches, it is essential to have a clear understanding of the data to be worked with. Therefore, an exploratory analysis of the DocExplore dataset is presented below, which will allow us to understand the characteristics and distributions of the documents and queries in this dataset.

4.1 Exploratory Data Analysis

To evaluate the performance of the proposed model, the DocExplore [38] dataset is used. This dataset consists of manuscripts (referred to as pages) and symbols (referred to as queries) from the $10th$ to the $16th$ century, specifically from the Municipal Library of Rouen, France. In detail, this dataset consists of a total of 1500 pages and 1447 queries distributed over 35 different categories.

The number of occurrences of each of the queries presented in figure 4.1 varies between two and more than one hundred. These queries exhibit variations in color, size, and some are distorted during the scanning process for digitization. In addition, they usually have a reduced size of 200 pixels (20×10).

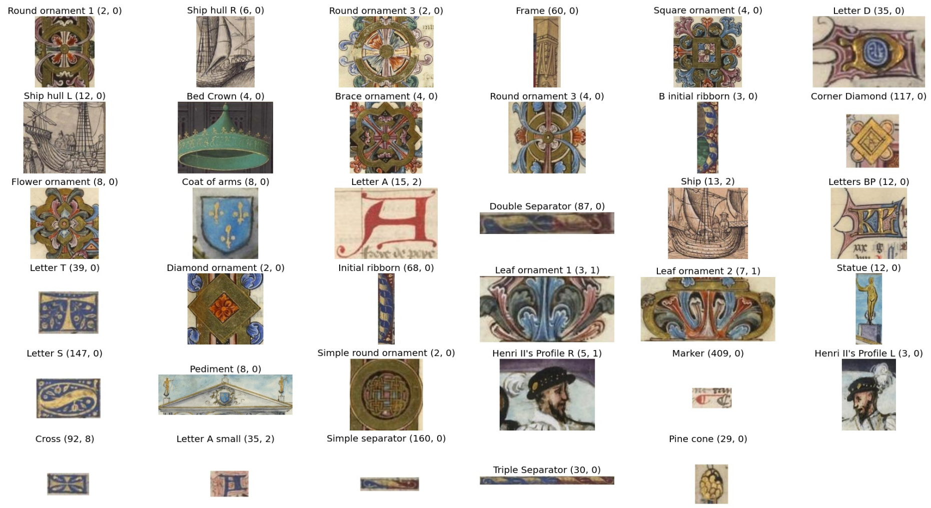


Figure 4.1: Number of occurrences of each category in the dataset.

With regard to the document pages, it can be observed that they share a similar structural composition, although some exhibit a higher degree of quality in the scanned image. Figure 4.2 illustrates that 75% of the pages have a size of less than or equal to 952 × 622. However, there are instances where the scan quality is notably superior, reaching values of 2758 × 3852. These pages contain drawings.

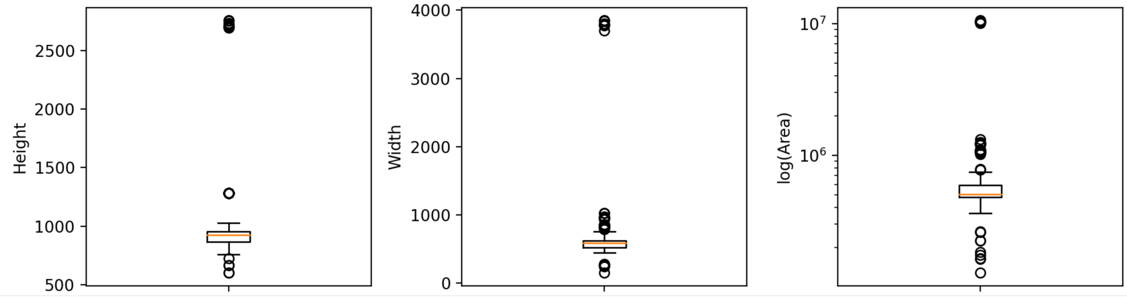


Figure 4.2: The first boxplot shows the heights of the pages. The second shows the widths of the pages. Finally, the last plot shows the total area of the pages in logarithmic scale.

Conversely, it is observed that not all of the 1, 500 pages contain information, and some are in a state of disrepair. For further details, please refer to figure 4.3. Furthermore, on average, the pages contain only 67% of the relevant information. A graphic example is shown in figure 4.4.

Although the sketches are employed instead of the queries provided by DocExplore, it is imperative to analyze these queries, as it is essential to comprehend the types of shapes and patterns that are being sought. Figure 4.1 depicts the frequency of occurrence for each query, which ranges from two to over four hundred. This reflects the inherent complexity of some queries in comparison to others. For instance, in the case of “marqueur”, there is a high degree of variability, including variations in size, shape, color, distortion upon scanning, and degradation. This variability is reflected in the 409 shapes representing the same class, which demonstrate a range of characteristics.

Figure 4.3: Pages of documents that are in poor condition or do not contain relevant information.

For a more detailed analysis, we utilize the proposal by Úbeda et al. [65], which suggests the use of two variables to categorize the objects in question: size ($w \times h$) and aspect ratio (h/w). This approach allows for the creation of two distinct categories: small/large and square/non-square.

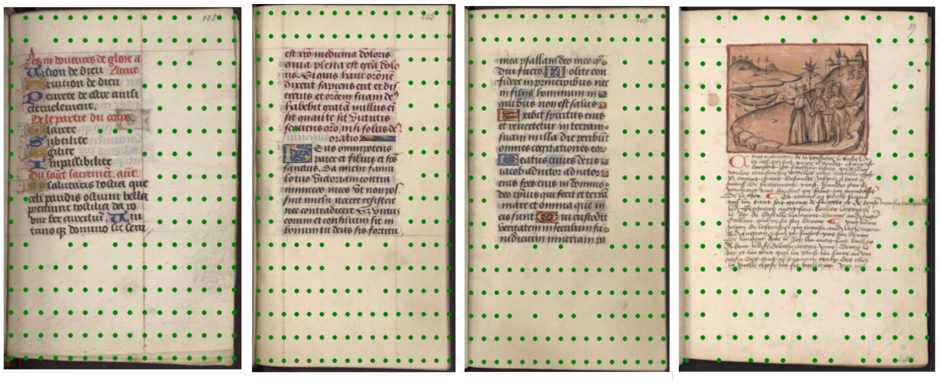


Figure 4.4: Green dots indicate sections of the document that do not contain information that is important to know.

Figure 4.5 illustrates the skewed distribution of query sizes, with the majority of queries falling below 100 pixels. This figure also shows that over 95% of the total queries fall within this range. Upon examination of the distribution of areas according to size, it becomes evident that the small queries exhibit a concentration at values close to 10^3 pixels. However, this category also contains a notable number of outliers. Conversely, the large queries exhibit a more concentrated distribution, with a smaller number of outliers.

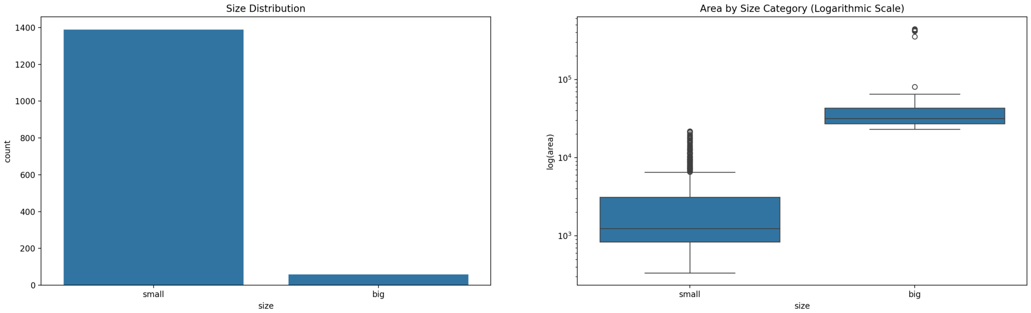


Figure 4.5: On the left is a bar chart analyzing the frequency of queries by size. On the right is a boxplot analyzing the distribution by size based on area (in logarithmic scale).

4.2 Data Collection and Preprocessing

This problem has been previously addressed as pattern detection problems without the use of sketches. For further details, see [65, 61, 20, 9]. Since this is the first time that a similar problem has been addressed using sketches, it was imperative to collect handmade sketches that closely resemble the queries delivered by DocExplore. To this end, a page developed in AIDA lab was employed, wherein it is possible to create sketches of the various figures being sought. The image figure 4.6 depicts a selection of the sketches that were generated.

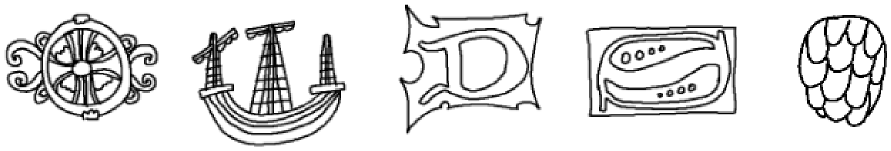


Figure 4.6: Handmade sketches of the different classes presented in DocExplore.

Conversely, a preprocessing evaluation was conducted on the documents. As previously stated in the EDA section 4.1, the average content of the documents comprises only 67% of relevant information. For this reason, an algorithm based on the proposal of Úbeda et al. [65] was developed to remove the background from the documents. Specifically, the following steps were followed: The initial step involved transforming each page into grayscale. Subsequently, a threshold and an erosion operation were applied to create the mask that will contain the relevant information. Finally, the area of the image with the highest density is identified and searched for, starting from the center of the image and proceeding towards the edges. It should be noted that all images were subjected to a preprocessing stage prior to being utilized. Figure 4.7 illustrates the algorithm in a step-by-step format.

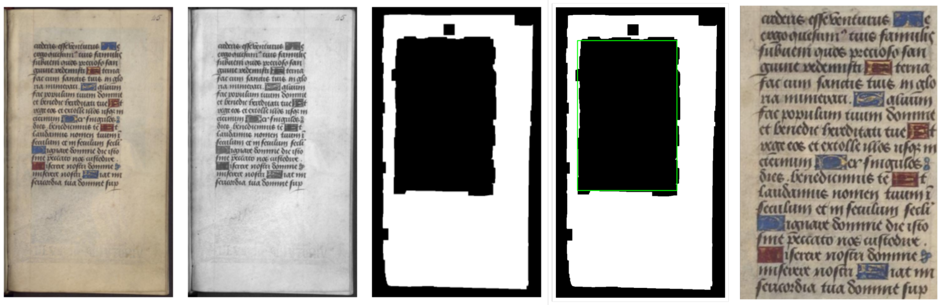


Figure 4.7: Preprocessing followed to remove the background of the documents.

4.3 Proposals

The proposals are structured around two principal phases: the offline phase and the online phase. The offline phase is primarily concerned with the extraction of features from a corpus of historical documents or catalogs. This phase is designated as “offline” due to the ability to pre-compute and store the embeddings of the documents for future utilization.

In the offline phase, historical documents or catalogs can be subjected to preprocessing in order to identify points of interest within the documents or images. This can be achieved either by cross-correlation or SAM. Finally, the identified points of interest are processed by the pretrained image encoder in conjunction with photo prompt learning (in a self-supervised manner) to obtain and save the resulting vector representations.

In contrast, the online phase is concerned with the acquisition of features from the input sketch, as this will be employed to search for and identify patterns within documents and images. As illustrated in figure 4.8, an image encoder is employed in conjunction with its prompt learning sketch to obtain the respective embedding.

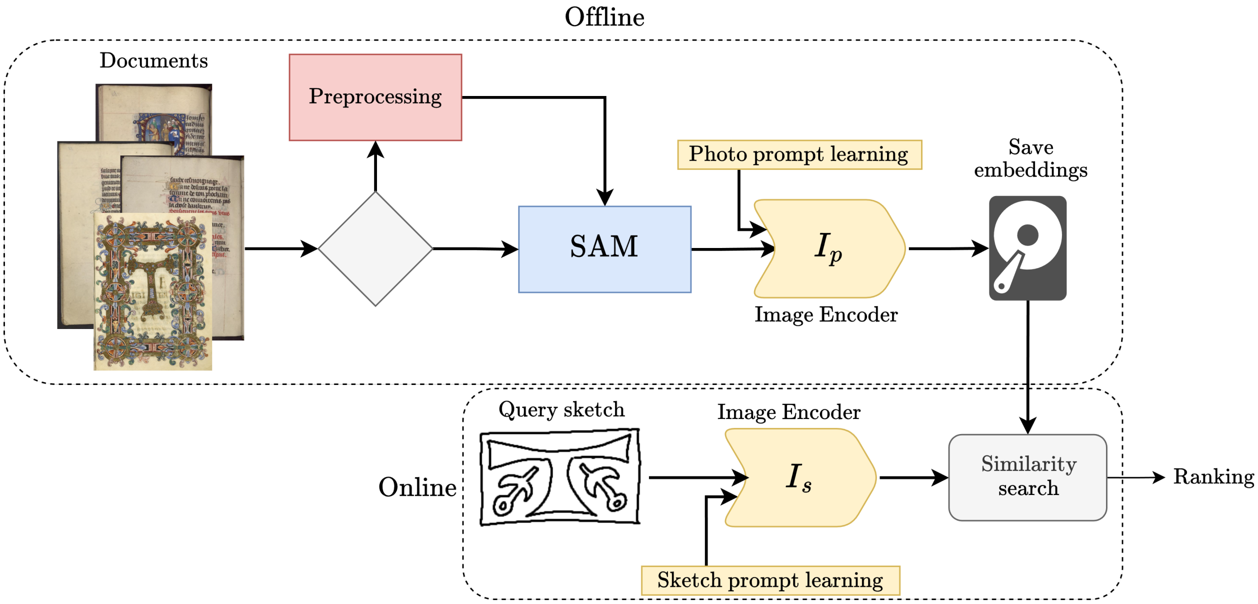


Figure 4.8: General diagram of the procedure followed by the proposals.

Finally, three distinct proposals have been put forth to address this problem. The initial proposal is established as a baseline, as it represents the first instance of employing sketches to identify patterns within DocExplore. Subsequently, the proposal is presented with SAM as a fundamental component of the solution. Finally, the use of prompting in SAM is proposed. This involves selecting specific points of the document using the similarity between the patches.

4.3.1 Base Proposal

This is the inaugural instance of employing sketches to assess performance in image retrieval and pattern spotting in the context of DocExplore. Consequently, a baseline is proposed to serve as a point of comparison for measuring the impact of the other proposals.

This approach is primarily based on cross-correlation. A document resized to 224 × 224 is utilized as input for the vision transformer, where it is divided into patches of dimensions 16 × 16 × 3. Each patch is represented by a vector of dimension 196. For an illustrative

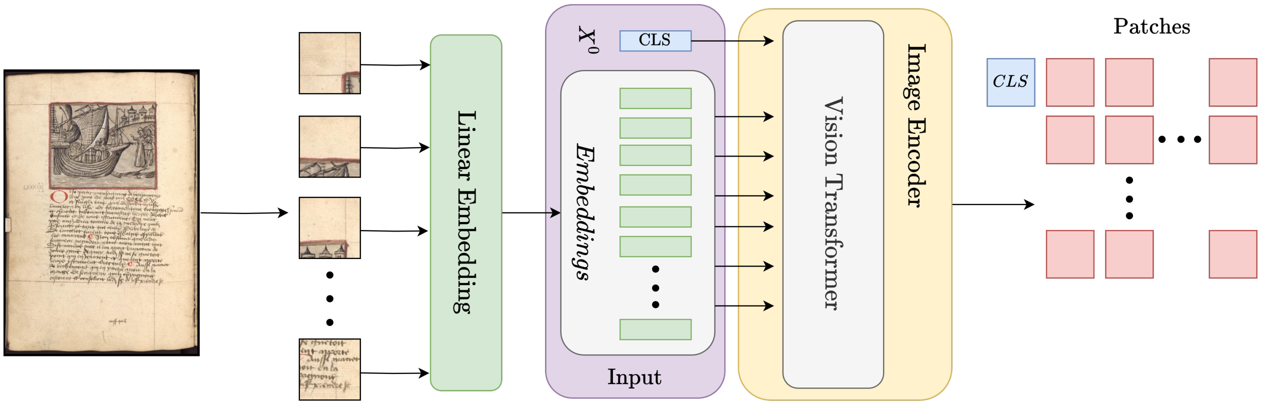


Figure 4.9: Process to obtain the image patches together with the classification token.

example, refer to figure 4.9. In a similar vein, we employ a sketch as input, with the distinction that here we do not retrieve the patches, but rather the classification token (CLS), which is a vector of dimension 768. Finally, a dot product is performed between the two vectors, resulting in the generation of a heatmap. The regions with the highest similarity between the sketch and the patches are highlighted. Figure 4.10 illustrates the heatmap generated by the cross-correlation between a sketch and a document.

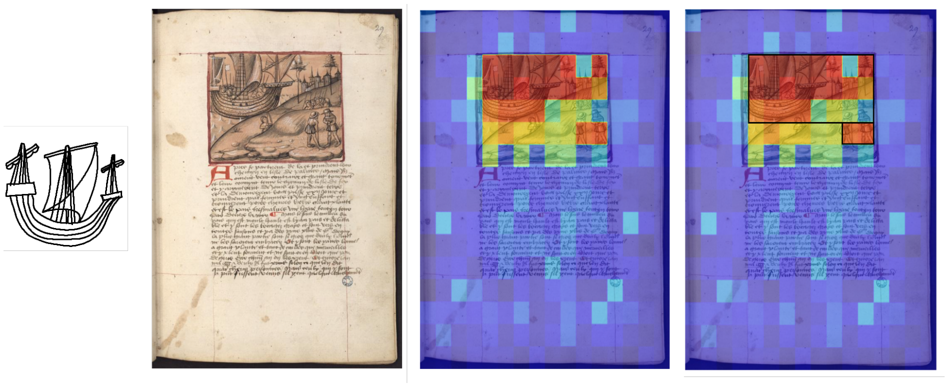


Figure 4.10: Exemplification of the correlation between a sketch and a target. From left to right is presented: 1) Hand-drawn sketch representing a bateau class query. 2) Image containing the searched query. 3) Heat map generated by the query on the document patches, where red color indicates high correlation, followed by yellow and finally blue representing low correlation. 4) Selected areas with higher correlation and their respective black bounding boxes.

This process is repeated with all the pages in DocExplore (1500) and the different sketches (1447). Finally, the pages can be ordered according to the maximum similarity achieved with the different sketches and the image retrieval task can be evaluated. A similar procedure is employed to obtain the respective bounding box. In this instance, a mask is generated that filters out those locations with the highest degree of similarity. This can be observed in the final document in figure 4.10.

4.3.2 SAM Proposal

This proposal places particular emphasis on the utilization of the SAM model as a fundamental piece of the solution. Figure 4.11 depicts the general scheme of the proposal for object localization based on sketches. It identifies the offline and online phase.

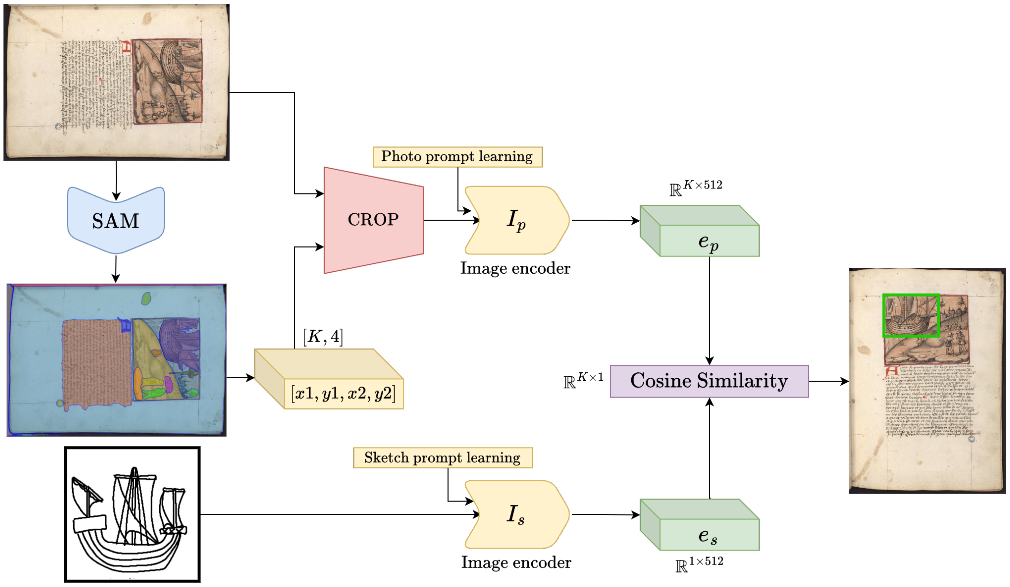


Figure 4.11: General scheme of the proposal using SAM.

- Offline phase: given document, with its original spatial dimensions $h \times w$, is utilized as input for SAM. By default, SAM employs a 32×32 grid of points over the entire document to segment and generate K segmentations or bounding boxes representing the different objects/patterns in the document.

  Subsequently, the input document is segmented, cropped from the original document while maintaining its aspect ratio and resized to 224×224. This is then processed through the S3BIR-CLIP/DINOv2 image encoder using the respective photo prompt learning. This process generates K embeddings, represented by $e_p \in \mathbb{R}^{K\times512}$, in the case of CLIP.

- Online phase: given a sketch of dimension 224×224, it is processed directly by the image encoder S3BIR-CLIP/DINOv2 using the sketch prompt learning, obtaining its embedding represented by $e_s \in \mathbb{R}^{1\times512}$. Then, the cosine similarity between e_p and e_s is computed to obtain those segmentations that have the greatest similarity to the input sketch. Finally, the bounding box of those segmentations that exceed a threshold is recovered.

4.3.3 SAM + Clusters Proposal

In the aforementioned proposal, SAM is utilized with the predefined values. This configuration entails SAM employing 1024 points distributed equidistantly throughout the document to segment the image. However, this configuration presents two challenges. The first challenge is that, as observed in the EDA (4.1), there are very small queries that are searched within the documents, and that SAM with the default configuration is unable to detect. One potential solution is to increase the number of points used to identify these patterns. However, this introduces a second challenge: the computational cost. While it is possible to increase the number of points used by SAM, this would result in a greater use of resources that may not always be available.

A more efficient strategy was therefore sought, with the aim of indicating to SAM in which parts of the document there may be patterns of interest. A document resized to 224 × 224 is used as input for the image encoder of DINOv2. It is divided into patches of dimensions $16 \times 16 \times 3$, and a vector of dimension 256 represents each patch. Figure 4.9 shows an example of the above. This proposal is based on calculating the similarity between the 256 patches. Therefore, given $P \in \mathbb{R}^{256\times768}$, which represents the embedding matrix of the patches, the dot product between P and its transpose P^T is calculated, thus obtaining the similarity between all the patches.

Once the similarity matrix between the patches has been calculated, the K-means clustering algorithm is employed to identify the distinct groups present in the document. In particular, the value of $K = 3$ is assumed, as it is postulated that there are three distinct clusters present within a document: background, text and figures.

The assumption is that the cluster with the highest intra-variance contains the images and figures. The hypothesis that underlies this approach is straightforward: the background of the different pages of the documents is typically homogeneous, suggesting that the variance within this cluster should be low. A similar phenomenon should occur with the text, as it typically comprises black letters of comparable size and on a uniform background. Consequently, its intra-variance is expected to be relatively low. Finally, it is postulated that the cluster exhibiting the highest intra-variance contains the figures and images, given that these vary in size, color, and shape.

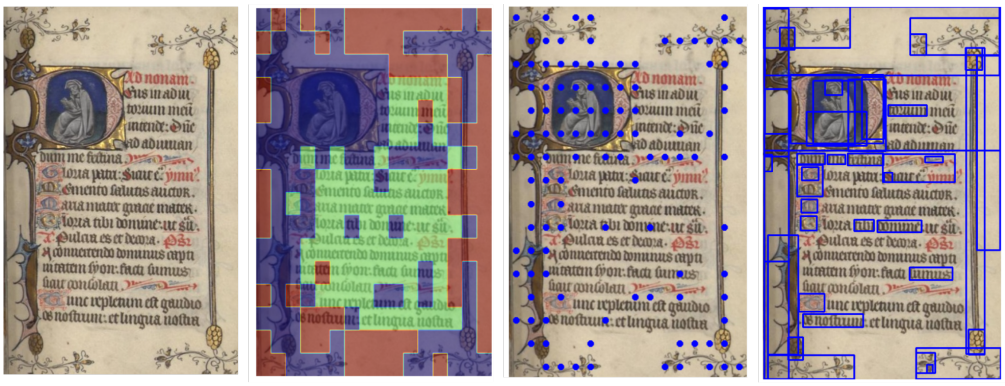


Figure 4.12: Results achieved in the clustering process. From left to right is presented: 1) The original input document. 2) The clustering result on the similarity matrix between patches. 3) The points generated on the highest intra-variance cluster. 4) The regions proposed by SAM.

Finally, the region with the highest intra-variance is the one that will be assigned points to later give them to SAM and segment the areas of the document where the figures and images are likely to be. Figure 4.12 exemplifies what was explained above. It is necessary to highlight that, as the chosen cluster is made up of patches, the center of each of these will be the points (as foreground) that SAM will receive.

4.4 Experimental Results

This section presents the results obtained from the models under the different proposals evaluated in DocExplore. In particular, the outcomes of the three most effective S3BIR models for each dataset are presented in table 3.3. The primary objective was to assess the models capacity for generalization within a highly specific context.

It is also noteworthy that DocExplore provides an evaluation tool developed by them to assess the performance of the models in two specific tasks: image retrieval and pattern spotting.

4.4.1 Quantitative Results

Table 4.1 presents the results of the three S3BIR models evaluated in the tasks of image retrieval and pattern recognition under the three proposals. In the image retrieval task, it can be observed that the base proposal produces comparable results across the three models. The SAM and SAM + Clusters proposals achieve the highest and similar mean average precision (mAP) for the S3BIR-DINOv2 (QD-Extended) model, with 27.9% and 25.3%, respectively.

Conversely, in the pattern recognition task, the S3BIR-CLIP model exhibited the least favorable performance among the three proposals. In contrast, the S3BIR-DINOv2 (Flickr25K) model achieved the best results using the SAM proposal, followed by the model trained in QD-Extended. The SAM + Clusters proposal yielded unsatisfactory results, with a difference of approximately one percentage point compared to the SAM proposal.

	Image Retrieval			Pattern Spotting		
Models	CC	SAM	SAM + Clusters	CC	SAM	SAM + Clusters
S3BIR-CLIP (eCommerce)	0.101	**0.141**	0.123	0.022	**0.054**	0.039
S3BIR-DINOv2 (Flickr25K)	0.124	0.223	**0.230**	0.039	**0.222**	0.123
S3BIR-DINOv2 (QD-Extended)	0.131	**0.279**	0.253	0.030	**0.210**	0.140

Table 4.1: The mAP of the image retrieval and pattern detection results of the S3BIR models in the different proposals is reported.

Table 4.2 illustrates the performance of the S3BIR-DINOv2 (QD-Extended) model across different sizes and aspect ratios on both tasks. In the domain of image retrieval, the Cross Correlation proposal achieved the highest mAP value of 38.0% for the 37 large square queries. The SAM proposal achieved 72.6% mAP for the 183 small square queries and 22.2% for the 1206 small non-square queries. The SAM + Clusters proposal achieved 64.4% and 19.0% mAP for small square and non-square queries, respectively, and 58.7% for large non-square queries.

In the pattern recognition task, the SAM proposal achieved the highest mAP values, with 31.4% and 69.3% for large square and small square queries, respectively. The SAM + Clusters proposal achieved a mAP of 14.2% for the 21 large non-square queries, outperforming the other proposals in that specific case.

Size	Aspect ratio	# queries	Freq	S3BIR-DINOv2 (QD-Extended)					
				Image Retrieval			Pattern Spotting		
				CC	SAM	SAM + Clusters	CC	SAM	SAM + Clusters
big	square	37	0.03	**0.380**	0.049	0.143	0.141	**0.314**	0.098
small	square	183	0.13	0.194	**0.726**	0.644	0.064	**0.693**	0.445
big	non-square	21	0.01	0.241	0.046	**0.587**	0.047	0.120	**0.142**
small	non-square	1206	0.83	0.103	**0.222**	0.190	0.021	**0.135**	0.093

Table 4.2: Performance (mAP) of the S3BIR-DINOv2 (QD-Extended) model in Image Retrieval and Pattern Spotting.

Table 4.3 illustrates the performance of the S3BIR-DINOv2 (Flickr25K) model, which employs a similar structure to that of the previous table. In the context of image retrieval, the Cross Correlation proposal achieved the highest mAP of 35.1% for large square queries. The SAM proposal achieved 69.7% mAP for the 183 small square queries and 15.6% for the 1206 small non-square queries. The SAM + Clusters proposal achieved 61.2% mAP for small square queries, with the highest values of 38.2% and 17.4% for large non-square and small non-square queries, respectively.

Size	Aspect ratio	# queries	Freq	S3BIR-DINOv2 (Flickr25K)					
				Image Retrieval			Pattern Spotting		
				CC	SAM	SAM + Clusters	CC	SAM	SAM + Clusters
big	square	37	0.03	**0.351**	0.119	0.043	0.133	**0.241**	0.035
small	square	183	0.13	0.217	**0.697**	0.612	0.055	**0.688**	0.435
big	non-square	21	0.01	0.336	0.111	**0.382**	0.040	**0.153**	0.104
small	non-square	1206	0.83	0.098	0.156	**0.174**	0.033	**0.152**	0.084

Table 4.3: Performance (mAP) of the S3BIR-DINOv2 (Flickr25K) model in Image Retrieval and Pattern Spotting.

In the pattern recognition task, the SAM proposal exhibits the highest values for all types of patterns being searched. Conversely, the SAM + Clusters proposal achieved 10.4% mAP for the 21 large non-square queries, while the Cross Correlation proposal achieved 13.3% mAP for large square queries.

Table 4.4 illustrates the average number of points utilized by SAM for each image in the document, as well as the total number of segmentations performed for the 1,500 pages in the document.

Method	Number of points (average)	Number of segmentations	Used memory (MB)
SAM	1024	114,169	351
SAM + Clusters	84	41,987	130

Table 4.4: The table shows the number of points used in each of the proposed solutions, together with the corresponding number of segmentations performed and the memory occupied (in megabytes) by the embeddings.

Table AA.1 presents a detailed analysis of the S3BIR-DINOv2 (QD-Extended) model for each query. The class, a reference image, and the frequency corresponding to each query are presented. Furthermore, the outcomes of the principal proposals, SAM and SAM + Clusters, in the domains of image retrieval and pattern recognition are presented. Table A.2 presents the same information, but for the S3BIR-DINOv2 (Flickr25k) model.

4.4.2 Qualitative Results

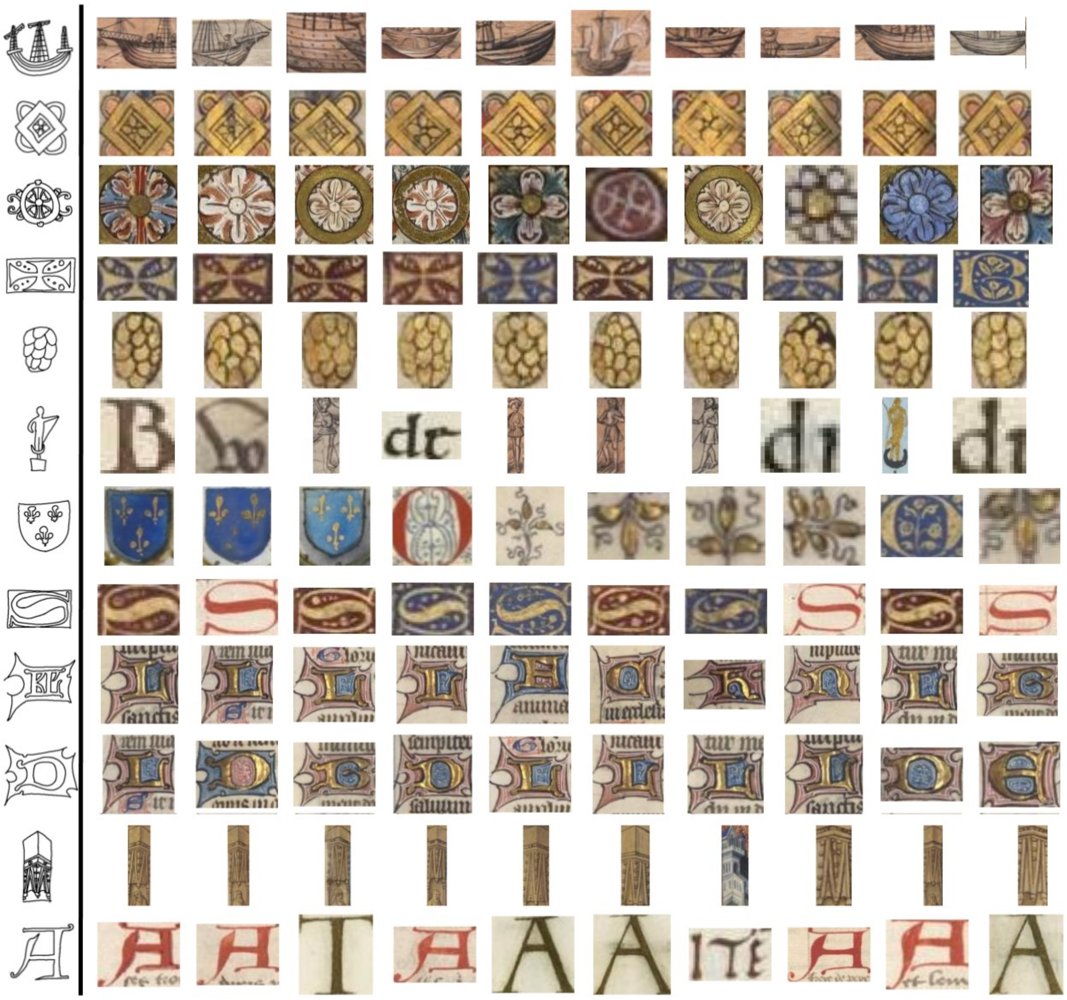


Figure 4.13: Examples of sketch-base image retrieval using S3BIR-DINOv2 (QD-Extended) using the SAM approach.

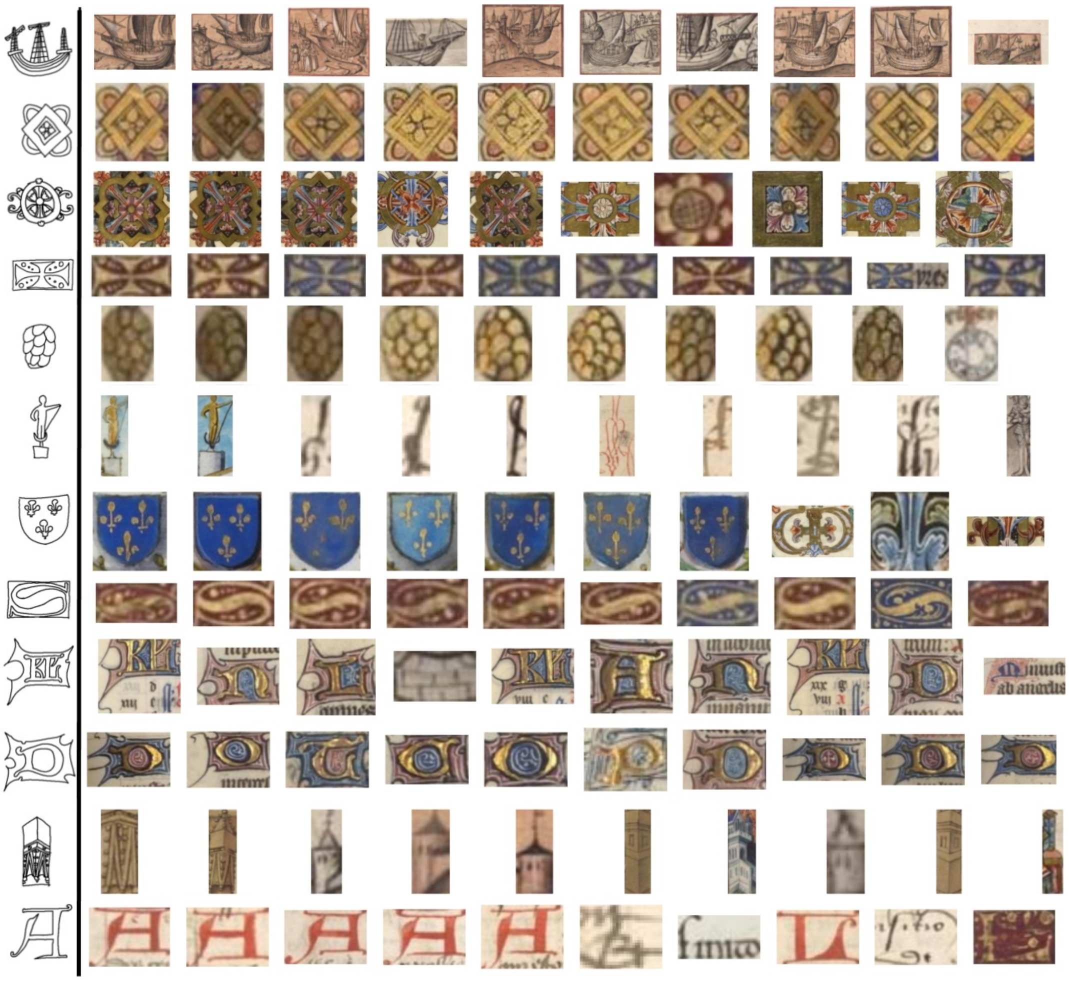


Figure 4.14: Examples of sketch-base image retrieval using S3BIR-DINOv2 (Flickr25K) using the SAM approach.

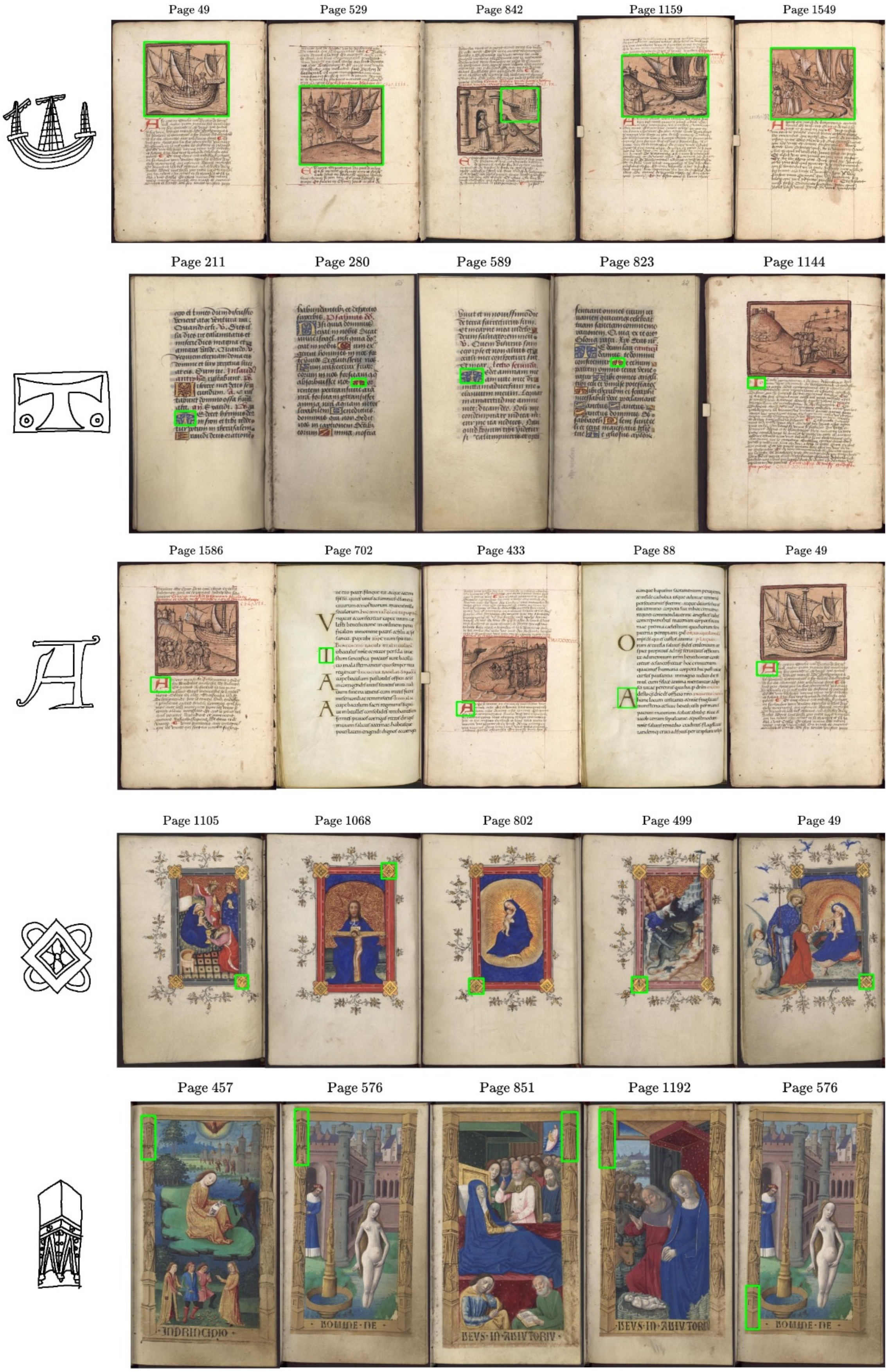


Figure 4.15: Examples of sketch-base detection using S3BIR-DINOv2 (QD-Extended) using the SAM approach. The green bounding box represents the detection based on the query.

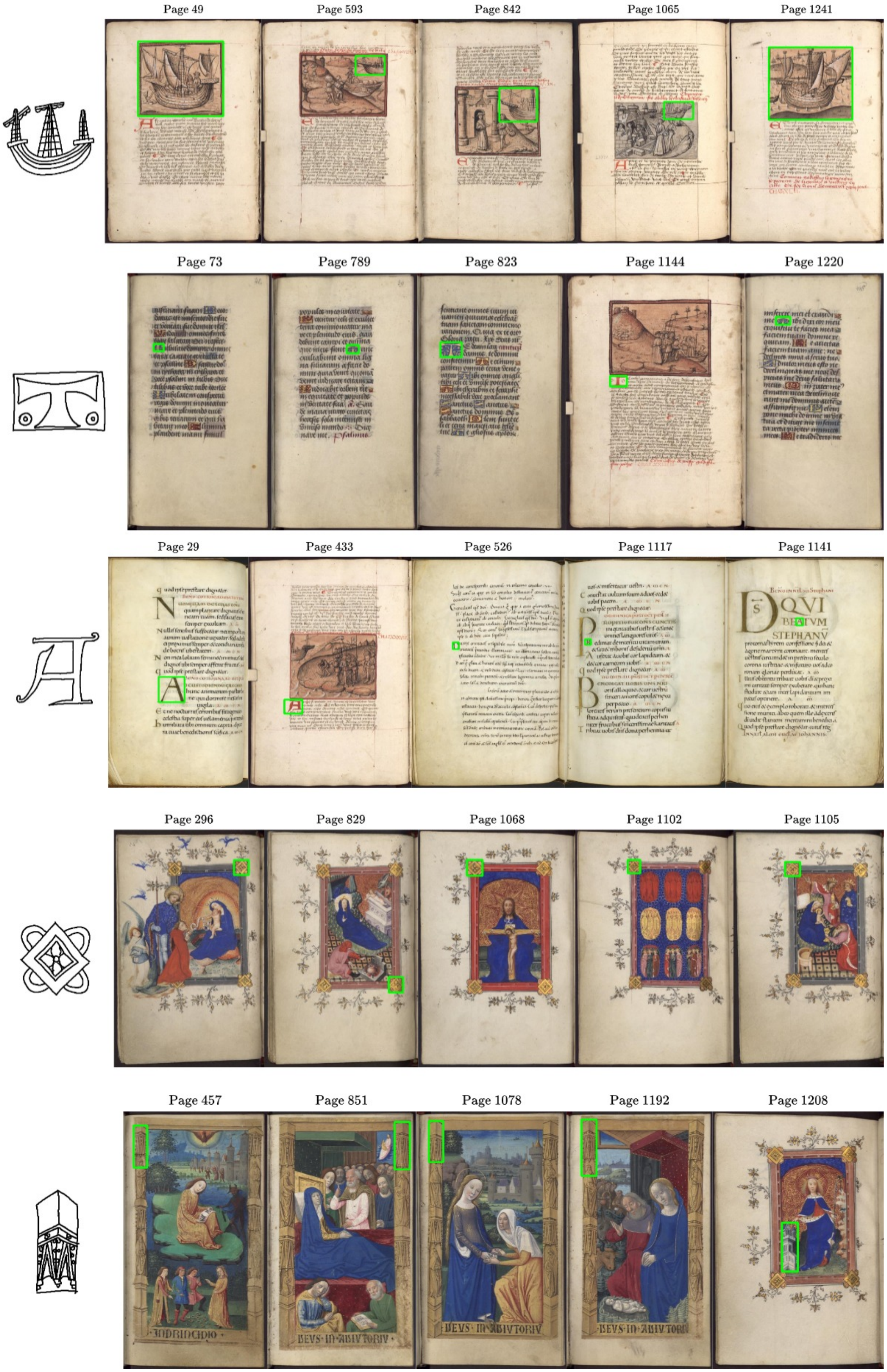


Figure 4.16: Examples of sketch-base detection using S3BIR-DINOv2 (Flickr25K) using the SAM approach. The green bounding box represents the detection based on the query.

Chapter 5

Discussion

This chapter presents a critical analysis and interpretation of the results obtained from the evaluation of the various proposals for image retrieval and pattern detection in the DocExplore dataset. Furthermore, a comparison with previous research is conducted, and the limitations of each proposal are discussed.

The quantitative results presented in table 4.1 demonstrate that the SAM proposal outperforms the baseline proposal on both tasks, particularly when applied to the S3BIR-CLIP and S3BIR-DINOv2 models trained on the QD-Extended and eCommerce datasets. These results are comparable to those presented by Úbeda et al. [65], Wiggers et al. [61], and En et al. [20], who, despite not working with sketches, also incorporate deep learning-based models to solve image retrieval and pattern spotting tasks.

In order to contextualize these results, table 5.1 presents a comparison of the proposed methods with those defined by Úbeda, Wiggers, Dias and En *et al.* in terms of mAP for the tasks of image retrieval and pattern spotting, considering only the first 1000 retrievals and IoU equal to 0.5 respectively. While the results are not directly comparable due to the use of sketches as queries in this work, this comparison demonstrates the generalization of the S3BIR models and the ability of SAM to segment previously unseen figures and patterns.

Method	Image Retrieval	Pattern Spotting
Úbeda *et al.* ES [65]*	0.286	0.139
Úbeda *et al.* PP [65]*	0.386	0.178
Wiggers *et al.* [61]*	0.386	0.174
Dias *et al.* [9]*	0.486	0.199
En *et al.* [20]*	**0.580**	0.157
Baseline proposal	0.131	0.03
SAM proposal	0.279	**0.210**

* it is not a sketch-based model because it has access to the real query.

Table 5.1: Results of the Image Retrieval and Pattern Spotting task on the DocExplore dataset. Values refer to the mAP of the 1000 best candidates.

It is crucial to highlight that although the SAM proposal does not demonstrate superior performance compared to previous methods in image retrieval, it exhibits competitive performance (mAP of 0.279), particularly when considering the additional challenge of using sketches as queries. Moreover, in the pattern spotting task, the SAM proposal outperforms all previous methods, achieving a mAP of 0.210.

The analysis by size and aspect ratio of tables 4.2 and 4.3 reveals valuable information about the behavior of the different proposals:

- In the majority of cases (83% for square queries and 13% for non-square queries), SAM consistently outperforms CC and, to a lesser extent, SAM + Clusters in both tasks.
- The SAM proposal demonstrates suboptimal performance on large, non-square queries in both tasks. This indicates that in instances where exceptionally large and intricate shapes are being sought, the 1024-point limit may be insufficient.

The SAM proposal shows evident robustness compared to the other two. This can be justified as follows:

- Point density: as is well known, SAM employs 1024 equidistant points within the input image or document, thus enabling the segmentation of a significant proportion of the figures and patterns present, as well as elements that are not relevant, such as text. The high point density enables more detailed and precise segmentation of the patterns being sought.
- Limitations of CC and SAM + Clusters: both approaches rely on ViT patches as a fundamental component of the solution. This presents an inherent problem: the relationship between heatmap and cluster granularity. When employing a ViT-B/14, there will be 14 × 14 patches of size 16 × 16, which may result in a single patch encompassing a combination of text, background, and figure in the most unfavorable scenario. Moreover, in the case of SAM + Clusters, the patch's central point may not select the figure being searched for, as illustrated in figure 5.1.

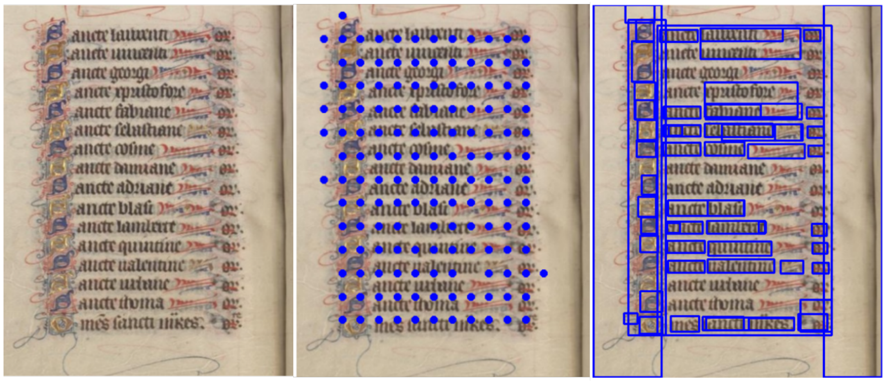


Figure 5.1: The points generated by the clusters approach are not sufficiently accurate to permit the selection of some regions on the right, and thus SAM is unable to segment them.

In addition to the performance results, it is crucial to analyze the computational efficiency of the proposals. The table 4.4 provides valuable information about the number of points used, the segmentations performed by SAM and SAM + Clusters, and the amount of memory used by these sets. This data reveals important aspects of the efficiency and performance of both proposals:

- Efficiency of SAM + Clusters: This proposal employs a significantly reduced number of points per image (84 on average, in contrast to 1024 for SAM). It achieves competitive results, particularly for larger queries. Figure 5.2 segments all relevant patterns with only 61 points.
- Resource reduction: The SAM + Clusters proposal performs a total of 47, 987 segmentations, in comparison to SAM's 114, 169. For purposes of comparison, Wiggers et al. [61] proposal identified over 36 million potential regions, while Dias et al. [9] proposal identified over 750, 000.

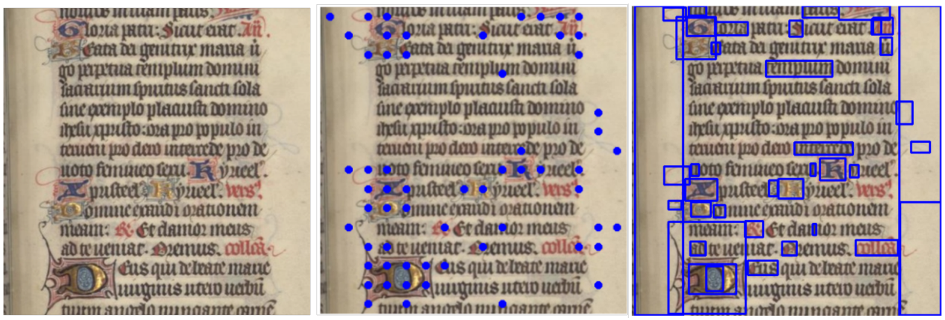


Figure 5.2: The generated points fit exactly on the patterns being searched for, allowing SAM to segment correctly.

Table A.1 presents a comprehensive analysis of the performance of the S3BIR-DINOv2 (QD-Extended) model under two different query proposals: SAM and SAM + Clusters. The frequency of appearance of each class is displayed, accompanied by a small reference image. It is important to note that many of the low-frequency figures or patterns, such as the figures "Round Ornament", "Brace Ornament", "Leaf Ornament", and even "Henri II's Profile", tend to appear in pages containing drawings, and it is in these cases that the SAM + Clusters proposal gives worse results in the IR and PS tasks. This is due to the strict assumption of 3 clusters, since in the cases with no text, only one of the two clusters with a figure remains.

However, the SAM proposal demonstrates consistent performance across all queries. If it is not outperformed by SAM + Clusters, it is considered to be close. In PS, SAM is differential in most queries, presenting challenges in those queries that are large and non-square, such as "Pidiment", "Round ornament 3" and "Flower ornament" which result in a mAP of 0 in some cases.

It is important to note that the inherent quality of the sketch plays an important role when searching for the figure within the documents. In the table A.1 it is possible to see the searched queries. In some of them it is possible to see figures that have many colors and

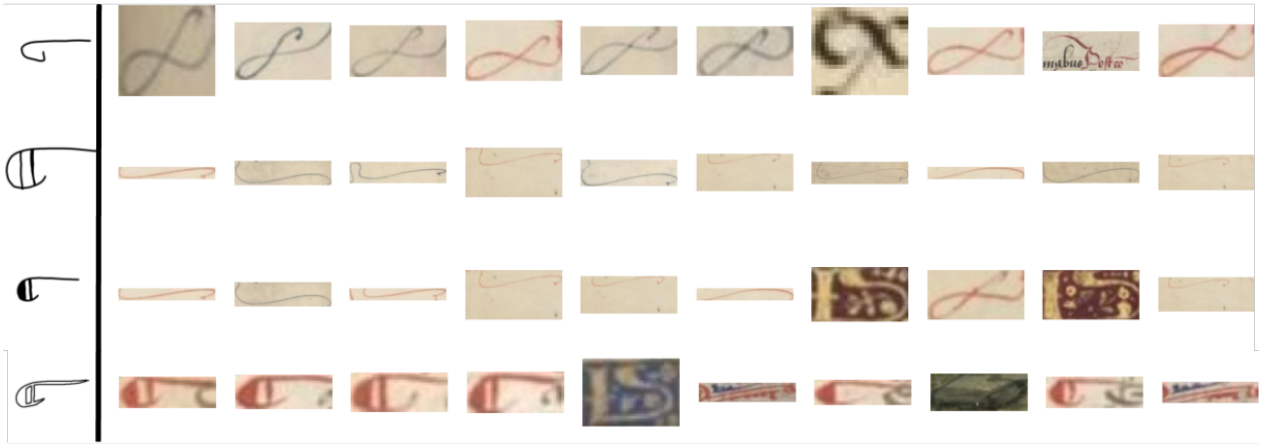


Figure 5.3: Sketches of varying complexity and detail for the class “Marker”.

shapes. Therefore, if the sketch does not faithfully represent the searched figure, you will get results like those shown in the figure 5.3.

Chapter 6

Conclusion

As digitization becomes a more widespread tool for preserving books and cultural heritage documents, it is becoming increasingly important to develop efficient and generalizable methods for searching for patterns and figures across documents. Manual searching of large volumes of documents is not only a complex and costly process, but also carries risks of deterioration in the handling of these documents. For this reason, solutions are needed that take advantage of novel approaches to improve the efficiency of professionals working in this area.

This thesis proposes a new approach that explores the use of sketches for pattern recognition in cultural heritage documents. The main challenge of working with sketches is the scarcity of photo-sketch data pairs when training these models. To overcome this limitation, a self-supervised approach is used, i.e., these models are able to generate a bimodal photo-sketch space without the need for explicitly paired data. The self-supervised models used, S3BIR-DINOv2 and S3BIR-CLIP, have shown competitive results, achieving an mAP of 61% on the Flickr15k dataset, 45% on Ecommerce, and 17% on QD-Extended.

In conjunction with these models, SAM was used to extract regions of interest in the documents and facilitate the localization of relevant patterns. Evaluation on the DocExplore dataset has shown that this approach is competitive in image retrieval (IR) and pattern segmentation (PS) tasks, achieving a mAP of 27.9% for IR and 21% for PS. These results validate the hypothesis that it is possible to develop a SAM-based architecture capable of locating patterns using a photo-sketch data pair as input, approaching the performance of state-of-the-art models that do not use sketches.

One of the main limitations of this research lies in the extraction of relevant elements in the documents. In this work, SAM is used to solve this task, the problem is that it is class agnostic, i.e. it will segment everything found by the 1024 entry points, detecting much more than the relevant objects being searched for. With the “SAM + clusters” proposal, we tried to filter the document so that SAM searches in areas where there may be figures, but with a strong assumption of using three clusters and selecting the one with the highest internal variance, the cluster containing the figures is not always selected, resulting in a deterioration of the mAP in a 33,3%. The quality of the input sketch also plays an important role in performing the search. Although more than 400 DocExplore queries were hand-drawn in detail for evaluation in this work, a greater variety of drawing styles and levels of detail would be necessary to evaluate the robustness of the model to different types of input.

6.1 Future Work

Although the results obtained in this thesis are competitive with proposals that do not use sketches, there is ample room for improvement to achieve a performance close to 1. The main directions in which this work could be extended are presented below:

- **Segmentation models**: One of the major limitations of the current approach is the use of SAM as the initial segmentation model. Since SAM is class agnostic, it segments all the elements it detects into its 1024 input points, which can introduce significant noise into the detection. A promising improvement would be the implementation of GroundingDINO [39], a model that allows segmentation of images using textual prompts as search conditioners. This approach would allow the extraction of relevant figures using specific prompts such as “figure - building - roof - sketch - person - symbol”, as illustrated in [18].
- **Fine-tuning**: Current models are capable of extracting representative features from different objects, such as DINO and CLIP. However, to maximize the performance of these encoders for a specific domain, it is crucial to strategically tune them to avoid catastrophic forgetting phenomen. In this work, we used prompt learning as a fine-tuning method, but there are other methods such as LORA [39] that have shown good performance, especially in LLMs.
- **Datasets Expansion**: This research focused on the evaluation using the DocExplore dataset. To validate the generalization of the proposed methods, it would be an idea to extend the evaluation to another dataset of historical documents. In this respect, a candidate dataset is HORAE [3], which contains more than 100,000 images of documents from different books, mainly from France. This extension would allow validating the robustness of the different methods in different historical document contexts.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[3] Mélodie Boillet, Marie-Laurence Bonhomme, Dominique Stutzmann, and Christopher Kermorvant. Horae: an annotated dataset of books of hours. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, pages 7–12, 2019.

[4] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John Collomosse. Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics*, 71, 01 2018.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640, 2021.

[6] Abhra Chaudhuri, Ayan Kumar Bhunia, Yi-Zhe Song, and Anjan Dutta. Data-free sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12084–12093, 2023.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. 2020.

[9] Caio da S. Dias, Alceu S. Britto Jr., Jean Paul Barddal, Laurent Heutte, and Alessandro L. Koerich. Pattern spotting and image retrieval in historical documents using deep hashing. In *SMC*, pages 2869–2875. IEEE, 2022.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[13] Sounak Dey, Pau Riba, Anjan Dutta, Josep Lladós, and Yi-Zhe Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2179–2188, 2019.

[14] Diego Donoso and Jose M. Saavedra. Survey on sketch-to-photo translation. *ACM Comput. Surv.*, 56(1), aug 2023.

[15] 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. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[16] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012.

[17] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1624–1636, 2011.

[18] Hassan El-Hajj and Matteo Valleriani. Prompt me a dataset: An investigation of text-image prompting for historical image dataset creation using foundation models. In Gian Luca Foresti, Andrea Fusiello, and Edwin Hancock, editors, *Image Analysis and Processing - ICIAP 2023 Workshops*, pages 247–257, Cham, 2024. Springer Nature Switzerland.

[19] Sovann En, Stéphane Nicolas, Caroline Petitjean, Frédéric Jurie, and Laurent Heutte. New public dataset for spotting patterns in medieval document images. *Journal of Electronic Imaging*, 26(1):011010, 2016.

[20] Sovann En, Caroline Petitjean, Stéphane Nicolas, and Laurent Heutte. A scalable pattern spotting system for historical documents. *Pattern Recognition*, 54:149–161, 2016.

[21] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Process. Mag.*, 39(3):42–62, 2022.

[22] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector, 2020.

[23] Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, and Jon Wetzel. Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 3(4):648–666, 2011.

[24] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735, 2020.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] D. L. Hoffmann, C. D. Standish, M. García-Diez, P. B. Pettitt, J. A. Milton, J. Zilhão, J. J. Alcolea-González, P. Cantalejo-Duarte, H. Collado, R. de Balbín, M. Lorblanchet, J. Ramos-Muñoz, G.-Ch. Weniger, and A. W. G. Pike. U-th dating of carbonate crusts reveals neandertal origin of iberian cave art. *Science*, 359(6378):912–915, 2018.

[29] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding*, 117(7):790–806, July 2013.

[30] David H Hubel and Torsten N. Wiesel. *Brain and Visual Perception: The Story of a 25Year Collaboration Illustrated Edition.* Oxford University Press, London, 2004.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, dec 1989.

[33] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, feb 2006.

[34] Xiang Li, Wenhai Wang, Lingfeng Yang, and Jian Yang. Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. *arXiv preprint arXiv:2205.10063*, 2022.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[37] Weidong Lin, Yuyan Deng, Yang Gao, Ning Wang, Jinghao Zhou, Lingqiao Liu, Lei Zhang, and Peng Wang. CAT: cross-attention transformer for one-shot object detection. *CoRR*, abs/2104.14984, 2021.

[38] LITIS. Pattern spotting in medieval document images. `http://spotting.univ-rouen.fr/`. [Accedido el 16 de julio de 2023].

[39] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[40] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *EMNLP*, pages 3698–3707. Association for Computational Linguistics, 2018.

[41] Javier Morales, Nils Murrugarra-Llerena, and Jose M. Saavedra. Leveraging unlabeled data for sketch-based understanding. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022*, 2022.

[42] 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 G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *CoRR*, abs/2304.07193, 2023.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[44] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[46] Jose M. Saavedra. Sketch based image retrieval using a soft computation of the histogram of edge local orientations (S-HELO). In *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, pages 2998–3002, 2014.

[47] Jose M. Saavedra and Benjamin Bustos. An improved histogram of edge local orientations for sketch-based image retrieval. In Michael Goesele, Stefan Roth, Arjan Kuijper, Bernt Schiele, and Konrad Schindler, editors, *Pattern Recognition - 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proceedings*, volume 6376 of *Lecture Notes in Computer Science*, pages 432–441, 2010.

[48] Aneeshan Sain, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. Clip for all things zero-shot sketch-based image retrieval, fine-grained or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2765–2775, 2023.

[49] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016.

[50] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6836–6845, 2017.

[51] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[52] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[53] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5097–5107, 2021.

[54] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[55] Pablo Torres and Jose M. Saavedra. Compact and effective representations for sketch-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 2115–2123, 2021.

[56] Aditay Tripathi, Rajath R. Dani, Anand Mishra, and Anirban Chakraborty. Sketch-guided object localization in natural images. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 532–547, 2020.

[57] Aditay Tripathi, Rajath R Dani, Anand Mishra, and Anirban Chakraborty. Sketch-guided object localization in natural images, 2020.

[58] Aditay Tripathi, Anand Mishra, and Anirban Chakraborty. Query-guided attention in vision transformers for localizing objects using a single sketch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1083–1092, 2024.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[60] Dirk B. Walther, Barry Chai, Eamon Caddigan, Diane M. Beck, and Li Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011.

[61] Kelly Lais Wiggers, Alceu de Souza Britto Junior, Alessandro Lameiras Koerich, Laurent Heutte, and Luiz Eduardo Soares de Oliveira. Deep learning approaches for image retrieval and pattern spotting in ancient documents. *CoRR*, abs/1907.09404, 2019.

[62] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[63] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[64] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017.

[65] Ignacio Úbeda, Jose M. Saavedra, Stéphane Nicolas, Caroline Petitjean, and Laurent Heutte. Pattern spotting in historical documents using convolutional models, 2019.

ANNEXES

Annex A

A.1 Metrics

When working with an object detection problem, it is necessary to quantify how many objects are correctly and incorrectly identified in order to understand the performance of the proposed model.

A.1.1 Precision

The first metric is called precision, which is expressed as the fraction of labels that are correctly predicted as positive. It is written as (.1):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{.1}$$

where TP is defined as the number of positive correct values and FP is defined as the number of positive incorrect values.

A.1.2 Recall

We will also use the metric Recall, which is expressed as the fraction of correctly predicted real labels. It is formulated as (.2):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{.2}$$

where FN is the number of negative false values.

A.1.3 mAP (mean Average Precision)

The mAP metric is typically used to evaluate the performance of image retrieval systems. The Average Precision (AP) refers to the area under the curve of the trade-off between precision-recall. That is, the mAP calculates the average of the APs for each "query".

We will define B as a set of images (database) to be searched on. Q will be our set of queries and R will be the ranking, i.e., the set of images ordered by similarity. Finally, we will define the set of relevant images for each query (.3):

$$\Gamma_q = \{x \in B, q \in Q | x \text{ is relevant for } q\} \tag{.3}$$

Its indicative function is defined as (.4):

$$f_{\Gamma_q} = \begin{cases} 1 & x \in \Gamma_q \\ 0 & \text{in another case} \end{cases} \tag{.4}$$

Then, it is possible to define how to calculate the *precision* (.5) and the *AP* (.6) for each *query q*:

$$\text{precision}_q(i; R) = \frac{\sum_{k=1}^{i} f_{\Gamma_q}(r_k)}{i} f_{\Gamma_q}(r_i) \tag{.5}$$

$$\text{AP}_q(R) = \frac{\sum_{i=1}^{|R|} \text{precision}_q(i; R)}{|\Gamma_q|} \tag{.6}$$

Finally, with *precision* and *AP* defined, we can calculate the *mean average precision*, which is defined as (.7):

$$\text{mAP} = \frac{\sum_{q \in Q} AP_q(R)}{|Q|} \tag{.7}$$

A.1.4 Cosine Similarity

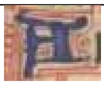
Given two vectors x and y, and considering the l_2 norm denoted as $\| \cdot \|_2$ and the dot product as $\cdot$, the cosine similarity is defined by the equation .8.

$$\text{cosine similarity} = \frac{x \cdot y}{\|x\|_2 \cdot \|y\|_2} \tag{.8}$$

When the vectors x_1 and x_2 are normalized using the l_2 norm, the denominator becomes 1. Therefore, the dot product between the vectors only depends on the angle difference between them. This means that, the cosine similarity measures the relative orientation of the vectors without considering the magnitude of the vectors.

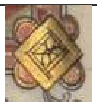
A.2 Average Precision of the best two models by class

			S3BIR-DINOv2 (QD-Extended)			
			Image Retrieval		Pattern Spotting	
Classes	Image	Freq	SAM	SAM + Cls.	SAM	SAM + Cls.
B initial ribborn		3	0.0013	**0.0291**	0	0
Bed Crown		4	0.0980	**0.5500**	**0.7391**	0.3129
Brace ornament		4	0.0069	**0.1250**	**1.0000**	0
Coat of arms		8	**0.8456**	0.6497	**0.8430**	0.3750
Corner Diamond		117	**1.0000**	0.8696	**0.9879**	0.6542
Cross		92	**0.3573**	0.0814	**0.6090**	0.2974
Diamond ornament		2	**0.0204**	0.0135	**0.0260**	0
Double Separator		87	0.0345	**0.0593**	0	**0.0001**
Flower ornament		8	**0.0365**	0.0076	0	0
Frame		60	0.6352	**0.6542**	0.1267	**0.3185**
Henri II's Profile L		3	0.0027	**0.2391**	0	0
Henri II's Profile R		5	0.0758	**0.2771**	0	0
Initial ribborn		68	**0.1520**	0.1504	0.0587	**0.0677**

Leaf ornament 1		3	**0.1806**	0.0154	**0.2005**	0
Leaf ornament 2		7	**0.0714**	0.0603	**0.0653**	0
Letter A		15	0.2238	**0.2338**	0.1770	**0.2044**
Letter A small		35	0.0418	**0.0663**	**0.0053**	0.0029
Letter D		35	**0.2061**	0.1468	0.1249	**0.2327**
Letter S		147	**0.4455**	0.3253	**0.3638**	0.2282
Letter T		39	**0.2851**	0.0502	**0.2456**	0.0201
Letters BP		12	0.0076	**0.0774**	**0.5556**	0.2306
Marker		409	0.1207	**0.1526**	**0.0021**	0.0005
Pediment		8	0.0054	**0.8091**	0	**0.1139**
Pine cone		29	**0.8855**	0.8254	**0.7918**	0.5478
Round ornament 1		2	**0.0417**	0.0020	0	0
Round ornament 3		2	0.0028	**0.0035**	0.0906	**0.4226**
Round ornament 3		4	0.0107	**0.6250**	0	0
Ship		13	0.0519	**0.6116**	**0.1945**	0.1557
Ship hull L		12	**0.3651**	0.3068	0.0171	**0.0346**
Ship hull R		6	0.0623	**0.1203**	0.0049	**0.0086**
Simple round ornament		2	1.0000	1.0000	**0.6124**	0.2536
Simple separator		160	**0.1291**	0.0907	0	0
Square ornament		4	**0.0792**	0.0345	0	0

Statue		12	0.0627	**0.3654**	**0.0590**	0.0428
Triple Separator		30	0.0175	**0.0546**	0	0.0006

Table A.1: Results in Image Retrieval and Pattern Spotting by query of the S3BIR-DINOv2 (QD-Extended) model on the SAM proposal.

			S3BIR-DINOv2 (Flickr25K)			
			Image Retrieval		Pattern Spotting	
Classes	Image	Freq	SAM	SAM + Cls.	SAM	SAM + Cls.
B initial ribborn		3	**0.0021**	0.0003	0	0
Bed Crown		4	**0.0143**	0.0097	**0.8809**	0.0018
Brace ornament		4	**0.5170**	0.0014	**0.9437**	0
Coat of arms		8	**0.8591**	0.3916	**0.7836**	0.3750
Corner Diamond		117	**0.9610**	0.8467	**0.9808**	0.6166
Cross		92	0.1995	**0.3001**	**0.6105**	0.4820
Diamond ornament		2	**0.0085**	0.0011	**0.0455**	0
Double Separator		87	0.0322	**0.0369**	0	**0.0025**
Flower ornament		8	**0.0247**	0.0027	0	0
Frame		87	0.0994	**0.2425**	**0.0779**	0.0575
Henri II's Profile L		3	0.0214	**0.0313**	0	0
Henri II's Profile R		5	0.0758	**0.0981**	0	0
Initial ribborn		68	**0.0335**	0.0163	**0.0559**	0.0008
Leaf ornament 1		3	0.0002	**0.0004**	**0.2119**	0
Leaf ornament 2		7	**0.0063**	0.0015	**0.0445**	0

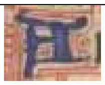
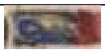
Letter A		15	0.3065	**0.3562**	0.1285	**0.2407**
Letter A small		35	0.0395	**0.0420**	0.0023	**0.0038**
Letter D		35	0.1495	**0.1839**	**0.1820**	0.1057
Letter S		147	0.2290	**0.2668**	**0.4209**	0.1225
Letter T		39	**0.1746**	0.1373	**0.3908**	0.0562
Letters BP		12	**0.1707**	0.1187	**0.3504**	0.1078
Marker		409	0.1728	**0.1754**	0.0141	**0.0230**
Pediment		8	0.1760	**0.7853**	0	**0.1235**
Pine cone		29	**0.8793**	0.7844	**0.8231**	0.6510
Round ornament 1		2	**0.0545**	0.0003	0	0
Round ornament 3		2	**0.0817**	0.0015	0.3843	**0.5878**
Round ornament 3		4	0.0230	**0.6250**	0	0
Ship		13	0.1284	**0.1412**	**0.1365**	0.0371
Ship hull L		12	0.0877	**0.1076**	**0.0877**	0.0008
Ship hull R		6	**0.0852**	0.0346	**0.0070**	0.0018
Simple round ornament		2	1.0000	1.0000	**0.6057**	0.5000
Simple separator		160	0.0768	**0.0889**	**0.0009**	0.0001
Square ornament		4	0.0012	**0.0029**	0	0
Statue		12	**0.2113**	0.1424	**0.2069**	0.0293
Triple Separator		30	0.0111	**0.0185**	0	0

Table A.2: Results in Image Retrieval and Pattern Spotting by query of the S3BIR-DINOv2 (Flickr25K) model on the SAM proposal.

A.3 Qualitative Examples of SAM + Proposed Clusters

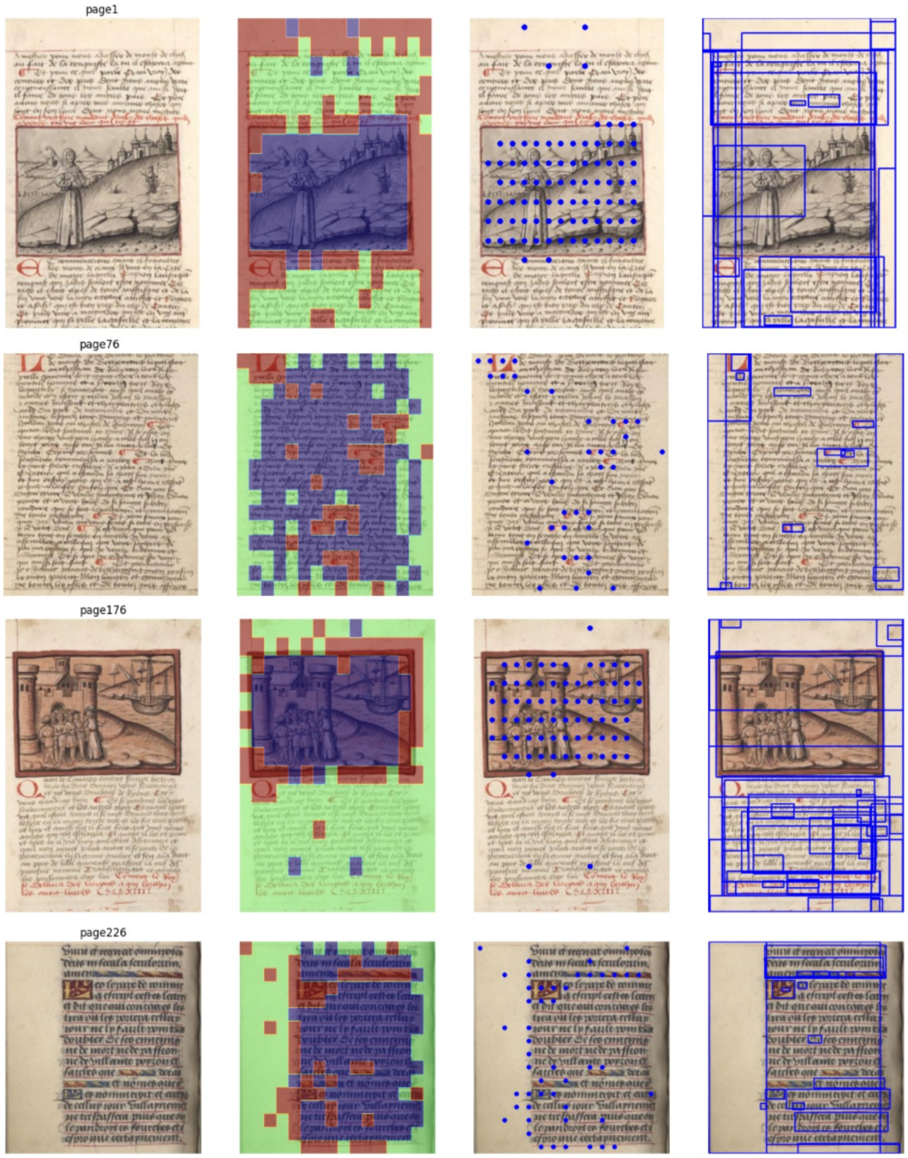


Figure A.1: SAM + Cluster approach results with prompting.

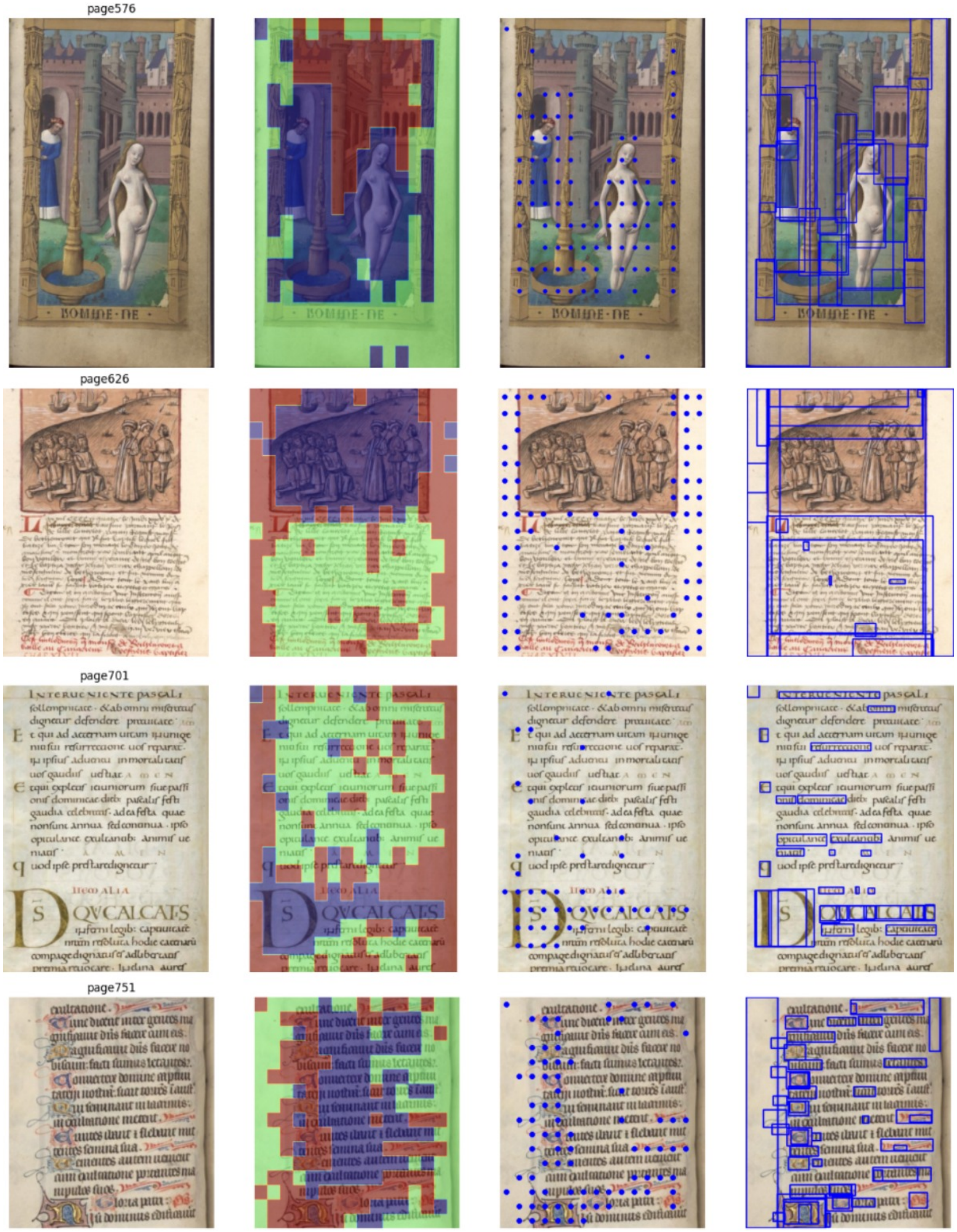


Figure A.2: SAM + Cluster approach results with prompting.

A.4 Results achieved

Here we present the main results achieved, focusing on the development of novel models for sketch-based image retrieval and one-shot detection, as well as the papers submitted for publication. The core of the work lies in the self-supervised sketch-based image retrieval, which allows training SBIR models without the need for real sketches. This methodology has shown excellent results on three different public datasets, demonstrating its robustness and generalizability. In addition, S3BIR models have been combined with SAM for the one-shot detection task, which has yielded competitive results on tasks such as image retrieval and pattern spotting on the DocExplore dataset.

A list of the achievements is provided below:

- **Trained SBIR Models**
  - S3BIR trained without using real sketches, based primarily on:
    - CLIP.
    - DINOv2.
- **One-Shot Detection Models**
  - Proposals for One-Shot Detection using SAM in combination with S3BIR models:
    - SAM + S3BIR-CLIP.
    - SAM + S3BIR-DINOv2.
- **Papers**
  - The results of the SBIR models have been summarized in two papers currently under review:
    - A Study on Self-Supervised Sketch-based Image Retrieval on Unpaired Dataset (S3BIR).
    - A Self-Supervised Learning Methodology for Sketch-based Image Retrieval on Unpaired Datasets.
  - The preliminary results on photo-to-photo models have been summarized in a paper currently under review:
    - Achieving High Performance on Pattern Spotting in Historical Documents by Self-Supervised Learning and Grounding Models.