

CAIO DA SILVA DIAS

**PATTERN SPOTTING AND IMAGE  
RETRIEVAL IN HISTORICAL  
DOCUMENTS USING DEEP HASHING**

Dissertation presented to the Graduate Program in Informatics of the Pontifícia Universidade Católica do Paraná (PUCPR) as a partial requirement for the degree of Master in Informatics.

Curitiba  
2023

CAIO DA SILVA DIAS

PATTERN SPOTTING AND  
IMAGE RETRIEVAL IN  
HISTORICAL DOCUMENTS  
USING DEEP HASHING

Dissertation presented to the Graduate Program in Informatics of the Pontifícia Universidade Católica do Paraná (PUCPR) as a partial requirement for the degree of Master in Informatics.

Concentration area: Computer Science

Advisor: Alceu de Souza Britto Junior  
Co-advisors: Jean Paul Barddal and Laurent Heutte

Curitiba  
2023

Dados da Catalogação na Publicação  
Pontifícia Universidade Católica do Paraná  
Sistema Integrado de Bibliotecas – SIBI/PUCPR  
Biblioteca Central  
Sônia Maria Magalhães da Silva – CRB 9/1191

D541p  
2023  
Dias, Caio da Silva  
Pattern spotting and image retrieval in historical documents using deep hashing  
/ Caio da Silva Dias ; advisor: Alceu de Souza Brito Junior ; co-advisors: Jean Paul  
Bardal, Laurent Heutte. – 2023  
ix, 52 f. ; il. : 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba,  
2023  
Bibliografia: f. 49-52

1. Aprendizado do computador. 2. Redes neurais (Computação). 3. Hashing  
(Ciência da computação). 4. Informática. I. Brito Junior, Alceu de Souza. II.,  
Barddal, Jean Paul. III. Heutte, Laurent. IV. Pontifícia Universidade Católica do  
Paraná. Programa de Pós-Graduação em Informática. VI. Título.

CDD. 20. ed. – 004



Pontifícia Universidade Católica do Paraná  
Escola Politécnica  
Programa de Pós-Graduação em Informática

Curitiba, 21 de novembro de 2023.

89-2023

## **DECLARAÇÃO**

Declaro para os devidos fins, que **CAIO DA SILVA DIAS** defendeu a dissertação de Mestrado intitulada “**PATTERN SPOTTING AND IMAGE RETRIEVAL IN HISTORICAL DOCUMENTS USING DEEP HASHING**”, na área de concentração Ciência da Computação no dia 21 de setembro de 2023, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

---

Prof. Dr. Emerson Cabrera Paraiso  
Coordenador do Programa de Pós-Graduação em Informática

# Acknowledgements

I would like to express my heartfelt gratitude to my family for their unwavering support throughout this challenging yet rewarding journey. To my mother and father, your love, encouragement, and sacrifices have been my pillars of strength. Your belief in my abilities has been a driving force, and I am profoundly thankful for your enduring support. A special acknowledgment goes to my brother, whose camaraderie and encouragement provided a welcome respite during intense research phases.

I am also deeply grateful to my girlfriend for her patience, understanding, and encouragement. Your unwavering support and belief in my dreams have been a constant source of inspiration. Thank you for being my anchor during both the highs and lows of this academic endeavor.

I extend my sincere appreciation to my advisors, Prof. Alceu, Prof. Jean and Prof. Laurent, for their guidance, mentorship, and invaluable feedback. Your expertise and dedication to the pursuit of knowledge have been instrumental in shaping the trajectory of my research. I am fortunate to have had the opportunity to learn under your tutelage.

Finally, I would like to express my thanks to all those who contributed, directly or indirectly, to the successful completion of this master's dissertation. Your support, insights, and encouragement have left an indelible mark on my academic journey.

# Contents

<b>Acknowledgements</b>	i
<b>Contents</b>	ii
<b>List of Algorithms</b>	iv
<b>List of Figures</b>	v
<b>List of Tables</b>	vii
<b>Resumo</b>	viii
<b>Abstract</b>	ix
<b>Chapter 1</b>	
<b>Introduction</b>	1
1.1 Objectives . . . . .	4
1.2 Hypotheses . . . . .	4
1.3 Proposal . . . . .	5
1.4 Contributions . . . . .	6
1.5 Publications . . . . .	6
1.6 Organization . . . . .	7
<b>Chapter 2</b>	
<b>Fundamentals</b>	8
2.1 Image retrieval . . . . .	9
2.2 Pattern Spotting . . . . .	11
2.3 Object Detection . . . . .	12
2.3.1 EdgeBoxes . . . . .	13
2.3.2 CornerNet . . . . .	14
2.3.3 Selective Search . . . . .	15
2.4 Feature Extraction . . . . .	18
2.4.1 Deep features . . . . .	19

2.4.2	Deep Hashing . . . . .	20
2.5	Similarity Measures and Distances . . . . .	21
2.6	Final Considerations . . . . .	23
<b>Chapter 3</b>		
<b>Related Works</b>		24
3.1	Final Considerations . . . . .	28
<b>Chapter 4</b>		
<b>Proposed Method</b>		29
4.1	Object Detection with Selective Search . . . . .	30
4.2	Feature Extraction using Deep Learning . . . . .	32
4.3	Feature Extraction using Deep Hashing . . . . .	35
4.4	Similarity Calculations . . . . .	36
4.5	Final Considerations . . . . .	37
<b>Chapter 5</b>		
<b>Experimental Results</b>		38
5.1	Experimental Protocol . . . . .	38
5.1.1	DocExplore . . . . .	39
5.2	Selective Search . . . . .	40
5.3	Image Retrieval Task . . . . .	40
5.4	Pattern Spotting Task . . . . .	42
5.5	Search Time and Storage Cost . . . . .	46
5.6	Final Considerations . . . . .	46
<b>Chapter 6</b>		
<b>Conclusions</b>		48
<b>Bibliography</b>		49

# List of Algorithms

1	Invalid candidate region filter . . . . .	31
---	---	----



# List of Figures

1.1	Example of how a single page of a historical document can contain several figures and special characters to be analyzed. . . . .	2
1.2	Image search process in IR. The image retrieval algorithm returns a list of non-repeating pages ordered by the distance measure. . . . .	3
1.3	Image search process in PS. The pattern spotting returns a list of non-repeating positions ordered by the distance measure and the document page. These positions are subject to minimal overlap between the query image and the processed images measured through the IoU. . . . .	3
2.1	Illustration of intersection over union (IoU). . . . .	12
2.2	In this example, it is shown how the scale variation influences the final result.	17
3.1	Overview of the pattern spotting system proposed in (EN et al., 2016b). . .	26
3.2	Overview of the pattern spotting system proposed in (ÚBEDA et al., 2019).	26
3.3	Overview of the pattern spotting system proposed in (WIGGERS et al., 2019). . . . .	28
4.1	Overview of the proposed method with selective search and siamese networks.	30
4.2	Examples of regions filtered out by the invalid candidate region algorithm.	31
4.3	Example of data preparation for the training set created using the Imagenet dataset. . . . .	33
4.4	Structure example of a Siamese Neural Network. . . . .	34
5.1	Samples of historical document pages available in DocExplore. . . . .	39
5.2	Qualitative results of the search of some queries in the DocExplore database. The figure shows the image used in the query and its first five results returned by the ResNet Conv method. . . . .	45

5.3	Qualitative results of the search of some queries in the DocExplore database. The figure shows the image used in the query and its first five results re- turned by the VGG19 Block4-5 Hashing method. . . . .	46
-----	--	----

# List of Tables

5.1	Image Retrieval results . . . . .	41
5.2	Image Retrieval results with Hashing . . . . .	41
5.3	Comparison of the methods with the state-of-the-art of IR. . . . .	42
5.4	Pattern Spotting results . . . . .	43
5.5	Pattern Spotting results with Hashing . . . . .	43
5.6	Pattern Spotting results with PP . . . . .	44
5.7	Comparison of the methods with the state-of-the-art PS . . . . .	45
5.8	Results for processing time and storage . . . . .	47

# Resumo

Este trabalho apresenta uma abordagem de aprendizagem profunda para image retrieval e pattern spotting em coleções digitais de documentos históricos. Um algoritmo de proposta de região foi usado para detectar candidatos a objetos nas imagens das páginas do documento. Modelos de aprendizagem profunda foram utilizados para extração de características, oferecendo duas variantes distintas que resultam em representações de código real ou binário. Posteriormente, a similaridade de características entre as imagens candidatas e uma imagem de pesquisa é calculada para classificar os resultados. Para avaliar a eficácia da abordagem proposta, foi seguido um protocolo experimental rigoroso usando o banco de dados de imagens DocExplore. Os resultados experimentais demonstram que os modelos profundos propostos superam os métodos de image retrieval do estado da arte para imagens de documentos históricos, superando outros modelos profundos em 2,56 pontos percentuais no pattern spotting. Além disso, a abordagem proposta reduz significativamente o tempo de busca em até 200 vezes e os custos de armazenamento em até 6.000 vezes em comparação com trabalhos existentes baseados em representações de valores reais.

**Palavras-chave:** Aprendizagem de máquina; Redes neurais convolucionais; Reconhecimento de objetos; Hashing; Localização de padrões.

# Abstract

This work introduces a deep learning approach for image retrieval and pattern recognition in digital collections of historical documents. A region proposal algorithm is employed to detect object candidates in the document page images. Deep learning models are then utilized for feature extraction, offering two distinct variants that yield either real-valued or binary code representations. Subsequently, the feature similarity between the candidate images and a given input query is computed to rank the results. To evaluate the effectiveness of the proposed approach, a rigorous experimental protocol is followed using the DocExplore image database. The experimental results demonstrate that the proposed deep models outperform state-of-the-art image retrieval methods for historical document images, surpassing other deep models by 2.56 percentage points in pattern recognition. Additionally, the proposed approach significantly reduces search time by up to 200 times and storage costs by up to 6,000 times compared to existing works based on real-valued representations.

**Keywords:** Machine learning; Convolutional neural networks; Object recognition; Hashing; Pattern spotting

# Chapter 1

## Introduction

Content-based image retrieval (CBIR), in particular the tasks of image retrieval (IR) and pattern spotting (PS), quickly evolved in recent years and has become essential in the area of computer vision. IR involves retrieving a set of images from a collection that contains a specific search image (query) based on their content. For each new query, a search is performed in the image collection, returning potential candidate images where the query may be found. In the same way, pattern spotting provides candidate images and identifies the precise locations of query occurrences. In the domain of historical documents, candidate images typically correspond to images of document pages.

The exponential growth of image collections held by art museums, medical institutes, environmental agencies, and governmental organizations has led to a significant information access challenge. Typically, these images are manually indexed by individuals who assign keywords to categorize them and facilitate future retrieval. However, this indexing process is time-consuming and expensive. An exemplary case is the digitization of historical document collections, which aims to increase accessibility to their content while ensuring the preservation of the original manuscripts. Since many of these documents date back to the 10th–16th centuries, continued physical handling poses risks that could damage these valuable artifacts. Consequently, historians rely on digitized documents to establish correlations between various elements, such as text and graphics, without further subjecting the fragile originals to harm.

Existing indexing methods rely on automated detection and search software, enabling efficient analysis of vast document collections. Nevertheless, with recent advancements in computer vision and machine learning, it is now feasible to develop applications capable of swiftly identifying correlations within seconds. These cutting-edge technologies empower the automation of correlation discovery, revolutionizing the indexing process and significantly expediting the retrieval of relevant information.

Historical documents predominantly consist of handwritten texts but can also feature a variety of graphical elements (YARLAGADDA et al., 2010), as illustrated in Figure 1.1 (an example from the DocExplore database). These graphic objects encompass special characters, text separators, intricate borders, stamps, coats of arms, and even vivid depictions of festive scenes. For historians, the primary challenge lies in establishing connections among diverse objects across multiple document collections. Such correlations offer valuable insights into cultural and temporal heritage, enabling characterizing patterns in figures and paintings, content-based document categorization, and analyzing writing variations (YARLAGADDA et al., 2010).

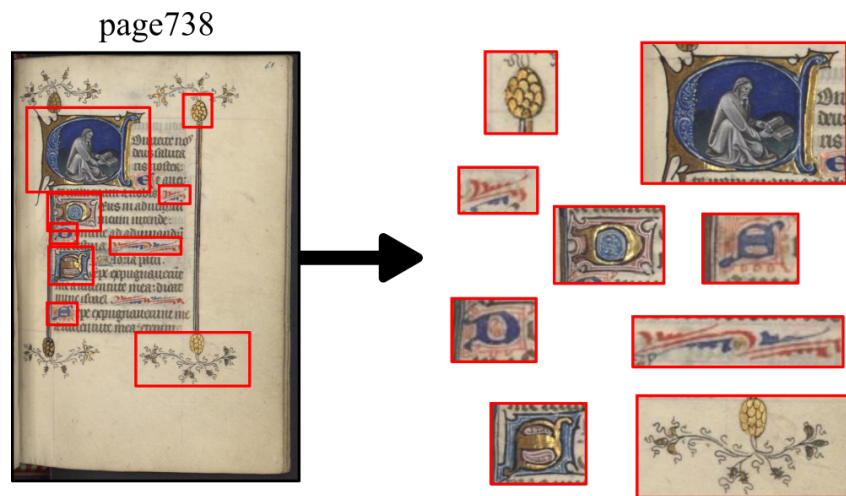


Figure 1.1: Example of how a single page of a historical document can contain several figures and special characters to be analyzed.

As mentioned earlier, IR involves searching for a page that contains the query in its content, while PS focuses on locating the specific occurrences of the query within that page (which may have multiple results). This close relationship between IR and PS implies that IR is an integral part of the PS process, as providing the location of queries within a document enables the identification of the corresponding page.

Various methods exist for object retrieval in images, but most involve offline and online phases. In the offline step, document image files are processed by an object detector. The candidate regions within these files are analyzed and segmented into separate files, then stored in the system to create a candidate image database for future searches. The processed images are indexed and organized into a predefined structure that includes information such as each image's page, position, and path. In the online phase, the similarity between the search image and the images within the stored candidate regions is computed, resulting in a similarity ranking. This ranking selects the  $n$  smallest distances to form the outcomes for both the IR and PS tasks.

One practical approach for calculating the similarity measure between the search and candidate region images is through hashing comparison. Hashing is widely utilized due to its computational and storage efficiency (CANTINI et al., 2021). The primary goal of hashing is to convert high-dimensional feature maps into low-dimensional hash codes. This transformation ensures that hash codes of similar objects are closer to each other, while hash codes of different things are more distinct. Low-dimensional hash codes enable faster similarity calculations compared to high-dimensional feature maps. Additionally, this approach reduces the storage requirements for storing processed image resources, as the hash codes are much smaller.

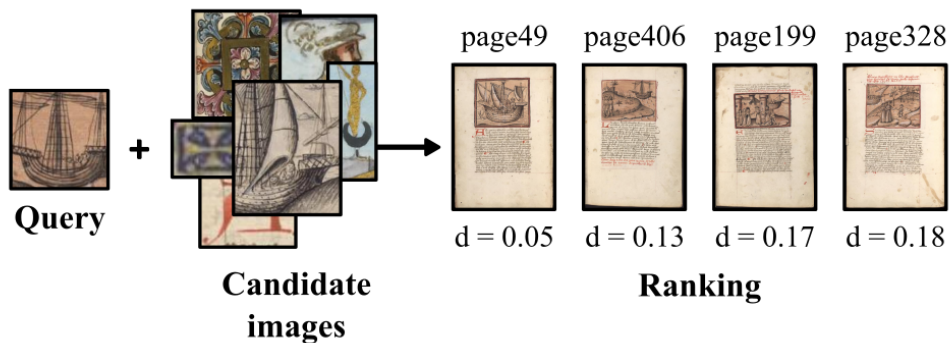


Figure 1.2: Image search process in IR. The image retrieval algorithm returns a list of non-repeating pages ordered by the distance measure.

In the IR task, the shortest ranking distances of the Top  $n$  candidates are used to return the list of pages where the query can be found, as shown in Figure 1.2. However, in PS, the image location within the document is also required in addition to returning the pages. For such an aim, the structure previously stored in the offline phase is used, and it returns this location, as shown in Figure 1.3 (green rectangles).

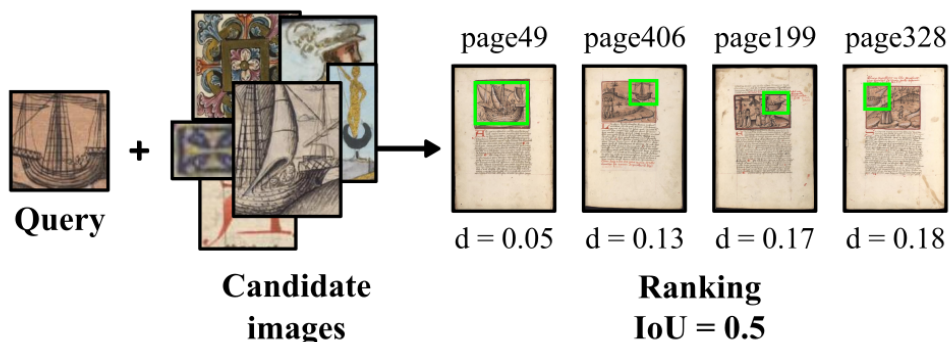


Figure 1.3: Image search process in PS. The pattern spotting returns a list of non-repeating positions ordered by the distance measure and the document page. These positions are subject to minimal overlap between the query image and the processed images measured through the IoU.



The relevance of a candidate region is determined by its intersection over union (IoU) with the query area. The IoU is obtained by dividing the intersection of the query area and the candidate region's area by their union. To analyze potential candidates, a threshold of  $\text{IoU} \geq 0.5$  was utilized (EN et al., 2016a). Precision and recall metrics are calculated, and the mean average precision (mAP) is then computed to evaluate the overall result across all queries.

## 1.1 Objectives

The primary objective of this work is to enhance the existing results in terms of processing time, storage cost, and accuracy of Segmentation based approaches. The problem comprising the query and image candidates will be represented using deep hashing techniques to achieve this goal. To fulfill this objective, the following tasks will be carried out:

1. Conducting a literature review on image retrieval and pattern spotting, deep features, deep hashing, and object candidate search methods;
2. Establishing a methodology for generating object candidates from collections of document images;
3. Define a deep representation that is compact and discriminant for queries and image candidates;
4. Implement the solutions using deep hashing for fast image retrieval and pattern spotting;
5. Evaluating the proposed IR and PS solutions using a benchmark dataset, considering accuracy, processing time, and storage cost. This evaluation will involve comparing the performance of the proposed approach against state-of-the-art methods.

## 1.2 Hypotheses

Given the significance of research focused on IR and PS in historical documents, this work presents several hypotheses, each of which will be examined and evaluated in the following sections.

**Hypothesis #1.** The utilization of binary representations of features results in reduced processing time for IR and PS tasks compared to real-value representations.

**Hypothesis #2.** Binary representations of features lead to reduced storage costs for IR and PS tasks compared to real-value representations.

**Hypothesis #3.** Binary representations yield higher precision levels in IR and PS tasks than real-valued representations.

**Hypothesis #4.** Optimizing the parameters of the object detector (Selective Search) improves the performance of image retrieval and pattern spotting tasks.

## 1.3 Proposal

This work presents an approach for addressing image retrieval and pattern spotting tasks in the context of historical documents by leveraging deep learning techniques. The primary objective is to enhance the current state-of-the-art methods by improving accuracy, reducing processing time, and minimizing storage costs. To achieve these goals, the proposed approach involves exploring and evaluating both real-valued and binary representations generated using diverse deep model architectures.

The proposed method encompasses the following key steps:

1. **Object Candidate Generation:** A method will be devised to identify and generate object candidates within the document images. This process aims to extract regions of interest that may contain relevant information for retrieval and pattern spotting.
2. **Deep Representation Design:** The design of deep representations will be explored to create compact yet discriminative feature embeddings for both the queries and the generated image candidates. This step captures essential visual information and characteristics for accurate matching and retrieval.
3. **Deep Hashing for Efficient Retrieval:** Deep hashing techniques will transform the high-dimensional feature representations into dynamic binary codes. This enables faster retrieval by efficiently comparing and matching the query with the stored image candidates.
4. **Performance Evaluation:** A comprehensive evaluation framework will be employed to assess the proposed solutions for image retrieval and pattern spotting tasks. Accuracy, processing time, and storage cost will be considered as evaluation metrics. The proposed approaches will be benchmarked against state-of-the-art methods to demonstrate their effectiveness and superiority.

Through the execution of this research, the aim is to advance the field of image retrieval and pattern spotting in historical documents by harnessing the potential

of deep learning. The proposed methods and techniques are expected to yield substantial improvements in accuracy, processing efficiency, and storage optimization, thereby facilitating efficient exploration and analysis of historical document collections.

## 1.4 Contributions

The first contribution is applying a filtering strategy during the offline phase to effectively reduce the number of candidate images generated by the Selective Search approach. This strategy aims to enhance the efficiency and effectiveness of subsequent processing steps by eliminating irrelevant or redundant candidates. Reducing the candidate set reduces the overall computational load, leading to improved processing time and resource utilization.

The second contribution is the proposal of a binary representation method to address the challenges associated with storage space and query search time. This method significantly reduces the space required to store feature maps during the offline phase and speeds up the search process for a given query during the online stage. By utilizing binary representations, the storage overhead is minimized without compromising the accuracy of the retrieval and pattern spotting tasks. This approach enables faster query search and reduces storage costs, making it more practical and efficient for large-scale historical document collections.

Overall, these contributions enhance the existing methods for image retrieval and pattern spotting in historical documents by improving processing efficiency, reducing storage requirements, and maintaining or even improving the system’s accuracy. The proposed filtering strategy and binary representation method collectively contribute to advancing the field by addressing important challenges and providing more effective solutions for efficient retrieval and analysis of historical document images.

## 1.5 Publications

- Caio da Silva Dias, Alceu de Souza Britto Jr, Jean Paul Barddal, Laurent Heutte and Alessandro L. Koerich. *Pattern Spotting and Image Retrieval in Historical Documents using Deep Hashing*. 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022.

## 1.6 Organization

This work is structured into six chapters to provide a comprehensive analysis of the IR and PS tasks, as well as the proposed methods and experimental results.

Chapter 2 focuses on presenting the problem statement of IR and PS tasks, as well as discussing related topics. It is divided into six sections: Image retrieval, Pattern spotting, Object detection, Feature extraction, Similarity measures and Distances, and Final considerations. Each section delves into relevant concepts and theories related to the tasks.

Chapter 3 discusses related works in the field and examines their contributions to the current research. This chapter critically analyzes existing approaches and highlights their impact on the development of the proposed methods.

In Chapter 4, the proposed IR and PS methods are presented in detail. Which includes the candidate generation technique, feature extraction approaches utilizing convolutional neural networks, the evaluation protocol used, and the similarity measure employed. The chapter comprehensively explains the methodologies and algorithms employed in the proposed methods.

Chapter 5 presents the experimental results obtained from the conducted experiments. This chapter showcases the outcomes of the IR and PS tasks and compares the performance of the proposed methods with existing approaches documented in the literature. The performance of the proposed IR and PS methods is further compared with available methods in the literature. This evaluation is conducted to assess the effectiveness and efficiency of the proposed methods.

Finally, Chapter 6 concludes the work by summarizing the main findings and presenting the conclusions drawn from the research. This chapter provides a concise overview of the research outcomes and discusses potential directions for future research.

# Chapter 2

## Fundamentals

The storage of various data collections, including images, documents, books, and more, provides information that specialists from different fields can explore. However, in traditional collections, the organization and labeling of document information could be better defined, particularly in collections composed of document images. This lack of predefined structures poses challenges in defining suitable queries due to the inherent difficulty in understanding the data semantics. The information retrieval community has recognized this as a significant problem and has proposed various approaches to address the indexing and retrieval of such collections.

Given the large volume and importance of these documents, developing efficient methods for accessing this information is crucial. Numerous techniques can be employed to retrieve information from an image collection. These techniques typically follow a conventional process: first, the candidates extracted from the images are indexed and represented in a suitable feature space. This indexing process is often performed in an offline phase. Subsequently, during the online phase, users can submit queries to the retrieval system, and a similarity measure is employed to compare the query with the stored image candidates, resulting in a ranked list. This iterative process continues until a stopping criterion is met.

To address these challenges and provide a comprehensive understanding of the selected processes involved in the creation of an image retrieval and pattern spotting system, this chapter presents the key definitions. Section 2.1 provides a literature review of the Image Retrieval task, while Section 2.2 focuses on Pattern Spotting. The object detection is discussed in Section 2.3. Furthermore, Section 2.4 emphasizes using deep learning for feature extraction, which plays a crucial role in effectively representing the images. Finally, Section 2.5 introduces various similarity measures that can be utilized in the retrieval process.

## 2.1 Image retrieval

Image retrieval is a fundamental task in the field of computer vision and information retrieval, which aims to retrieve relevant images from a large collection based on a given query. With the increasing availability of digital image collections, effective image retrieval systems play a crucial role in various domains, including art, medicine, e-commerce, and social media.

The primary objective of image retrieval is to provide users with efficient and accurate means of searching and retrieving images based on their content, rather than relying solely on textual descriptions or metadata (SMEULDERS et al., 2000). This enables users to explore and navigate image databases in a more intuitive and visual manner, facilitating tasks such as content-based image browsing, image organization, and similarity-based image recommendation.

Conceptually, image retrieval involves two main components: representation and similarity measurement.

1. **Representation:** The representation involves extracting and encoding the visual content of images into a feature space that can capture the characteristics of the images. These features can be derived from various visual descriptors, such as color, texture, shape, and spatial layout. Popular techniques for feature extraction include Convolutional Neural Networks (CNNs), which have demonstrated exceptional performance in learning rich and discriminative representations from images (SMEULDERS et al., 2000).
2. **Similarity Measurement:** Once the images are represented by their respective feature vectors, the next step is to measure the similarity or distance between the query image and the images in the database. Various similarity metrics are employed, such as Euclidean distance, Cosine similarity, or a combination of multiple distance measures. The choice of similarity metric depends on the nature of the features and the specific requirements of the application (SMEULDERS et al., 2000).

Image retrieval methods can be categorized into two main approaches: content-based image retrieval (CBIR) and text-based image retrieval (TBIR).

1. **Content-Based Image Retrieval:** CBIR methods focus on retrieving images based on their visual content. These methods leverage the extracted visual features of images to measure the similarity between the query image and the images in the database. CBIR systems are particularly useful when textual annotations or metadata are limited or unavailable (MUNJAL; BHATIA, 2019).

2. Text-Based Image Retrieval: TBIR methods, on the other hand, rely on textual information, such as captions, tags, or annotations, to retrieve images. These methods involve matching the textual query with the available textual information associated with the images. TBIR systems are advantageous when textual descriptions are well-curated and comprehensive (MUNJAL; BHATIA, 2019).

In recent years, there has been a growing trend towards integrating both textual and visual information in hybrid image retrieval systems, aiming to exploit the complementary nature of these modalities for improved retrieval performance (MUNJAL; BHATIA, 2019). This integration allows users to efficiently search and retrieve images from large-scale collections by leveraging advanced techniques in feature representation and similarity measurement. By combining textual and visual cues, image retrieval systems provide valuable tools for visual exploration, organization, and recommendation.

The evaluation of image retrieval performance is essential to assess the effectiveness of these systems. It involves returning a list of non-repeating images sorted by their confidence level in containing the query object. Mean Average Precision (mAP) is a commonly used metric for evaluating image retrieval tasks. It measures the area under each query’s precision/recall curve, providing a comprehensive assessment of the system’s ability to retrieve relevant images. By considering both precision and recall, mAP offers a quantitative measure of the retrieval performance, allowing researchers to compare and analyze different retrieval algorithms and techniques.

Precision is defined as the ratio of true positive images retrieved to the total number of positive images retrieved, considering true positives (TP) and false positives (FP) as shown in Equation 2.1.

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Recall measures the ability of the retrieval system to identify all relevant images. It is defined as the ratio of true positive images retrieved to the total number of positive images in the corpus, considering true positives (TP) and false negatives (FN) as shown in Equation 2.2.

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

It is important to note that if an image returned by the system does not contain any correct occurrences of the query object but includes unrelated objects, it will be considered irrelevant, and the evaluation system will disregard it in the result list. This ensures that only relevant images are considered for performance evaluation.

## 2.2 Pattern Spotting

Pattern spotting (PS) is a fundamental task in computer vision that focuses on the detection and localization of specific objects or patterns of interest within an image. Its primary objective is to identify and localize instances of predefined objects or patterns, which in turn enables automated analysis and comprehension of visual content. In recent years, pattern spotting has made significant strides, driven by the advancements in deep learning and the availability of large-scale labeled datasets. These advancements have resulted in enhanced accuracy and robustness in localizing objects or patterns of interest, paving the way for new applications in fields like autonomous vehicles, augmented reality, and industrial automation. By precisely detecting and localizing objects or patterns of interest, pattern spotting empowers advanced capabilities and provides valuable insights in these domains.

Pattern spotting typically involves the use of machine learning techniques, particularly deep learning-based approaches, due to their ability to learn complex patterns and features from data. Convolutional Neural Networks (CNNs) are commonly used for pattern spotting tasks due to their effectiveness in capturing hierarchical representations of visual features.

The process of pattern spotting consists of two main steps: training and inference. During the training phase, a pattern spotting model is trained on a labeled dataset, where the objects or patterns of interest are annotated with bounding boxes. The model learns to recognize and localize these objects by extracting relevant features and optimizing the parameters through the use of a loss function. Once the model is trained, it can be used for inference on new, unseen images. During inference, the pattern spotting model analyzes the input image and generates predictions of the presence and location of the desired objects or patterns. These predictions are typically represented as bounding boxes or pixel-level masks.

Evaluation in pattern spotting goes beyond the image retrieval task, as it assesses not only the relevance of returned objects but also their precise alignment with the ground-truth object positions. The evaluation involves analyzing the location accuracy of the detected objects within the image. Instead of returning a list of non-repeated images, the system generates a list of objects with the image name and corresponding bounding boxes. To measure the system's ability to locate image regions containing the object, the Intersection over Union (IoU) criterion is utilized. IoU measures the overlap between the bounding box of the true object (Region 1) and that of the returned object (Region 2), as illustrated in Figure 2.1.



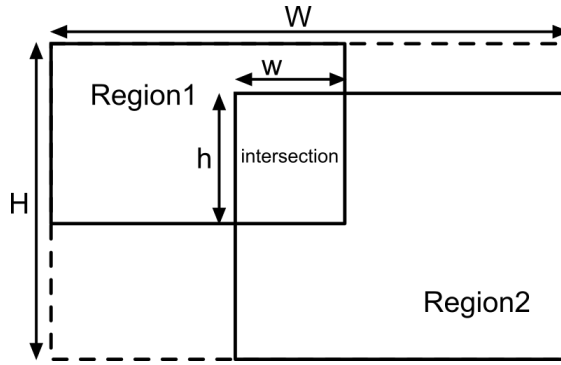


Figure 2.1: Illustration of intersection over union (IoU).

The intersection considers the query area, represented by  $q_1$ , and the area of the candidate region, represented by  $o_1$ , as defined by Equation 2.3 (NOWOZIN, 2014).

$$IoU(x, y) = \frac{q_1 \cap o_1}{q_1 \cup o_1} \quad (2.3)$$

The sizes of the query and candidate regions influence the IoU value. For instance, if the query region occupies 10% of the candidate region with an area of 10 units, the maximum IoU value achievable is 0.1. This occurs when the entire query region is contained within the candidate region, resulting in an intersection area of 1 unit and a union area of 10 units.

## 2.3 Object Detection

Object Detection is a well-known technique in the field of Computer Vision that plays a crucial role in localizing objects within images or video. The fundamental concept involves creating bounding boxes that accurately encompass specific objects of interest. These bounding boxes could be utilized in conjunction with a classification algorithm to assign a probabilistic output, which provides information about the presence and type of the enclosed object.

Exhaustive search method is a commonly used approach in object detection. This method involves sliding windows of different sizes across an image to identify potential objects. However, the need to search through a large number of windows, even for relatively small-sized images, poses computational challenges. Although optimizations such as employing windows of varying proportions can improve efficiency to some extent, the overall effectiveness remains limited due to the sheer number of windows involved. To address this challenge, researchers have proposed innovative methods over the years that optimize the window application process for object localization.

Incorporating these advancements, the object detection techniques employed in this work offer improved performance in terms of both confidence and computational efficiency. These methods leverage innovative optimizations in the application of windows for locating objects in images, resulting in superior object detection capabilities. This ensures that the system can accurately and efficiently identify objects within the images, thereby supporting the overall objective of the pattern spotting system.

While the traditional approach for object detection involves using bounding boxes and classification algorithms, some methods focus solely on object localization without explicit classification. These methods leverage object proposals and keypoint detection techniques to locate objects in images precisely. Here are a few notable examples:

1. EdgeBoxes ([ZITNICK; DOLLÁR, 2014](#)): A fast object proposal method that generates potential object bounding boxes by exploiting the edges present in an image. It selects boxes based on their ability to fit objects, enabling efficient and accurate object localization tightly.
2. CornerNet ([LAW; DENG, 2020](#)): An object detection method that focuses on detecting object corners or keypoints instead of relying solely on bounding boxes. By detecting corners and regressing to complete bounding boxes, CornerNet achieves accurate and robust object localization.
3. Selective Search ([UIJLINGS et al., 2013](#)): An object proposal method that combines multiple low-level image cues, such as color, texture, and size, to generate a diverse set of potential object regions. These regions serve as candidate bounding boxes for object detection.

These methods, prioritizing precise object localization rather than explicit classification, have demonstrated their effectiveness in various applications. By leveraging the strengths of these techniques, researchers can develop object detection systems tailored to specific requirements, such as those involved in pattern spotting or fine-grained object recognition tasks.

### 2.3.1 EdgeBoxes

EdgeBoxes ([ZITNICK; DOLLÁR, 2014](#)) is a fast and efficient object proposal method that leverages edge information for generating potential object bounding boxes in an image. The method exploits the observation that objects in images are often associated with solid edges. By capitalizing on this edge-based cue, EdgeBoxes aims to generate accurate, tightly-fitting bounding boxes.

The algorithm starts by computing edge responses using an edge detection technique like structured forests. These edge responses capture the presence of edges at different orientations and scales within the image. Based on these edge responses, EdgeBoxes performs a comprehensive search for potential object regions by evaluating rectangular boxes of varying aspect ratios and scales. The algorithm assigns a score to each box based on its ability to tightly enclose object-like regions.

To efficiently explore the search space, EdgeBoxes adopts a hierarchical grouping strategy. It hierarchically groups similar boxes based on their location and appearance. This grouping process helps eliminate redundant and overlapping proposals, significantly reducing the number of candidate regions and improving computational efficiency.

After generating a set of potential bounding boxes, EdgeBoxes ranks them based on their objectness score. The objectness score reflects the likelihood of a box containing an object rather than background clutter. By selecting boxes with high objectness scores, EdgeBoxes provides a diverse set of high-quality proposals that cover a wide range of objects in the image.

The EdgeBoxes method offers several advantages. It operates in a computationally efficient manner, allowing real-time object detection applications. Additionally, it balances recall and efficiency well, generating a compact set of proposals while maintaining high recall rates. These characteristics make EdgeBoxes a valuable tool in object detection pipelines, facilitating subsequent steps such as object classification and localization (ZITNICK; DOLLÁR, 2014).

### 2.3.2 CornerNet

CornerNet (LAW; DENG, 2020) is an object detection method that takes a unique approach by focusing on detecting object corners or keypoints instead of using traditional bounding boxes. By detecting corners and regressing to complete bounding boxes, CornerNet achieves accurate and robust object localization.

The method consists of two main stages: corner keypoint detection and corner-based box regression. In the keypoint detection stage, CornerNet generates a heatmap for each corner keypoint, indicating the likelihood of a corner being present at each spatial location in the image. This heatmap is generated using a convolutional neural network (CNN) architecture trained to predict corner keypoints. To obtain accurate corner locations, CornerNet employs a keypoint grouping technique. It groups the corner keypoints based on their spatial proximity and assigns each group a unique instance ID. This grouping process helps refine the corner keypoint locations and improve their accuracy.

CornerNet regresses from the detected corner keypoints to complete bounding boxes in the box regression stage. It predicts the offsets between the corner keypoints and uses them to reconstruct the bounding box coordinates. By leveraging the corner keypoints, CornerNet achieves precise localization of objects in the image.

CornerNet introduces an innovative and efficient architecture called the CornerNet-Saccade. This architecture utilizes a two-stage cascaded network that progressively refines the corner keypoint locations. The first stage, the detection network, focuses on capturing global information to generate an initial set of corner keypoints. The refinement network's second stage operates at a higher resolution and refines the corner keypoint locations with finer details.

The CornerNet method has shown impressive performance in object detection tasks. It offers several advantages, including accurate localization and robustness to occlusion and scale variations. By explicitly detecting corner keypoints, CornerNet provides a more precise representation of object boundaries compared to traditional bounding box methods. This makes it particularly useful in scenarios where precise object localization is critical, such as fine-grained object recognition and pose estimation tasks (LAW; DENG, 2020).

### 2.3.3 Selective Search

Selective Search (SS) is a method introduced in 2012 (UIJLINGS et al., 2013) that combines the strengths of exhaustive search and segmentation for object detection. This approach aims to capture objects at various scales and orientations while diversifying the grouping of regions based on different metrics. Selective Search captures both these features, yet with some additional benefits:

1. Capturing all scales, addressing the fact that objects in images can present at different sizes and orientations. This step is based on the intuition of the hierarchical structure of images. The initial regions are created via a graph-based greedy algorithm, which starts grouping the most similar regions until the whole image becomes a unique region;
2. Diversifying the grouping of regions according to different metrics. The authors introduced four different diversification strategies based on color (C), texture (T), size (S), and fill (F). For each feature, a similarity score (between each couple of regions) is computed. The final similarity score is a linear combination of the above four similarity scores;

3. Finally, the method is computationally fast, compared with its predecessor exhaustive search (UIJLINGS et al., 2013).

As a first part of the method, a hierarchical grouping algorithm is used to form the basis of the SS. As the grouping process itself is hierarchical, it is naturally possible to generate locations at all scales, continuing the grouping process until the entire image becomes a single region. This satisfies the condition of capturing all scales. The method uses region-based features whenever possible because regions can provide richer information than pixels. To obtain a set of small initial regions that ideally do not span multiple objects, the fast method of Felzenszwalb and Huttenlocher (FELZENSZWALB; HUTTENLOCHER, 2004) is used.

From there, the grouping procedure works as follows. First, (FELZENSZWALB; HUTTENLOCHER, 2004) is used to create initial regions. Then, a greedy algorithm is used to group the regions iteratively: first, the similarities between all neighboring regions are calculated. The two most similar regions are grouped, and new similarities are calculated between the resulting region and its neighbors. The process of grouping the most similar regions is repeated until the entire image becomes a single region.

The first strategy is the use of Complementary Color Spaces, where different scenes and lighting conditions are taken into account. Therefore, the hierarchical clustering algorithm runs on a variety of color spaces with a variety of invariance properties. Specifically, the following color spaces with an increasing degree of invariance: (1) RGB, (2) the intensity (grey-scale image) I, (3) Lab, (4) the rg channels of normalized RGB plus intensity denoted as rgI, (5) HSV, (6) normalized RGB denoted as rgb, and finally (8) the Hue channel H from HSV.

The second strategy is the use of Complementary Similarity Measures, where four complementary and fast-to-calculate similarity measures are defined. These measures are all in the range  $[0,1]$  which facilitates combinations of these measures. The first measures color similarity where for each region, the histogram of each color channel present in the image is generated. Where 25 dimensions are used in the histogram of each color channel. Which generates 75 dimensions (25 for each R, G, and B), and all channels are combined into a vector ( $n = 75$ ) for each region. The second measures texture similarity where for each region, the texture measurement is calculated using 8 Gaussian derivations generated from the image as a SIFT. After that, histograms with 10 dimensions are extracted for each color channel. Which generates a 240-dimensional vector for each region. The third encourages small regions to merge early. Suppose this similarity is not taken into account. In that case, more significant regions will continue to merge with larger regions,

and proposed multi-scale regions will be generated only at this location. Finally, the fourth measures how well region  $r_i$  and  $r_j$  fit into each other. If two regions fill each other well (for instance, one region is present in the other) they should be merged, if two regions do not touch each other they should not be merged.

The final strategy is the use of Complementary Starting Regions, where a third diversification strategy is used to vary the complementary starting regions. The algorithm to image segmentation (FELZENSZWALB; HUTTENLOCHER, 2004) is used to generate these initial regions. As noted earlier, different initial regions are obtained by varying the color spaces, each with different invariance properties. In addition, the threshold parameter  $k$  in (FELZENSZWALB; HUTTENLOCHER, 2004) is also varied. Figure 2.2 shows two examples of the selective search applied in different scales.

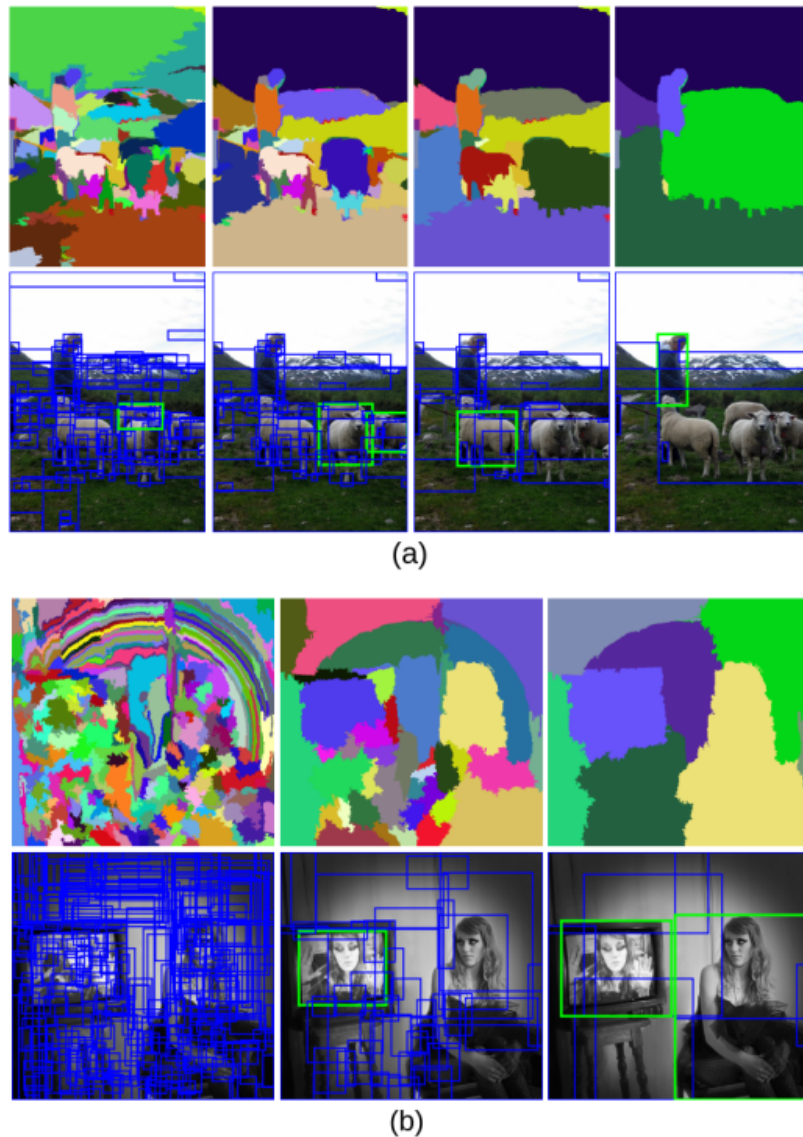


Figure 2.2: In this example, it is shown how the scale variation influences the final result.

Selective Search has been shown to be computationally fast compared to exhaustive search, making it a practical choice for object detection tasks. By capturing objects at multiple scales, diversifying region grouping, and utilizing complementary strategies, Selective Search provides an effective approach for localizing objects in images.

## 2.4 Feature Extraction

Feature extraction is the process of transforming raw data into numeric features that can be effectively processed while preserving the relevant information in the original dataset. It has proven to be more effective than directly applying machine learning algorithms to raw data (ALPAYDIN, 2014). The feature extraction process can be performed manually or automatically, depending on the context:

- Manual feature extraction involves identifying and describing the features that are relevant to a specific problem and implementing methods to extract those features. A good understanding of the domain or background knowledge can be valuable in making informed decisions about useful features. Over the years, engineers and scientists have developed handcrafted feature extraction methods for images, signals, and text. For example, averaging a window on a signal can be a simple handcrafted feature extraction method. Generally, methods using manual feature extraction are referred to as handcrafted features.
- Automated feature extraction utilizes specialized algorithms or deep networks to automatically extract features from signals or images without the need for human intervention. This technique is particularly useful when moving quickly from raw data to developing machine learning algorithms is needed. Wavelet scattering is an example of an automated feature extraction method. Features extracted automatically using deep networks are often referred to as deep features.

With the emergence of deep learning, feature extraction using deep networks has largely replaced traditional feature extraction methods, especially in the domain of image data. However, feature extraction remains a significant challenge for applications involving signals and time series, requiring domain knowledge and expertise to build effective predictive models.

### 2.4.1 Deep features

Deep features are characterized by the hierarchical response of neural networks, starting from the input layer, passing through multiple hidden layers, and ending at the output layer. In neural network-based approaches, the weights are learned using large training datasets. This learning process involves iteratively adjusting the neural network parameters and weights and requires a substantial amount of training data. The performance of neural networks is related by the strategy employed for gathering and selecting samples during the training process (KAMNITSAS et al., 2017).

The initial layers of convolutional neural networks (CNNs) compress the input image by extracting low-level features such as edges and curves, focusing more on local patterns. The hidden layers respond to and create their own feature filters for capturing more complex patterns in the input data, such as textures, shapes, or variations of previously processed features (GKELIOS et al., 2021).

Consequently, while a conventionally trained network may have downstream nodes capable of identifying specific features, such as faces, they might not be able to distinguish a face from similar objects. However, the response from deeper layers in the network hierarchy serves as a feature filter that the model can utilize not only to differentiate faces from non-facial objects but also to create new classifiers during classification tasks.

Deep neural networks have achieved remarkable success in high-dimensional feature extraction. With the introduction of convolutional neural networks for image processing, several influential deep network architectures have been proposed and demonstrated promising results. For instance, AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) consists of five convolutional layers followed by three fully connected layers. VGGNet (SIMONYAN; ZISSERMAN, 2014) increased model depth, resulting in improved image classification performance. Researchers discovered that the depth of representations plays a vital role in achieving high performance across various visual recognition tasks. However, the challenge of vanishing/exploding gradients made it difficult to construct very deep neural networks. ResNet (HE et al., 2016a) addressed this issue by employing residual learning to deepen the network and benefit from extremely deep models.

Deep features have significantly transformed the field of computer vision by revolutionizing the way we extract information from images. These features, obtained through the hierarchical response of neural networks, have proven to be highly effective in capturing rich and discriminative representations. This progress has led to the development of diverse applications, with deep hashing standing out as a prominent example.



### 2.4.2 Deep Hashing

When working with high-dimensional and large-scale data, the computational time required to accurately find the closest sample to a query can be substantial. To address this challenge, researchers have increasingly focused on approximate nearest-neighbor search, as it often satisfies search requirements while significantly reducing search complexity (LUO et al., 2021).

Hashing methods can be broadly classified into two main categories: local sensitive hashing and learning to hash. Local sensitive hashing aims to map original data onto multiple hash buckets, grouping objects with similar distances in the original space into the same bucket. However, to enhance retrieval accuracy, these methods often require the creation of numerous hash tables, limiting their applicability to large-scale datasets. As local sensitive hashing is data-independent, researchers have focused on learning hash functions to generate high-quality hash codes. Learning to hash has garnered significant interest in academic fields such as machine learning and data mining, leading to the development of pioneering methods like spectral hashing and semantic hashing (SALAKHUTDINOV; HINTON, 2009; WEISS; TORRALBA; FERGUS, 2008).

On the other hand, learning to hash involves optimizing the parameters of deep neural networks using extensive labeled datasets and loss functions designed explicitly for binary code learning. By harnessing the power of deep learning, learning to hash enables the creation of retrieval systems that are both effective and efficient. This approach has gained considerable attention in recent years due to its ability to address the challenges posed by traditional hashing methods while leveraging the rich representations learned by deep neural networks to enhance retrieval performance and scalability. Deep hashing offers several advantages for hash code generation in comparison to traditional methods.

Firstly, deep learning models, with their powerful representation capabilities, can learn highly complex hash functions that capture intricate patterns and relationships in the data. This enables the generation of more discriminative hash codes, enhancing the performance of tasks such as similarity search, clustering, and information retrieval.

Secondly, deep learning facilitates the end-to-end generation of hash codes, which proves to be highly advantageous in many applications. End-to-end deep hashing models directly take raw data as input and output hash codes without requiring manual feature engineering or intermediate steps. This streamlined approach eliminates the need for handcrafted features and simplifies the overall process of generating hash codes. Furthermore, deep hashing methods are particularly useful for large-scale datasets, as they offer scalable solutions that can handle the computational demands of high-dimensional data.

Deep hashing has found applications in various domains. For instance, in image retrieval tasks, deep hashing enables efficient similarity search in large image databases. By mapping images to compact binary codes, retrieving visually similar images within a reduced search space is feasible. Deep hashing techniques can efficiently index and search through a vast collection of documents based on their semantic similarity in document retrieval. This is particularly valuable in applications such as plagiarism detection, content recommendation, and information retrieval systems.

Furthermore, deep hashing has also been applied to video retrieval, where it allows for efficient searching and indexing of videos based on their visual content or semantic information. It enables tasks such as video summarization, action recognition, and video recommendation.

In summary, deep hashing techniques have emerged as a powerful solution for efficient and effective approximate nearest-neighbor search. By leveraging the representation capabilities of deep neural networks and enabling end-to-end hash code generation, deep hashing methods offer improved performance and scalability for tasks involving high-dimensional and large-scale data.

## 2.5 Similarity Measures and Distances

In pattern spotting or image retrieval tasks, the performance of a method relies heavily on the numerical measure used to quantify the similarity between two images: the query image and the candidate image present in the document. The similarity calculation often involves the representation of images using feature maps. This section provides an overview of several commonly used similarity and dissimilarity measures in the field of computer vision.

The term "distance" is frequently used when referring to dissimilarity calculations. In the context of this work, the distance is computed between feature maps extracted by a convolutional neural network (CNN) to perform a retrieval task. The choice of distance measure depends on the specific set of images and their representation. However, all dissimilarity functions adhere to certain criteria (TAN; STEINBACH; KUMAR, 2005), where  $d$  is the distance and  $x$ ,  $y$  and  $z$  are the points:

- $d(x, y) \geq 0$  for all  $x$  and  $y$ ;
- $d(x, y) = 0$  only if  $x = y$ ;
- $d(x, y) = d(y, x)$ , when the distance between two elements is equal;

- $d(x, y) + d(y, z) \geq d(x, z)$ , known as a triangular difference to  $x$ ,  $y$ , and  $z$  points.

In the task of retrieving images from historical documents, the image collection is organized as a set of feature maps, with  $c_j = a_{1,j}, a_{2,j}, \dots, a_{n,j}$  representing the feature map of document candidate  $j$ , and  $q_i = a_{1,i}, a_{2,i}, \dots, a_{n,i}$  representing the query vector with index  $i$ . Here,  $n$  corresponds to the number of features in each image. To obtain a ranking of results, the query must be compared against all candidate vectors in the collection, and the results must be ordered according to the chosen measure. The following are some of the main distance measures commonly found in the literature.

Minkowski distance is the generalization of the distance between two points in an  $n$ -dimensional characteristic space (TAN; STEINBACH; KUMAR, 2005). The distance is defined by Equation 2.4.

$$d(c_j, q_i) = \left( \sum_{k=1}^n |c_{kj} - q_{ki}|^r \right)^{1/r} \quad (2.4)$$

where  $k$  is the index of  $c_j$  and  $q_i$ , and the parameter  $r$  can represent different values and variations. The most famous are:

- $r = 1$ . Hamming distance is used between two binary feature maps, defined by Equation 2.5:

$$d(c_j, q_i) = \sum_{k=1}^n |c_{kj} - q_{ki}| \quad (2.5)$$

- Euclidean distance is a function that is traditionally used and corresponds to the Equation 2.6:

$$d(c_j, q_i) = \sqrt{\sum_{k=1}^n (c_{kj} - q_{ki})^2} \quad (2.6)$$

In Cosine similarity, the attributes are used as a vector to find the normalized dot product of a pair of bit vectors. Two vectors with the same direction have a similarity equal to 1, and two opposite vectors have a similarity equal to -1. The similarity is calculated by the Equation 2.7 (TAN; STEINBACH; KUMAR, 2005):

$$\text{Similarity}(c_j, q_i) = \frac{\vec{c}_j \cdot \vec{q}_i}{|\vec{c}_j| * |\vec{q}_i|} = \frac{\sum_{k=1}^n (c_{kj} * q_{ki})}{\sqrt{\sum_{k=1}^n c_{kj}^2} * \sqrt{\sum_{k=1}^n q_{ki}^2}} \quad (2.7)$$

where  $\vec{c}_j \cdot \vec{q}_i$  is the dot product between the vectors  $c_j$  and  $q_i$ , in this way, the distance between two correlated vectors is presented in Equation 2.8.

$$d_{\cosine}(c_j, q_i) = 1 - \cos^{-1}(\text{Similarity}(c_j, q_i)) \quad (2.8)$$

The choice of similarity measure depends on the features used in the image representation. Euclidean distance and cosine similarity are commonly used for cluster analysis (EN et al., 2016b), while Hamming distance is appropriate for binary information (Riba et al., 2017).

A comparison of different similarity measures, including Euclidean distance, and cosine similarity, was performed in image retrieval approaches by (MALIK; BAHARUDIN, 2013). They concluded that Euclidean distance provides good precision. However, both Euclidean distance and cosine similarity yield good results, as presented by (EN et al., 2016b; Riba et al., 2017).

All these measures can be used in the context of historical document images. However, if the comparison needs to be applied to more complex data samples with features of different dimensionality and types that may require compression before processing, using these measures alone would be inadequate. In such cases, a Siamese Neural Network (SNN) may be the best choice (CHICCO, 2021). An SNN is a neural network architecture that combines two identical networks with the same configuration, parameters, and weights. Two images are used as input to the network, and the output generated by the SNN execution can be considered the semantic similarity between the projected representations of these two input images (CHICCO, 2021).

## 2.6 Final Considerations

This chapter provides a comprehensive literature review focusing on content-based image retrieval (CBIR) systems. Each step of the CBIR process, including candidate generation, feature extraction using CNN models, and computation of similarity measures, is discussed in detail. The subsequent chapter will delve into the specifically related works that have served as a foundation for the current study.

# Chapter 3

## Related Works

In this work, the focus lies on the application of Content-Based Image Retrieval (CBIR) methods in historical document images. The term Content-Based Image Retrieval was first introduced in 1992 when experiments conducted by T. Kato aimed to automatically retrieve images from a collection ([HUNEITI; DAOUD, 2015](#)). The primary goal of CBIR methods is to retrieve information without relying on prior contextual knowledge. Thus, the application needs to assess variations in color, texture, shape, and location of the query within the image collection ([EAKINS et al., 1999](#)). The query itself can be presented as either a complete image or specific regions of interest within the image.

With the continuous advancements in data storage and image acquisition technologies, the volume of image collections has witnessed a remarkable surge. As a result, the need for efficient Content-Based Image Retrieval (CBIR) tools has become increasingly crucial in order to effectively manage and navigate through these extensive collections ([TORRES; FALCÃO, 2006](#)). CBIR encompasses a wide range of techniques, tools, and algorithms that draw upon diverse fields such as statistics, pattern recognition, signal processing, and computer vision. The underlying goal of CBIR is to enable users to retrieve images based on their visual content. To achieve this, CBIR techniques typically involve two primary steps: image feature extraction and query processing.

In the initial step, an image is analyzed, and relevant features such as color, texture, shape, and others are extracted ([HEBBAR; NIRANJAN; MUSHIGERI, 2013](#)). Numerous features can be employed based on the specific objectives of the technique, and there are various approaches to combining these features to enhance the retrieval outcomes. The subsequent step involves query processing, which entails comparing the extracted features from the query image with those of the images in the collection. A metric, such as the Euclidean distance, is often employed to determine the closest matches to the query. Smaller distances between images indicate higher similarity between them.

When considering the application of CBIR in historical documents, the image collection can be likened to a compilation of book pages or a collection of books. At the same time, the query corresponds to a graphic element that may or may not be present within these pages. Given that this is a relatively recent challenge in comparison to other computer vision problems, only a limited number of approaches have been proposed to address the issue of Pattern Spotting (PS) and Image Retrieval (IR) in historical documents, particularly within the context of the DocExplore database (EN et al., 2016a).

A comprehensive system for image search and localization of small graphic objects in medieval documents was proposed by Sovann et al. (EN et al., 2016b). The system is based on extracting and indexing regions of interest within the images, representing these regions using handcrafted descriptors, and employing compression and approximation techniques to search for similarity between the query and image candidates.

The system consists of two main processing phases: the offline phase and the online phase. The offline phase begins with a background filtering process to eliminate non-informative areas that do not represent graphic elements. Subsequently, the informative zones undergo an analysis procedure for each filtered image using sliding windows. Within each subwindow, a descriptor is extracted using one of the Bag-of-Visual-Words (BoVW), Locally Aggregated Descriptor (VLAD), or Fisher Vectors (FV) representations. To address memory consumption, the concept of product quantization (PQ) is utilized to efficiently compress the vectors, with the later use of asymmetric distance computation (ADC). However, directly applying PQ to the BoVW model may result in losses due to the sparseness of the resource vectors. To mitigate this issue, Latent Semantic Analysis (LSA) is employed to transform the BoVW space into a compressed low-rank topic space. For VLAD and FV, Principal Component Analysis (PCA) is employed to project the representation into a lower-dimensional feature space.

Additionally, to expedite the search in the online phase, the inverted file structure (IVF) is utilized to avoid exhaustive searches in the subwindows. Finally, a similarity measure determines the most suitable indexed subwindows to be returned. An overview of the system can be observed in Figure 3.1, with the steps of the offline and online procedure.

While this system has exhibited good performance on medieval document images within the DocExplore database (EN et al., 2016a), it does possess certain limitations that render it less suitable for other types of document images. For instance, it is sensitive to variations in size, shape, color, and patterns that must be detected. Furthermore, the system lacks scaling support and necessitates post-processing to accurately locate objects in regions of interest using traditional correlation methods.

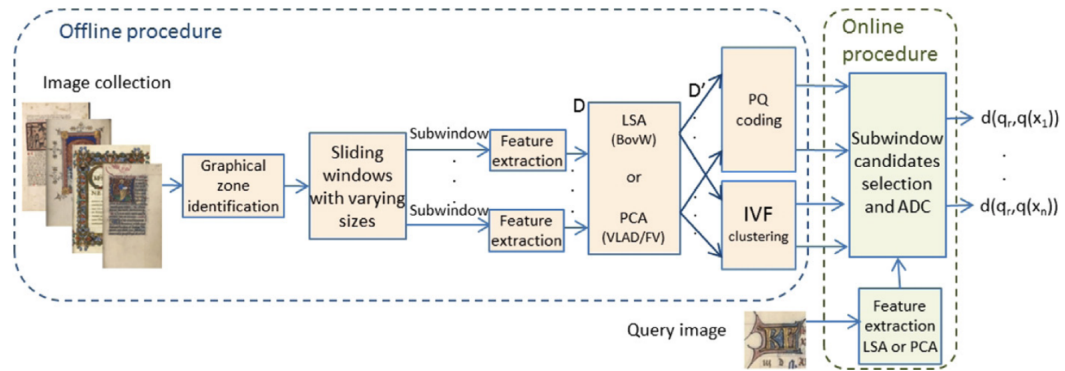


Figure 3.1: Overview of the pattern spotting system proposed in (EN et al., 2016b).

In a different study, a method proposed by Ignacio et al. (ÚBEDA et al., 2019) introduced an approach that utilizes convolutional neural networks (CNN) based on feature pyramid networks (FPN) as the feature extractor for the system. Similar to the method presented by Sovann et al. (EN et al., 2016b), this approach comprises an offline and an online phase.

In the offline phase, the primary focus was on processing and indexing the collection of historical documents, where subsequent searches would be conducted. The processing step involved utilizing the FPN to extract descriptors from localized regions of the documents, allowing for indexing at various scales with a single pass through the network. Pre-processing techniques were employed, such as background removal from the images within the DocExplore database (EN et al., 2016a) and image centering on a black background canvas.

Moving on to the online phase, the query image was processed, and its features were extracted and normalized in a manner similar to the offline phase. The subsequent steps of this phase were centered around processing and locating multiple occurrences of objects similar to the query image within the indexed document collection. An overview of the pipeline can be observed in Figure 3.2.

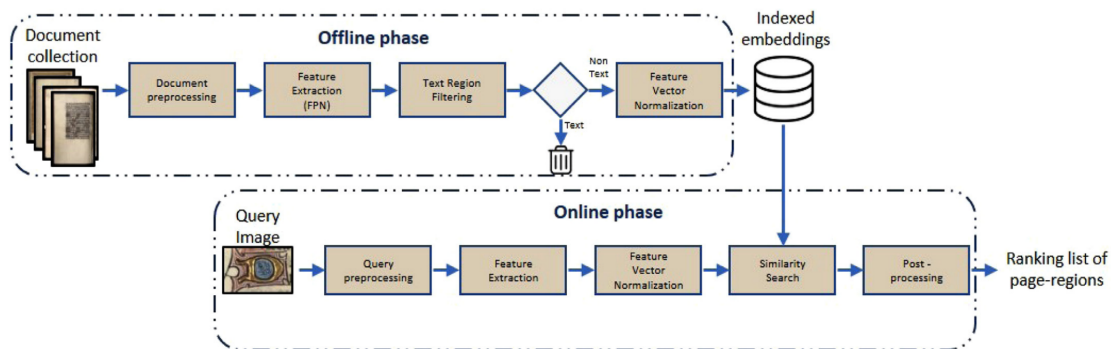


Figure 3.2: Overview of the pattern spotting system proposed in (ÚBEDA et al., 2019).

Although this approach demonstrated reduced memory requirements and processing time compared to the system proposed by Sovann et al. (EN et al., 2016b), it does have certain limitations. Firstly, in terms of retrieval tasks, it yielded a 13% lower mean Average Precision (mAP) compared to the results obtained by Sovann et al. (EN et al., 2016b). Furthermore, the FPN exhibited sensitivity to the shape of the object being searched within the document collection, achieving good precision only for objects with a square format.

In a separate study, Kelly et al. (WIGGERS et al., 2019) proposed an approach that also incorporates offline and online phases, utilizing convolutional neural networks (CNNs) in a Siamese Neural Network (SNN) framework. Transfer learning and fine-tuning techniques were employed, leveraging a pre-trained CNN model. The selective search algorithm (SS) proposed by Uijlings et al. (UIJLINGS et al., 2013) was also utilized in this approach.

In the offline phase, Selective Search (SS) was applied as a pre-processing step to generate candidate regions within the image collection. The goal was to capture all possible object locations within the images of historical documents, regardless of their sizes, shapes, and colors. As a result, candidate regions corresponding to the detected objects were saved.

A pre-trained SNN based on the AlexNet model proposed by Krizhevsky et al. (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) was employed to measure the similarity between the query image and the candidate regions. The SNN was trained using pairs of positive and negative images from the Imagenet dataset, enabling the network to learn to distinguish between similar and dissimilar objects. The Euclidean distance was used to calculate the similarity between the query and the candidate regions. The AlexNet model was further fine-tuned using images from the context of historical documents.

In the online phase, an interaction was performed for each candidate region returned by the SS. The image of the candidate region object and the query image was fed into the SNN, which produced a similarity value as output. A ranking was generated after calculating the similarity between the query and all candidate objects. A post-processing step was then carried out to merge candidate regions that had intersections with each other. An overview of the system can be observed in Figure 3.3.

While this approach achieved improvements over the results obtained by Sovann et al. (EN et al., 2016b) and Ignacio et al. (ÚBEDA et al., 2019) in the pattern spotting (PS) task, it does have a significant drawback. The excessive number of candidate regions that do not represent objects poses a challenge. As the query needs to be compared with each of these regions, the search process becomes time-consuming.



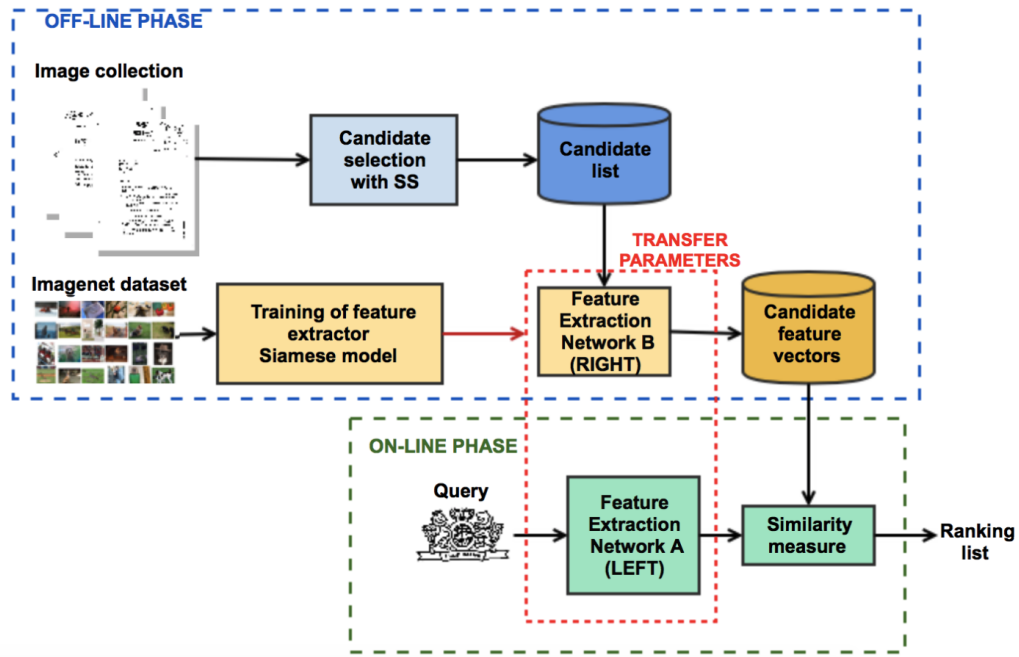


Figure 3.3: Overview of the pattern spotting system proposed in (WIGGERS et al., 2019).

### 3.1 Final Considerations

This chapter has provided an overview of relevant works that have applied Content-Based Image Retrieval (CBIR) methods in the domain of historical documents, particularly using the DocExplore database (EN et al., 2016a), which is also the dataset used in the experiments of this study. Building upon the insights gained from these works, the next chapter (Chapter 4) will present the proposed method, outlining a new and efficient system for Image Retrieval (IR) and Pattern Spotting (PS) in historical documents. The method aims to address the limitations of existing approaches and offer improved performance in terms of accuracy and efficiency.

# Chapter 4

## Proposed Method

The proposed method represents a significant advancement in the field of PS and IR in historical documents, effectively addressing various limitations observed in previous approaches. One of the primary areas for improvement of prior methods is their high computational complexity, resulting in lengthy search times for a query. Additionally, these methods often require substantial storage space to store feature maps, further impeding their efficiency. Moreover, achieving scalability with previous methods proves to be a challenging task.

To overcome these limitations, the proposed method introduces several key enhancements. Firstly, it incorporates the selective search algorithm, as illustrated in Figure 4.1, to detect object candidates within the document images effectively. This algorithm enables a more focused and precise analysis, enhancing the accuracy of subsequent steps. Furthermore, deep learning models are employed for feature extraction, leveraging their ability to capture intricate patterns and representations in the data. These models can produce either real-valued or binary code representations, depending on the application's specific requirements.

In the final stage of the proposed method, candidate images are ranked based on their feature similarity with the given input query. This similarity calculation enables the system to identify and present the most relevant and closely related images to the user. The method significantly reduces search times and enhances the overall retrieval performance by employing efficient feature comparison techniques.

The proposed method offers a more streamlined and efficient approach to PS and IR in historical documents. Integrating selective search, deep learning models, and feature ranking addresses previous shortcomings and provides an effective solution for retrieving and spotting patterns in historical document collections.

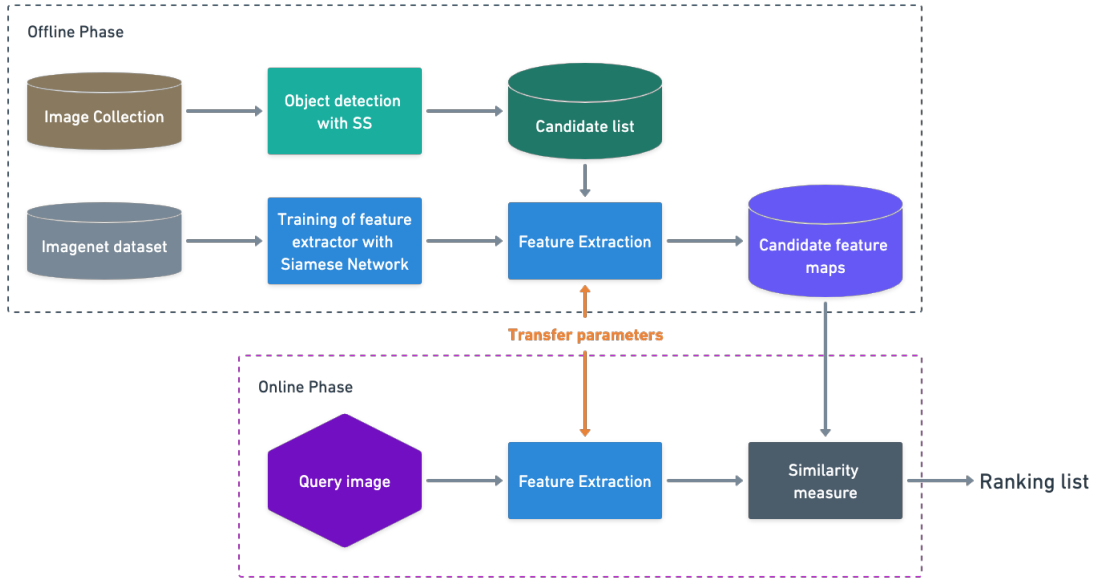


Figure 4.1: Overview of the proposed method with selective search and siamese networks.

## 4.1 Object Detection with Selective Search

The method utilizes the Selective Search (SS) algorithm (UIJLINGS et al., 2013) for object recognition in the offline phase. However, when applied to historical document images, this approach encounters specific difficulties, resulting in the detection of numerous invalid regions, such as antiquity stains, ink smudges, and page edges, which do not represent objects of interest. To address this issue, a modified version of the SS algorithm was employed, considering only one scale variation using the Felzenszwalb and Huttenlocher algorithm (FELZENSZWALB; HUTTENLOCHER, 2004) and utilizing the CTSF (Color, Texture, Size, and Fill) combination for measuring region similarity. While this modification significantly reduces the number of candidate regions compared to the approach in (WIGGERS et al., 2019), it still generates a considerable number of invalid regions. Consequently, a post-processing algorithm was developed to filter out these invalid regions.

The purpose of the post-processing algorithm is to exclude invalid regions, and regions that are either too small or too large along the  $x$  and  $y$  axes. To filter regions based on texture, an edge detection filter was applied to highlight the edges within the image. Based on Gaussian derivatives, this filter calculates gradient intensities and reduces noise effects. The potential edges are then reduced to 1-pixel curves by removing non-maximum pixels from the gradient magnitude. Subsequently, a hysteresis thresholding technique is applied to retain or discard edge pixels based on the magnitude of the gradient. The result is a binary image with a value of 1 representing the object edges.

---

**Algorithm 1:** Invalid candidate region filter
 

---

**Input** : A image  $Img$  of dimension  $h \times w$  of the candidate region and a threshold  $\alpha$   
**Output** : Is the candidate region valid

```

[1] binaryImage  $\leftarrow$  GetBinaryImage( $Img[h, w]$ );
[2] if Mean(binaryImage)  $<$   $\alpha$  then
[3]   return false; // candidate region is invalid

[4] /* generate eight sections from binary image */
[5] sectors  $\leftarrow$  GetSectors(binaryImage, 8);
[6] sectorsInvalid  $\leftarrow$  0;
[7] for sector in sectors do
[8]   if Mean(sector)  $<$   $\alpha$  then
[9]     sectorsInvalid  $\leftarrow$  sectorsInvalid + 1;
[10]  if sectorsInvalid  $>$  4 then
[11]    return false; // candidate region is invalid

[12] return true; // candidate region is valid

```

---

The binary images undergo specific manipulations. First, the mean pixel value of the image is computed. Next, a minimum average threshold, denoted as  $\alpha$ , was established empirically. If the proportion of edge pixels to the total number of pixels falls below  $\alpha$ , the image is immediately excluded. However, some invalid regions may go undetected using only the mean threshold. To address this, the image is segmented into eight sectors, and the average is computed for each sector. If more than 50% of the sectors have an average value lower than  $\alpha$ , the image is excluded. The pseudo-code for the invalid candidate region filter is shown in Algorithm 1. This filter reduces the number of candidate regions the SS algorithm returns by up to 1/5, without requiring any training on the context images. The training was avoided to ensure the generalization of the proposed method when applied to other image databases. Figure 4.2 provides examples of invalid regions that have been successfully filtered out.



Figure 4.2: Examples of regions filtered out by the invalid candidate region algorithm.

Finally, features are extracted from the remaining valid candidate regions and stored for use in the online phase. In the following section, two CNN architectures are evaluated as base feature extractors, and their performance is discussed in detail.

## 4.2 Feature Extraction using Deep Learning

The proposed method utilizes a Siamese neural network (SNN) to compute the similarity between the query and image candidates. As an initial approach, two convolutional neural networks (CNN) architectures were evaluated as alternatives for composing the SNN: VGG19 (SIMONYAN; ZISSERMAN, 2014) and ResNet50V2 (HE et al., 2016b). These architectures were chosen for their well-established feature extraction capabilities and their convenient availability within the deep learning framework used for the experiments.

VGG19 is a CNN architecture composed of 19 layers that have been specifically designed for large-scale image classification tasks. It is structured with five convolutional blocks, each followed by a max-pooling layer. The experiments conducted in this study revealed the potential of utilizing VGG19 as a feature extractor. By extracting the outputs from each block and concatenating them to create a feature map, it was observed that specific color and shape characteristics were effectively emphasized in the final outcome. Additionally, the exploration of combining pairs of blocks was carried out to optimize the feature extraction process, as elaborated in Chapter 5.

ResNet50V2 belongs to the family of residual deep networks, which are known for their remarkable depth and high precision (HE et al., 2016a). The philosophy of VGG networks inspires this architecture (SIMONYAN; ZISSERMAN, 2014) and employs  $3 \times 3$  filters in its convolutional layers. The design principles of ResNet50V2 ensure that the number of filters remains consistent for the same output feature map size and is doubled if the feature map size is halved. ResNet50V2 comprises 50 convolutional layers, along with max-pooling and average-pooling layers.

To leverage transfer learning, both VGG19 and ResNet50V2 were pre-trained on the supervised ImageNet dataset (RUSSAKOVSKY et al., 2015), which consists of 1.28 million training samples and 50 thousand validation samples distributed across 1,000 classes, encompassing various contexts and objects. Multiple SNN models were constructed to comprehensively compare the two architectures and explore variations within each architecture. In the subsequent sections, the performance and effectiveness of these CNN architectures and the variations introduced will be thoroughly examined and analyzed. This evaluation will provide insights into the suitability of these architectures.

To train the Siamese neural networks (SNNs), pairs of images were generated from the ImageNet dataset. The images were resized to a dimension of  $224 \times 224$  pixels using bilinear interpolation, as depicted in Figure 4.3. In total, 250,000 image pairs were generated, consisting of 150,000 negative pairs and 100,000 positive pairs. This ratio of  $1.5 \times$  more negative pairs than positive pairs follows the approach proposed by (MELEKHOV; KANNALA; RAHTU, 2016).

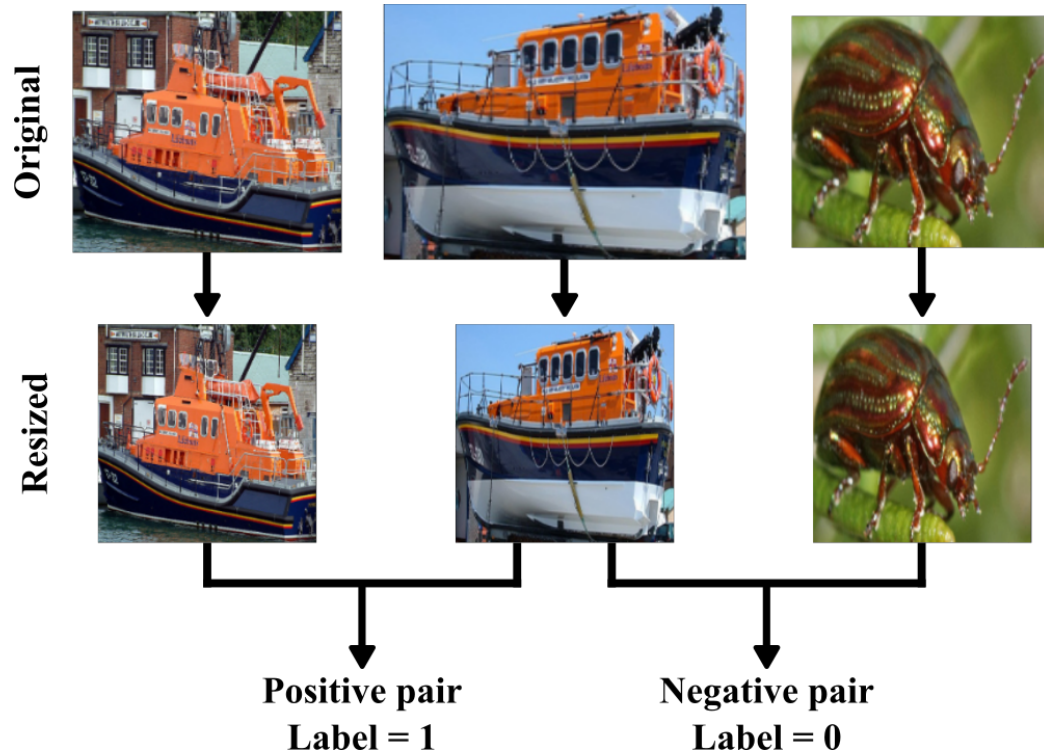


Figure 4.3: Example of data preparation for the training set created using the Imagenet dataset.

Each pair of images was used as input for the SNNs during the experiment. The resulting feature maps were then utilized to calculate the Euclidean distance, measuring the similarity between the images. A dense layer with a sigmoid activation function was employed to ensure a normalized distance measurement, as illustrated in Figure 4.4. This step enhances the interpretability and comparability of the similarity scores.

The contrastive loss function, as proposed in (HADSELL; CHOPRA; LECUN, 2006), was chosen as the loss function for training the SNNs. This particular loss function was selected because it facilitates the learning process by encouraging the SNNs to minimize the distance between positive pairs and maximize the distance between negative pairs. The optimization of the SNNs using contrastive loss aims to improve the network’s capability to discern between relevant and irrelevant image pairs, thereby enhancing its discriminative power.

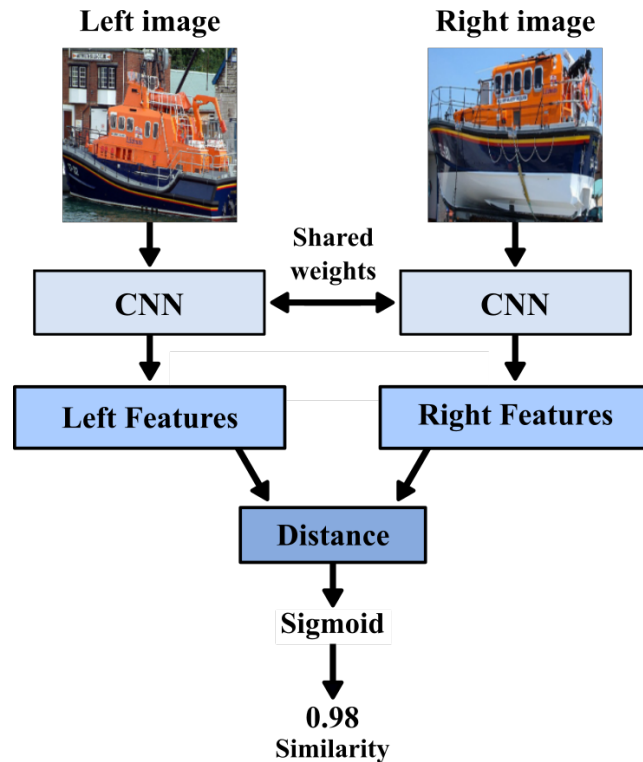


Figure 4.4: Structure example of a Siamese Neural Network.

Two variants of the ResNet50V2 network model were constructed for feature extraction. After applying the flattening operation to the convolution filters, the first variant utilizes the last convolutional layer, resulting in 100,352 features. This variant is referred to as *ResNet Conv*. The second variant utilizes the global average pooling layer, which yields 2,048 features. This variant is referred to as *ResNet GAPool*. By employing different layers for feature extraction, the capabilities of the ResNet50V2 architecture can be explored, and the impact of feature dimensionality on the performance of the proposed PS and IR system can be assessed.

Four variants of the VGG19 network architecture were developed to facilitate feature extraction. As previously mentioned, the VGG19 architecture's block outputs exhibit high sensitivity to color and texture variations, making them well-suited for feature representation. To determine which blocks yield superior feature extraction performance, models were created by concatenating all blocks as well as pairs of these blocks. The first variant, named *VGG19 Blocks*, incorporates all blocks and produces 1,472 features. The second variant, *VGG19 Block4-5*, combines blocks 4 and 5 to generate 1,024 features. The third variant, *VGG19 Block2-3*, merges blocks 2 and 3, resulting in 384 features. Lastly, the fourth variant, *VGG19 Block2-5*, combines blocks 2 and 5, producing 640 features. These variants allow for the evaluation of different feature extraction configurations, enabling the identification of the most effective combination of blocks for system.

Once the models were created, they were utilized to build the Siamese neural networks (SNNs) and trained using pairs of positive and negative images. Following the training process, the six models were employed to extract features from all candidate images, and the resulting feature values were stored in separate files. Up to this point, the features were represented using 32-bit floating-point values, which incurred significant storage costs.

To address this storage issue, deep hashing techniques were employed. The primary objective of deep hashing is to transform the feature maps, initially represented by floating-point values, into a more compact feature map using a binary code representation. By employing deep hashing, the storage requirements for the feature maps are significantly reduced, thereby alleviating the storage burden.

### 4.3 Feature Extraction using Deep Hashing

Deep hashing is a technique that enhances computation and storage efficiency in information retrieval systems. Its goal is to convert high-dimensional original features into compact hash codes, ensuring similar objects have similar hash codes while dissimilar objects have diverse hash codes. This involves mapping the original feature space to a Hamming space, resulting in binary hash codes composed of 0s and 1s, which are highly efficient for computation and storage in the binary form (LUO et al., 2021).

In this solution, the deep learning models that were previously developed were enhanced to incorporate deep hashing as an additional step. To enable the conversion of floating-point feature values into binary values, a new layer was introduced at the end of each network. This layer played a crucial role in discretization, mapping each element in the continuous interval to a binary value. During the construction of this layer, a margin value of 1 was considered, ensuring an effective discretization procedure.

By incorporating the discretization layer into the Siamese neural networks (SNNs) extractors, the models were re-trained using pairs of negative and positive images following the same process previously explained. The layer weights of the models served as initial weights, allowing for a seamless transition. Additionally, all layers of the models underwent fine-tuning to optimize their performance with the updated output. This comprehensive re-training ensured that the models were equipped with the knowledge to generate binary hash codes from the input feature vectors effectively.

However, instead of using traditional similarity calculations like Euclidean distance, the Hamming distance was employed as the similarity measure for the hash codes. This enables efficient comparison and evaluation of similarity between binary hash codes.



After the training phase, the six models with deep hashing capabilities were used to extract features from the candidate images, marking the completion of the offline phase. These hash-based features provide a compact representation of the images, facilitating efficient storage and computation during the subsequent online phase of the approach.

## 4.4 Similarity Calculations

The system is ready to transition into the online phase after completing the offline phase. This phase involves extracting features from the query and comparing them with the entire list of candidates generated during the offline phase. For features represented by floating-point values, the comparison is performed using the Euclidean distance. On the other hand, binary code features are compared using the Hamming distance, which is calculated as the sum of the absolute differences between each corresponding feature. Additionally, the XOR operation can be applied when dealing with binary or single-bit values. In this operation, equal values yield 1, while different values yield 0. The processor can perform the XOR operation more efficiently, as it is a native bitwise operation. After comparing all elements, the results are summed, as indicated in Equation (4.1).

$$d = \sum_{i=1}^n (q_i \text{ XOR } p_i) \quad (4.1)$$

After the calculation, the results for each query are sorted based on both the Euclidean and Hamming distances. Subsequently, lists of the top  $n$  candidates are generated to evaluate the models further.

It has been observed by En *et al.* (EN *et al.*, 2016b) and Wiggers *et al.* (WIGGERS *et al.*, 2019) that multiple candidate images often only partially cover the query or overlap with each other, which can hinder system performance. To address this issue, a post-processing step is proposed, wherein a union of the selected candidate images is performed to identify rectangular regions that can improve the effectiveness of the Pattern Spotting task. To implement this, the first  $\beta$  candidates are selected, and the union step is applied, considering an Intersection over Union (IoU) threshold of  $\gamma$ . If two images have an IoU measurement greater than  $\gamma$ , the image with the smaller distance is retained, while the other image is discarded. After the union process, the result is returned to the evaluation system for further analysis.

## 4.5 Final Considerations

In this chapter, the methods proposed to improve the results obtained in the study by (WIGGERS et al., 2019) have been outlined. Deep hashing techniques were developed to reduce processing time and storage space. The subsequent chapter (Chapter 5) will present the experimental results derived from the application of these methods. These results will offer valuable insights into the effectiveness and performance enhancements achieved through the utilization of deep hashing within the context of this study.

# Chapter 5

## Experimental Results

The Experiments section is structured as follows: Section 5.1 provides an overview of the experimental protocol adopted for the work. Section 5.2 presents the results obtained and the improvements achieved through the utilization of Selective Search. Subsequently, Section 5.3 discusses the results obtained for the IR task, while Section 5.4 delves into the results of the PS task. Finally, the last section presents the findings related to the processing time and storage costs of the proposed method.

### 5.1 Experimental Protocol

This section outlines the experimental protocol followed for training the SNN models. The training dataset comprised 250,000 pairs of images derived from the ImageNet database. A holdout strategy was employed to ensure a robust evaluation, a holdout strategy was used, allocating 70% of the data for training and the remaining 30% for validation.

The training set comprised 105,000 negative pairs and 70,000 positive pairs, while the validation set consisted of 45,000 negative pairs and 30,000 positive pairs. This division ensured a diverse representation of negative and positive pairs in the training and validation sets, allowing for effective model training and performance evaluation.

All models were trained using the same holdout strategy, with a total of 25 epochs. The selection of this epoch count was determined through iterative experimentation, considering the need to balance model convergence and computational efficiency. The objective was to train the models consistently and for an adequate number of epochs to capture optimal weights and ensure robust performance across all evaluated models in the subsequent experiments.

### 5.1.1 DocExplore

The experiments utilized the DocExplore database (EN et al., 2016a), a specialized resource for historical document analysis. The database consists of 1,500 images of historical documents from the 10th to the 16th century, as depicted in Figure 5.1. The Municipal Library of Rouen, France, provided the original documents, which were subjected to high-resolution scanning at 600 dpi. The resulting images had varying dimensions between 3000 and 4000 pixels.

The images were compressed to optimize computational resources and storage requirements, limiting the maximum size to 1024 pixels in each dimension and reducing the resolution to 72 dpi. This compression approach maintained the images' suitability for analysis while effectively mitigating computational burdens and storage demands.

The DocExplore database includes 1,447 unique queries, encompassing various document sizes. The query images exhibit diverse dimensions, ranging from  $20 \times 11$  pixels to  $1307 \times 319$  pixels, effectively representing the various characteristics encountered in historical documents. These queries constitute an extensive and comprehensive set of test cases, enabling evaluating of the proposed methods' performance.



Figure 5.1: Samples of historical document pages available in DocExplore.

## 5.2 Selective Search

The application of the selective search algorithm in the DocExplore database involved utilizing diversification strategies that combined color, texture, size, and fill elements. This approach resulted in the generation of 976,486 candidate regions of objects. Through initial experimentation, a value of 0.06 was established for the  $\alpha$  parameter of the invalid region filter function, removing invalid regions and subsequently reducing the number of candidate regions to 786,718. This reduction amounted to a decrease of 19.4%.

In contrast, the algorithm employed by Wiggers *et al.* (WIGGERS *et al.*, 2019) produced a significantly higher number of candidate regions, with a total of 36,159,870, this is approximately 46 times larger than the number obtained in this work. It is worth noting that such a high number of candidate regions not only imposes more significant storage costs but also increases processing time. This is due to the need to compare the query against a substantial number of candidates. This approach effectively mitigates the challenges associated with storage and processing resources by achieving a notable reduction in the number of candidate regions.

## 5.3 Image Retrieval Task

The experimental results for the image retrieval (IR) task, utilizing the ResNet50V2 and VGG19 architectures applied to the set of candidate images returned by selective search (SS), are presented in Table 5.1. In addition to ResNet50V2 and VGG19, the performance of the AlexNet network, as used by Wiggers *et al.* (WIGGERS *et al.*, 2019), was also evaluated. The AlexNet architecture underwent training following the same steps described in Wiggers *et al.* (WIGGERS *et al.*, 2019) and was applied to the same set of candidate images as ResNet50V2 and VGG19. For the IR task, the evaluation was conducted on the top 100, 300, 500, 700, and 1000 best results.

Among the tested architectures, VGG19 Block4-5 achieved the best result in all the rankings, with a mean average precision (mAP) of 53.21% in top 1000. This performance surpassed that of AlexNet by 10.4 percentage points and exceeded the result of Wiggers *et al.* (WIGGERS *et al.*, 2019) by 14.6 percentage points. Notably, Wiggers *et al.* (WIGGERS *et al.*, 2019) utilized selective search with a significantly more significant number of candidate images, totaling 36,159,870. The results indicate that employing a filtered selective search approach, which returns a reduced number of candidates, can positively impact the image retrieval outcome. This finding underscores these proposed methods' effectiveness in enhancing the IR task's performance.

Table 5.1: Image Retrieval results

mAP for Image Retrieval					
Method	Top n				
	100	300	500	700	1000
VGG19 Block4-5	0.4313	0.5058	0.5247	0.5307	<b>0.5321</b>
VGG19 Block2-5	0.4227	0.4939	0.5153	0.5217	0.5233
ResNet GAPool	0.4058	0.4617	0.4797	0.4880	0.4928
ResNet Conv	0.3956	0.4395	0.4567	0.4659	0.4723
VGG19 Block2-3	0.3303	0.4060	0.4276	0.4346	0.4355
VGG19 Blocks	0.3293	0.3934	0.4193	0.4279	0.4291
AlexNet used in (WIGGERS et al., 2019)	0.3168	0.3996	0.4220	0.4271	0.4282

To evaluate the performance of the models incorporating deep hashing, the two top-performing networks from each of the ResNet50V2 and VGG19 architectures were selected. The experimental protocol remained consistent, with the only modification being the adoption of the Hamming distance as the similarity calculation. Table 5.2 presents the results obtained by this deep hashing (H) networks.

Table 5.2: Image Retrieval results with Hashing

mAP for Image Retrieval					
Method	Top n				
	100	300	500	700	1000
VGG19 Block4-5 H	0.3952	0.4564	0.4744	0.4822	<b>0.4862</b>
VGG19 Block2-5 H	0.3302	0.3777	0.3939	0.4026	0.4090
ResNet GAPool H	0.2945	0.3437	0.3632	0.3727	0.3784
ResNet Conv H	0.0754	0.0854	0.1012	0.1169	0.1257

The results highlight the impact of deep hashing on the image retrieval task. Comparing the performance of the deep hashing models to the models using floating-point features, a minor reduction in mean average precision (mAP) was observed. However, this reduction is outweighed by the benefits of deep hashing in terms of efficiency and scalability, making it a highly valuable technique for image retrieval.

The results in Table 5.2 demonstrate the performance achieved by the networks utilizing deep hashing. When comparing the experimental results with state-of-the-art approaches, it becomes apparent that the method proposed by En *et al.* (EN et al., 2016b) outperforms both the VGG19 Block4-5 and VGG19 Block4-5 Hashing methods by 4.8 and 9.4 percentage points, respectively. An important observation is that the

methods proposed in this work offer the advantage of not depending on any information from the DocExplore database to refine their results. This characteristic ensures that the models presented are independent and can be applied to diverse datasets.

A comprehensive overview of the main results achieved by state-of-the-art approaches and the best results obtained in this work can be seen in Table 5.3. This comparison clearly demonstrates the competitiveness and effectiveness of the proposed models in image retrieval. Despite being outperformed by the approach proposed by En *et al.* (EN *et al.*, 2016b), the methods in this work still exhibit good performance, thereby highlighting their potential for practical applications in various domains.

Table 5.3: Comparison of the methods with the state-of-the-art of IR.

Methods	IR Top 1000
En <i>et al.</i> (EN <i>et al.</i> , 2016b)	<b>0.580</b>
VGG19 Block4-5	0.532
VGG19 Block4-5 H	0.486
Wiggers <i>et al.</i> (WIGGERS <i>et al.</i> , 2019) PP	0.386
Ubeda <i>et al.</i> (ÚBEDA <i>et al.</i> , 2019) PP	0.386
Ubeda <i>et al.</i> (ÚBEDA <i>et al.</i> , 2019) ES	0.286

## 5.4 Pattern Spotting Task

In the pattern spotting (PS) task, the mean average precision (mAP) was evaluated considering an intersection over union (IoU) threshold of  $\geq 0.5$ . The evaluation used the top 100, 300, 500, 700, and 1000 best similarity results. This allowed for a comprehensive assessment of the performance of the PS methods across varying levels of result granularity. The detailed results of the pattern spotting task applied in all models are presented in Table 5.4.

Similar to the image retrieval task, the evaluation of the models using deep hashing in the pattern spotting (PS) task involved selecting the top-performing networks from the ResNet50V2 and VGG19 architectures. Table 5.5 presents the results obtained when applying deep hashing (H) to these networks. Contrary to initial expectations, the VGG19 Block4-5 network demonstrated superior performance in the binary code representation compared to the floating-point values. This unexpected outcome highlights the effectiveness of the deep hashing technique in enhancing the performance of the VGG19 Block4-5 network specifically for the PS task.

Table 5.4: Pattern Spotting results

mAP for Pattern Spotting					
Method	Top n				
	100	300	500	700	1000
ResNet Conv	0.1447	0.1705	0.1738	0.1751	<b>0.1761</b>
ResNet GAPool	0.1225	0.1478	0.1524	0.1542	0.1557
VGG19 Block2-5	0.1118	0.1339	0.1386	0.1407	0.1425
VGG19 Block4-5	0.0997	0.1196	0.1237	0.1254	0.1268
VGG19 Blocks	0.0724	0.0848	0.0876	0.0888	0.0898
VGG19 Block2-3	0.0643	0.0761	0.0795	0.0811	0.0825
AlexNet used in (WIGGERS et al., 2019)	0.0610	0.0674	0.0689	0.0697	0.0703

Table 5.5: Pattern Spotting results with Hashing

mAP for Pattern Spotting					
Method	Top n				
	100	300	500	700	1000
VGG19 Block4-5 H	0.1094	0.1341	0.1388	0.1409	<b>0.1426</b>
VGG19 Block2-5 H	0.0935	0.1129	0.1163	0.1176	0.1186
ResNet GAPool H	0.0911	0.1040	0.1061	0.1070	0.1077
ResNet Conv H	0.0303	0.0311	0.0313	0.0314	0.0315



Post-processing was applied using specific parameter values to refine the results obtained from the PS task. The parameter  $\beta$  was set to 3000, allowing the selection of the top 3000 candidates based on their similarity scores. For the IoU parameter  $\gamma$ , a value of 0.85 was chosen, ensuring that regions with an IoU greater than 0.85 were merged together.

After performing the union operation, the resulting regions were further evaluated using the mAP metric. The top 100, 300, 500, 700, and 1000 regions with the highest mAP scores were selected for further analysis. Notably, an improvement in mAP was observed across all these result sets after applying the post-processing step (PP). This indicates that the post-processing procedure effectively enhances the accuracy and quality of the PS task results.

Table 5.6: Pattern Spotting results with PP

<b>mAP for Pattern Spotting</b>					
<b>Method</b>	<b>Top n</b>				
	100	300	500	700	1000
ResNet Conv PP	0.1716	0.1946	0.1974	0.1986	<b>0.1996</b>
ResNet GAPool PP	0.1432	0.1674	0.1712	0.1729	0.1743
VGG19 Block2-5 PP	0.1331	0.1535	0.1577	0.1598	0.1615
VGG19 Block4-5 H PP	0.1302	0.1531	0.1572	0.1593	0.1610
VGG19 Block4-5 PP	0.1181	0.1364	0.1401	0.1417	0.1429
VGG19 Block2-5 H PP	0.1119	0.1294	0.1322	0.1335	0.1345
ResNet GAPool H PP	0.1059	0.1173	0.1192	0.1200	0.1207
VGG19 Blocks PP	0.0861	0.0975	0.1000	0.1011	0.1021
VGG19 Block2-3 PP	0.0755	0.0867	0.0899	0.0915	0.0927
AlexNet used in (WIGGERS <i>et al.</i> , 2019) PP	0.0691	0.0753	0.0768	0.0775	0.0781
ResNet Conv H PP	0.0332	0.0340	0.0342	0.0344	0.0345

When comparing the results with state-of-the-art methods, it could be observed that the ResNet Conv PP method outperforms the approach proposed by Wiggers *et al.* (WIGGERS *et al.*, 2019) PP by 2.56 percentage points. This improvement highlights the effectiveness of the proposed method in the context of the PS task.

Table 5.7 provides a comprehensive overview of the main results achieved by state-of-the-art methods and the results obtained by the models presented in this paper. The comparison allows us to assess the competitiveness and performance of the proposed models in pattern spotting. While the method using deep hashing may not outperform all state-of-the-art approaches, it still demonstrate good performance and hold promise for practical applications in various domains.

Table 5.7: Comparison of the methods with the state-of-the-art PS

Method	PS Top 1000
ResNet Conv PP	<b>0.1996</b>
Wiggers <i>et al.</i> (WIGGERS <i>et al.</i> , 2019) PP	0.1740
Ubeda <i>et al.</i> (ÚBEDA <i>et al.</i> , 2019) PP	0.1730
VGG19 Block4-5 H PP	0.1610
En <i>et al.</i> (EN <i>et al.</i> , 2016b)	0.1570
Ubeda <i>et al.</i> (ÚBEDA <i>et al.</i> , 2019) ES	0.1390

Figure 5.2 displays the qualitative results obtained from the search process using the ResNet Conv PP feature map, which has 100,352 dimensions for five different queries. The results are visually promising, as most of the top five retrieved images closely resemble those used as search queries. This indicates that the retrieval system successfully captures and retrieves similar images, highlighting the approach’s effectiveness in this work. Figure 5.3 shows the qualitative results of VGG19 Block4-5 Hashing for the same five queries. It is important to note that this network extracts a 1,024-dimensional binary feature map.































Query	ResNet Conv Search Results				
	1°	2°	3°	4°	5°
					
					
					
					
					

Figure 5.2: Qualitative results of the search of some queries in the DocExplore database. The figure shows the image used in the query and its first five results returned by the ResNet Conv method.

























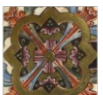
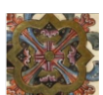
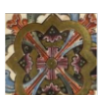
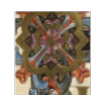
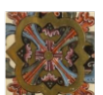

Query	VGG19 Block4-5 Hashing Search Results				
	1°	2°	3°	4°	5°
					
					
					
					
					

Figure 5.3: Qualitative results of the search of some queries in the DocExplore database. The figure shows the image used in the query and its first five results returned by the VGG19 Block4-5 Hashing method.

## 5.5 Search Time and Storage Cost

The Table 5.8 presents the average query search time and the corresponding storage requirements for the feature maps of all candidates during the offline phase. The average time is calculated based on 50 different queries, providing a comprehensive overview of the system’s efficiency. The storage space refers specifically to the size of the feature maps, excluding any additional structures implemented for indexing and storage optimization. These metrics offer insights into the computational demands and storage considerations associated with the approach in this work.

## 5.6 Final Considerations

The experimental results indicate that the models based on the VGG19 architecture outperformed the ResNet50V2 models in the Image Retrieval (IR) task, which involves finding the best occurrence of a query on a page. On the other hand, the ResNet50V2 models achieved better performance in the Pattern Spotting (PS) task, where the objective is to identify and locate multiple similar images on a page.

Table 5.8: Results for processing time and storage

Method	Features	Time (s)		Storage (GB)	
		FP	Binary	FP	Binary
ResNet Conv	100 352	44.59	20.75	294.11	9.19
ResNet GAPool	2 048	4.54	3.07	6.00	0.19
VGG19 Blocks	1 472	4.19	—	4.31	—
VGG19 Block4-5	1 024	4.10	2.94	3.00	0.09
VGG19 Block2-3	384	3.91	—	1.13	—
VGG19 Block2-5	640	3.94	2.85	1.88	0.06
AlexNet used in (WIGGERS <i>et al.</i> , 2019)	4 096	12.81	—	12.00	—
Wiggers <i>et al.</i> (WIGGERS <i>et al.</i> , 2019)	4 096	588.65	—	551.76	—

FP: Floating-point.

One of the significant contributions of this work lies in the substantial reduction of computational effort required to solve the problem. While the previous method proposed by Wiggers *et al.* (WIGGERS *et al.*, 2019) involved more than 36 million candidate regions, the proposed method successfully reduced this number to approximately 780 thousand candidate regions, confirming Hypothesis #4. This optimization not only improved the overall results but also significantly reduced the processing time, as demonstrated in Tables 5.7, 5.3, and 5.8. For instance, while the method proposed by Wiggers *et al.* (WIGGERS *et al.*, 2019) required over 580 seconds to search for a query, the proposed method accomplished the same task in a maximum of 45 seconds, with superior results and confirming Hypothesis #1.

Another advantage of the approach in this work is the application of deep hashing techniques. The performance was relatively maintained despite transforming the feature representations into binary values. In contrast, considering that the query search in the method proposed by Wiggers *et al.* (WIGGERS *et al.*, 2019) could take 200 times longer than using hashing methods, the trade-off between search time and result quality is favorable. Additionally, the storage cost was significantly reduced as well. While the previous method would require over 550 GB of storage, the approach using VGG19 with Hashing requires less than 0.09 GB, confirming Hypothesis #2.

The transformation of feature representations to the binary domain can sometimes yield improved results, as observed in the VGG19 Block4-5 model, where an improvement of nearly two percentage points was achieved, which could confirm Hypothesis #3. Furthermore, converting to binary codes resulted in approximately a 30% reduction in query search time for methods utilizing VGG19. In terms of storage, the transition from 32-bit floating-point values to 1-bit binary codes reduced the overall storage cost by a factor of 32.

# Chapter 6

## Conclusions

This research addressed the challenges of the Image Retrieval (IR) and Pattern Spotting (PS) tasks for a collection of historical document images. Two distinct approaches were proposed to improve the performance and efficiency of these tasks.

The first approach focused on enhancing an existing method by reducing the number of candidate images returned by the selective search algorithm. This optimization improved multiple aspects, including mAP, processing time, and storage cost. The reduction in the number of candidates yielded more accurate and efficient results. However, it was observed that while the IR task benefited significantly from this improvement, the PS task did not necessarily exhibit the same performance gains across different convolutional neural network (CNN) architectures.

The second approach aimed to reduce storage costs and processing time by leveraging deep hashing techniques. The storage requirements were drastically reduced by transforming feature representations into binary codes, and the search process became more efficient. Interestingly, this investigation revealed that in some instances, the conversion to the binary domain could even enhance the performance compared to using real-valued features. Even when performance gains were not achieved, the trade-off between reduced computational resources and minor performance loss made the approach viable.

Future research efforts will focus on fine-tuning the CNNs used as feature extractors for historical document images. Currently, these CNNs are pretrained on the ImageNet dataset, which might not capture historical documents' unique characteristics and nuances. By fine-tuning the CNNs on images of a similar context, it is expected that the mAP could be further improved, leading to more accurate and reliable results in the IR and PS tasks.

# Bibliography

- ALPAYDIN, E. *Introduction to Machine Learning, third edition*. [S.l.]: MIT Press, 2014. (Adaptive Computation and Machine Learning series). ISBN 9780262325752.
- CANTINI, R.; MAROZZO, F.; BRUNO, G.; TRUNFIO, P. Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs. *ACM Transactions on Knowledge Discovery from Data*, v. 16, 05 2021.
- CHICCO, D. Siamese neural networks: An overview. In: \_\_\_\_\_. *Artificial Neural Networks*. New York, NY: Springer US, 2021. p. 73–94. ISBN 978-1-0716-0826-5.
- EAKINS, J.; GRAHAM, M.; EAKINS, J.; GRAHAM, M.; FRANKLIN, T. Content-based image retrieval. *Library and Information Briefings*, v. 85, p. 1–15, 1999.
- EN, S.; NICOLAS, S.; PETITJEAN, C.; JURIE, F.; HEUTTE, L. New public dataset for spotting patterns in medieval document images. *Journal of Electronic Imaging*, v. 26, p. 011010, 11 2016.
- EN, S.; PETITJEAN, C.; NICOLAS, S.; HEUTTE, L. A scalable pattern spotting system for historical documents. *Pattern Recognition*, Elsevier, v. 54, p. 149–161, jun. 2016.
- FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient graph-based image segmentation. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 59, n. 2, p. 167–181, 2004.
- GKELIOS, S.; SOPHOKLEOUS, A.; PLAKIAS, S.; BOUTALIS, Y.; CHATZICHRISTOFIS, S. A. Deep convolutional features for image retrieval. *Expert Systems with Applications*, v. 177, p. 114940, 2021. ISSN 0957-4174.
- HADSELL, R.; CHOPRA, S.; LECUN, Y. Dimensionality reduction by learning an invariant mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. [S.l.: s.n.], 2006. v. 2, p. 1735–1742.

- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Identity mappings in deep residual networks. In: LEIBE, B.; MATAS, J.; SEBE, N.; WELLING, M. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 630–645.
- HEBBAR, H.; NIRANJAN, U.; MUSHIGERI, S. Content based image retrieval based on cumulative distribution function a performance evaluation. *International Journal of Computer Applications*, v. 81, p. 16–22, 11 2013.
- HUNEITI, A.; DAOUD, M. Content-based image retrieval using som and dwt. *Journal of Software Engineering and Applications*, v. 8, p. 51–61, 03 2015.
- KAMNITSAS, K.; LEDIG, C.; NEWCOMBE, V. F.; SIMPSON, J. P.; KANE, A. D.; MENON, D. K.; RUECKERT, D.; GLOCKER, B. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, v. 36, p. 61–78, 2017. ISSN 1361-8415.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGESS, C.; BOTTOU, L.; WEINBERGER, K. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2012. v. 25.
- LAW, H.; DENG, J. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, v. 128, 03 2020.
- LUO, X.; WU, D.; CHEN, C.; DENG, M.; HUANG, J.; HUA, X.-S. *A Survey on Deep Hashing Methods*. 2021.
- MALIK, F.; BAHARUDIN, B. Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the dct domain. *Journal of King Saud University - Computer and Information Sciences*, v. 25, n. 2, p. 207–218, 2013. ISSN 1319-1578.
- MELEKHOV, I.; KANNALA, J.; RAHTU, E. Siamese network features for image matching. *2016 23rd International Conference on Pattern Recognition (ICPR)*, p. 378–383, 2016.
- MUNJAL, M. N.; BHATIA, S. A novel technique for effective image gallery search using content based image retrieval system. In: *2019 International Conference on Machine*

- Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. [S.l.: s.n.], 2019. p. 25–29.
- NOWOZIN, S. Optimal decisions from probabilistic models: The intersection-over-union case. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014.
- Riba, P.; Lladós, J.; Fornés, A.; Dutta, A. Large-scale graph indexing using binary embeddings of node contexts for information spotting in document image databases. *Pattern Recognition Letters*, v. 87, p. 203–211, fev. 2017.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015.
- SALAKHUTDINOV, R.; HINTON, G. Semantic hashing. *Int. J. Approx. Reasoning*, v. 50, p. 969–978, 07 2009.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- SMEULDERS, A.; WORRING, M.; SANTINI, S.; GUPTA, A.; JAIN, R. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 12, p. 1349–1380, 2000.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367.
- TORRES, R. D. S.; FALCÃO, A. X. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, v. 13, p. 161–185, 2006.
- ÚBEDA, I.; SAAVEDRA, J. M.; NICOLAS, S.; PETITJEAN, C.; HEUTTE, L. Pattern spotting in historical documents using convolutional models. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. New York, NY, USA: Association for Computing Machinery, 2019. (HIP '19), p. 60–65. ISBN 9781450376686.
- UIJLINGS, J.; SANDE, K.; GEVERS, T.; SMEULDERS, A. Selective search for object recognition. *International Journal of Computer Vision*, v. 104, p. 154–171, 09 2013.



WEISS, Y.; TORRALBA, A.; FERGUS, R. Spectral hashing. In: . [S.l.: s.n.], 2008. p. 1753–1760.

WIGGERS, K. L.; JUNIOR, A. de S. B.; KOERICH, A. L.; HEUTTE, L.; OLIVEIRA, L. E. S. de. *Deep Learning Approaches for Image Retrieval and Pattern Spotting in Ancient Documents*. 2019.

YARLAGADDA, P.; MONROY, A.; CARQUE, B.; OMMER, B. Recognition and analysis of objects in medieval images. In: SPRINGER. *Proceedins of the Aian Conference on Computer Vision, Workshop on e-Heritage*. [S.l.]: Springer, 2010. p. 296–305.

ZITNICK, C. L.; DOLLÁR, P. Edge boxes: Locating object proposals from edges. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014. p. 391–405. ISBN 978-3-319-10602-1.