

**ZACARIAS CURI FILHO**

**IMAGE RETRIEVAL AND PATTERN  
SPOTTING IN HISTORICAL DOCUMENTS  
USING FULLY CONVOLUTIONAL  
APPROACHES**

Thesis presented to the Graduate Program  
in Informatics of the Pontifícia Universidade  
Católica do Paraná as a partial requirement for  
the degree of Doctor in Informatics.

**Curitiba**

**2023**

**ZACARIAS CURI FILHO**

**IMAGE RETRIEVAL AND PATTERN  
SPOTTING IN HISTORICAL DOCUMENTS  
USING FULLY CONVOLUTIONAL  
APPROACHES**

Thesis presented to the Graduate Program  
in Informatics of the Pontifícia Universidade  
Católica do Paraná as a partial requirement for  
the degree of Doctor in Informatics.

Major Field: Computer Science

Supervisor: Prof. Dr. Alceu de Souza Britto  
Junior

Co-supervisor: Prof. Dr. 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

C975i  
2023 Curi Filho, Zacarias  
Image retrieval and pattern spotting in historical documents using fully  
convolutional approaches / Zacarias Curi Filho ; supervisor: Alceu de Souza Britto  
Junior ; co-supervisor: Laurent Heutte. – 2023  
118 f. ; il. : 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2023  
Bibliografia: f. 101-111

1. Recuperação de imagem baseada em conteúdo. 2. Processamento de  
imagens. 3. Reconhecimento de padrões. 4. Informática. I. Britto Júnior, Alceu de  
Souza. II. Heutte, Laurent. III. Pontifícia Universidade Católica do Paraná.  
Programa de Pós-Graduação em Informática. IV. Título.

CDD. 20. ed. – 004



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

Curitiba, 01 de fevereiro de 2024.


04-2024

## DECLARAÇÃO

Declaro para os devidos fins, que **ZACARIAS CURI FILHO** defendeu a tese de Doutorado intitulada **“Image Retrieval and Pattern Spotting in Historical Documents Using Fully Convolutional Approaches”**, na área de concentração Ciência da Computação no dia 13 de setembro de 2023, no qual foi aprovado.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.

Documento assinado digitalmente  
 **EMERSON CABRERA PARAISO**  
Data: 01/02/2024 11:42:13-0300  
Verifique em <https://validar.iti.gov.br>

---

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

# Acknowledgements

First and foremost, I express my gratitude to God for the gift of life and for guiding me throughout this journey.

To my family, my deepest thanks for their unwavering support. To my parents, Eliane and Zacarias, for being constant sources of encouragement and support in my personal and professional growth. To my wife, Milena, for her tireless partnership and patience.

To professors Alceu de Souza Britto Jr. and Laurent Heutte, I am immensely grateful for their guidance, valuable teachings, and advice, which were fundamental to the completion of this work and to my development.

To professors Stéphane Nicolas, Pierrick Tranouez, and José M. Saavedra, my acknowledgment for the numerous meetings and constructive suggestions that significantly contributed to shaping this thesis.

To all colleagues from PPGIa (PUCPR) and LITS (Université de Rouen Normandie), whose presence and exchange of ideas enriched my academic journey.

This study was made possible by financial support from CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil) - Finance Code 001 and the PlaIR2018 project, co-financed by the European Union with the European Regional Development Fund (ERDF) and by the Normandy Region.

Furthermore, I express my profound gratitude to all other friends, family members, colleagues, and institutions who, in various ways, contributed to the development of this work and to my personal and professional growth.

*“The important thing in science is not so much  
to obtain new facts as to discover new ways of  
thinking about them.”  
(Sir William Lawrence Bragg)*

# Resumo

O aumento contínuo do acervo das bibliotecas digitais torna necessário a criação de ferramentas capazes de extrair informações e realizar buscas em grandes volumes de documentos com características variadas. Uma das maneiras de se realizar essas buscas é a utilização dos padrões visuais presentes nos documentos. Visando encontrar padrões visuais similares a uma determinada query em um conjunto de imagens de pesquisa, este trabalho apresenta novas soluções para as tarefas de Content-Based Image Retrieval (CBIR) e Graphical Pattern Spotting (PS) em manuscritos históricos digitalizados. Esses padrões podem variar em relação ao tamanho, formato, cor e contexto. Além das diferenças existentes, nenhum conjunto pré-definido de padrões é disponibilizado para o treinamento, sendo necessário o uso de soluções capazes de usar qualquer objeto presente nas páginas do dataset como query. Neste trabalho, são propostos métodos que não necessitam de treinamento com as imagens e queries do problema. Para isso, a query e as imagens dos documentos são representados por mapas de características obtidos por camadas intermediárias de uma rede Fully Convolutional treinada com fotografias. A comparação entre os mapas de características é realizada por uma operação de correlação cruzada, produzindo mapas de calor com as similaridades. Duas variações do método são apresentadas: uma com o uso de descritores do tipo float e outra com descritores binários. Os novos métodos foram implementados e avaliados com um protocolo experimental robusto, composto por três datasets diferentes. Os experimentos realizados no Dataset DocExplore mostraram que o método proposto obteve um desempenho superior ao estado da arte nas duas variações. O método com descritores do tipo float permitiu um aumento no mAP de 136.8% para PS e 40.2% para CBIR, enquanto o método binário proporcionou um aumento de 134.6% para PS e 38.46% para CBIR. Além disso, os experimentos realizados para a tarefa de PS no dataset Horae e logo spotting no dataset Tobacco800 confirmaram a alta capacidade de generalização do método proposto.

**Palavras-chaves:** Content-Based Image Retrieval, Pattern Spotting, Fully-Convolutional Network, Cross-Correlation

# Abstract

The constant increase of the collection of digital libraries brings the necessity about create tools capable of extracting information and conduct searches in a large number of documents within distinct characteristics. One way to perform such searches is to use visual patterns present in the documents. Aiming to find similar visual patterns to a given query in a set of search images, this work presents new solutions for the tasks of Content-Based Image Retrieval (CBIR) and Graphical Pattern Spotting (PS) in digitized historical manuscripts. These patterns can vary in size, shape, color and context, and no pre-defined set of patterns is available for training. Therefore, it is necessary to use solutions that can use any object present in the pages of the dataset as a query. In this work, methods are proposed that do not require training with the images and queries of the problem. For this, the query and the images of the documents are represented by feature maps obtained by intermediate layers of a Fully Convolutional Network trained with photographs. The comparison between the feature maps is performed by a cross-correlation operation, producing heatmaps with similarities. In addition, two variations of the method are presented: one with the use of float type features and another with binary features. The new methods were implemented and evaluated with a robust experimental protocol, composed of three different datasets. The experiments performed in the DocExplore Dataset showed that the proposed method obtained better performance than the state of the art in both variations. The method with float type features allowed an increase in the mAP of 136.8% for PS and 40.2% for CBIR, while the binary method provided an increase of 134.6% for PS and 38.46% for CBIR. Furthermore, the experiments performed for the PS task in the Horae dataset and logo spotting in the Tobacco800 dataset confirmed the high generalization ability of the proposed method.

**Keywords:** Content-Based Image Retrieval, Pattern Spotting, Fully-Convolutional Network, Cross-Correlation



# List of Figures

|           |  |    |
|-----------|--|----|
| Figure 1  | – Visual representation of Content-Based Image Retrieval (top) and Pattern Spotting (bottom) using the DocExplore dataset  | 19 |
| Figure 2  | – Example of a simple FCN network that takes an input image of arbitrary size and generates a feature map as output . . .  | 25 |
| Figure 3  | – An illustration of a convolution operation, whereby a region of the input matrix (highlighted in blue) is convolved with a convolutional filter . . . . .                          | 25 |
| Figure 4  | – Example of a 2x2 max-pooling operation . . . . .   | 26 |
| Figure 5  | – Example of a convolution operation with different strides values, highlighting the impact of stride on the resulting feature map size . . . . .                                    | 27 |
| Figure 6  | – Padding operation on a 5x5 matrix . . . . .  | 28 |
| Figure 7  | – Illustration of the Image Retrieval process, in which an image query is compared with all images in a database and a ranking is generated based on their similarities . . . . .    | 29 |
| Figure 8  | – Overview of a traditional CBIR process . . . . .   | 30 |
| Figure 9  | – Illustration of the Pattern Spotting process, in which an image query is compared to images in a database, and an object ranking list is generated based on the similarities . . . | 31 |
| Figure 10 | – Local Binary Pattern calculation. (VERMA; RAMAN, 2018)   | 35 |
| Figure 11 | – Sample images from the DocExplore dataset . . . . .  | 44 |
| Figure 12 | – Categories of objects annotated in DocExplore dataset. (EN et al., 2016a) . . . . .  | 45 |
| Figure 13 | – Intra-category variability of the annotated objects. (EN et al., 2016a) . . . . .  | 45 |
| Figure 14 | – Box plot related to the size of DocExplore queries . . . . .   | 46 |
| Figure 15 | – Box plot related to the aspect ratio of DocExplore queries   | 47 |
| Figure 16 | – Sample images from the Horae dataset. . . . .  | 47 |
| Figure 17 | – Sample images from the Tobacco800 dataset . . . . .  | 48 |
| Figure 18 | – Categories of objects in the Tobacco800 dataset. The number in parenthesis represents the amount of occurrences . . .  | 49 |
| Figure 19 | – Box plot related to the size of Tobacco800 queries . . . . .   | 50 |

|           |   |    |
|-----------|---|----|
| Figure 20 | –Box plot related to the aspect ratio of Tobacco800 queries   | 50 |
| Figure 21 | –Overview of the proposed framework for image retrieval and pattern spotting. . . . .   | 57 |
| Figure 22 | –Fully convolutional version of VGG16 architecture. . . . .   | 59 |
| Figure 23 | –Cross-correlation function. (a) feature map of search image, (b) feature map of query image, (c) generated heatmap. The white part of (a) indicates the margin. . . . .  | 62 |
| Figure 24 | –Process for selecting the best values from the heatmap using $p = 3$ . (a) input query, (b) input search image, (c) original heatmap, (d) heatmap after removal of borders, (e) selection of best values and removal of neighbors. . . . . | 65 |
| Figure 25 | –Overview of the proposed framework for image retrieval and pattern spotting with binary features . . . . .   | 67 |
| Figure 26 | –Proposed method for creating the global squeeze vector . . . . .   | 67 |
| Figure 27 | –XOR Cross-Correlation used to compute the heatmap between the query and each document page image. . . . .  | 69 |
| Figure 28 | –Nemenyi test for different PCA configurations in the PS task. . . . .  | 77 |
| Figure 29 | –Box plot with IR results for the DocExplore dataset . . . . .  | 80 |
| Figure 30 | –Box plot with PS results for the DocExplore dataset . . . . .  | 82 |
| Figure 31 | –Retrieval results for DocExplore. The objects in red represent errors. . . . .   | 84 |
| Figure 32 | –Categories with the worst results. . . . .   | 85 |
| Figure 33 | –Samples from bed crown category. . . . .   | 86 |
| Figure 34 | –Query resized for multiple input sizes of the system. . . . .  | 87 |
| Figure 35 | –Retrieval results for Horae dataset. . . . .   | 89 |
| Figure 36 | –Results for one page in the Horae dataset. The white bound boxes represent the results found. The black bound box represents the object that was not detected. . . . .   | 90 |
| Figure 37 | –Retrieval results for Tobacco800 dataset. . . . .  | 91 |
| Figure 38 | –Nemenyi test considering float and binary representations of different sizes (128, 64, 32 and 16) . . . . .  | 94 |
| Figure 39 | –Qualitative results obtained by applying the binary method on the DocExplore dataset . . . . .   | 95 |

Figure 40 – Different input styles evaluated in experiments with multiple  
inputs. . . . . 116

# List of Tables

|          |  |     |
|----------|--|-----|
| Table 1  | – Summary of the main works for Image Retrieval and Pattern Spotting in ancient documents . . . . .  | 51  |
| Table 2  | – Results with different layers of VGG16 architecture for the DocExplore dataset using PCA with 64 components. The time represents the average time used for all stages of the online phase for one query. The memory represents the space used to store features for all images in the dataset. . . . . | 74  |
| Table 3  | – Results of PS with different layers of the VGG16 architecture by category for the DocExplore dataset. . . . .  | 76  |
| Table 4  | – Effect of applying different PCA components to the output of the block3. The time represents the average time used for all stages of the online phase for one query. . . . .   | 77  |
| Table 5  | – Results for DocExplore dataset (using the evaluation protocol presented in (EN et al., 2016a)). . . . .  | 78  |
| Table 6  | – DocExplore results for IR considering query sizes and the aspect ratio (using the evaluation protocol presented in (ÚBEDA et al., 2020)). . . . .  | 78  |
| Table 7  | – DocExplore results for PS considering query sizes and the aspect ratio (using the evaluation protocol presented in (ÚBEDA et al., 2020)). . . . .  | 79  |
| Table 8  | – Image Retrieval and Logo Spotting results for Tobacco800 (using the evaluation protocol presented in (WIGGERS et al., 2019a)). . . . .   | 91  |
| Table 9  | – Experimental results of the proposed method on the DocExplore dataset considering float and binary feature vectors with different sizes (number of PCA components). mAP for IR and PS tasks plus the memory consumption in GigaBytes   | 93  |
| Table 10 | – Results for DocExplore dataset (using the evaluation protocol presented in (EN et al., 2016a)) . . . . .   | 96  |
| Table 11 | – Results of feature normalization for the DocExplore subset.  | 114 |
| Table 12 | – Results of heatmap normalization for the DocExplore subset.  | 114 |

|          |   |     |
|----------|---|-----|
| Table 13 | –Results of using different numbers of regions for the DocExplore subset. . . . .                           | 115 |
| Table 14 | –Results of using multiple inputs for the DocExplore subset. . . . .  | 116 |
| Table 15 | –Results of experiments with a Gaussian filter applied to query features for the DocExplore subset. . . . . | 116 |
| Table 16 | –Results of experiments with a Sobel filter for the DocExplore subset. . . . .                              | 117 |
| Table 17 | –Results of experiments with pos-processing for the DocExplore subset. . . . .                              | 118 |
| Table 18 | –Results of the proposed method with varying threshold values for the DocExplore subset. . . . .            | 118 |

# List of abbreviations and acronyms

|       |  |
|-------|--|
| AP    | Average Precision                          |
| BING  | Binarized Normed Gradients                 |
| BoW   | Bag of Words                               |
| BRIEF | Robust Independent Elementary Features     |
| BRISK | Binary Robust Invariant Scalable Keypoints |
| CBIR  | Content-Based Image Retrieval              |
| CNN   | Convolutional Neural Network               |
| FV    | Fisher Vector                              |
| FCN   | Fully Convolutional Neural Network         |
| FREAK | Fast Retina Keypoint                       |
| LBP   | Local Binary Pattern                       |
| LDA   | Latent Dirichlet Allocation                |
| IR    | Image Retrieval                            |
| IoU   | Intersection over Union                    |
| mAP   | mean of Average Precision                  |
| MSE   | Mean Squared Error                         |
| NMS   | Non-maximum Suppression                    |
| PS    | Pattern Spotting                           |
| PCA   | Principal Component Analysis               |
| ROI   | Region of Interest                         |
| SIFT  | Scale Invariant Feature Transform          |

|        |   |
|--------|---|
| SCNN   | Siamese Convolutional Neural Network      |
| SGD    | Stochastic Gradient Descent               |
| SVM    | Support Vector Machine                    |
| TF-IDF | Term Frequency–Inverse Document Frequency |
| VLAD   | Vector of Locally Aggregated Descriptors  |
| XOR    | Exclusive OR                              |

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>17</b> |
| 1.1      | Proposal   | 21        |
| 1.2      | Objectives   | 21        |
| 1.3      | Hypotheses   | 22        |
| 1.4      | Contributions  | 22        |
| 1.5      | Publications   | 22        |
| 1.6      | Document Structure   | 23        |
| <b>2</b> | <b>Literature Review</b>                                     | <b>24</b> |
| 2.1      | Convolutional Feature Maps                                   | 24        |
| 2.2      | Image Retrieval and Pattern Spotting                         | 28        |
| 2.2.1    | Preprocessing  | 31        |
| 2.2.2    | Image representation   | 34        |
| 2.2.3    | Similarity measures  | 41        |
| 2.2.4    | Post-processing  | 42        |
| 2.2.5    | Datasets of Historical Documents Images                      | 43        |
| 2.2.5.1  | DocExplore   | 44        |
| 2.2.5.2  | Horae  | 46        |
| 2.2.5.3  | Tobacco800   | 48        |
| 2.2.6    | Image Retrieval and Pattern Spotting in Historical Documents | 49        |
| 2.3      | Final Considerations   | 54        |
| <b>3</b> | <b>Proposed Method</b>                                       | <b>56</b> |
| 3.1      | Proposed Method for Image Retrieval and Pattern Spotting     | 56        |
| 3.1.1    | Offline phase  | 56        |
| 3.1.1.1  | Adaptive Batch   | 57        |
| 3.1.1.2  | Feature Extraction   | 58        |
| 3.1.1.3  | Principal Component Analysis                                 | 60        |
| 3.1.1.4  | Features Normalization                                       | 60        |
| 3.1.2    | Online phase   | 60        |
| 3.1.2.1  | Features Extraction and Processing                           | 61        |



|          |   |            |
|----------|---|------------|
| 3.1.2.2  | Cross-Correlation . . . . .                                 | 61         |
| 3.1.2.3  | Bicubic Interpolation . . . . .                             | 62         |
| 3.1.2.4  | Region Selection . . . . .                                  | 63         |
| 3.1.2.5  | Ranking Creation . . . . .                                  | 66         |
| 3.1.2.6  | Multi-Scale Input . . . . .                                 | 66         |
| 3.2      | Proposed Method with Binary Features . . . . .              | 66         |
| 3.2.1    | Binarization Strategy . . . . .                             | 67         |
| 3.2.2    | XOR Cross-Correlation . . . . .                             | 69         |
| 3.3      | Evaluation Protocol . . . . .                               | 70         |
| 3.4      | Final Considerations . . . . .                              | 71         |
| <b>4</b> | <b>Experimental Results . . . . .</b>                       | <b>73</b>  |
| 4.1      | Experiments with the Method Using Float Features . . . . .  | 73         |
| 4.1.1    | Results on DocExplore Dataset . . . . .                     | 73         |
| 4.1.2    | Results on Horae Dataset . . . . .                          | 88         |
| 4.1.3    | Results on Tobacco800 Dataset . . . . .                     | 90         |
| 4.1.4    | Discussion and Analysis . . . . .                           | 92         |
| 4.2      | Experiments with the Method Using Binary Features . . . . . | 92         |
| 4.2.1    | Results on DocExplore Dataset . . . . .                     | 92         |
| 4.3      | Final Considerations . . . . .                              | 96         |
| <b>5</b> | <b>Conclusions . . . . .</b>                                | <b>98</b>  |
|          | <b>Bibliography . . . . .</b>                               | <b>101</b> |
|          | <b>Appendix . . . . .</b>                                   | <b>112</b> |
|          | <b>APPENDIX A Supplementary Experiments . . . . .</b>       | <b>113</b> |
| A.1      | Detailed Exploration of Experimental Results . . . . .      | 113        |
| A.1.1    | L2 normalization . . . . .                                  | 114        |
| A.1.2    | Heatmap Normalization . . . . .                             | 114        |
| A.1.3    | Number of Selected Regions . . . . .                        | 114        |
| A.1.4    | Use of Multiple Inputs . . . . .                            | 115        |
| A.1.5    | Gaussian Filter . . . . .                                   | 116        |
| A.1.6    | Sobel Filter . . . . .                                      | 117        |
| A.1.7    | HOG for Pos-processing . . . . .                            | 117        |
| A.1.8    | Threshold . . . . .   | 118        |

# 1 Introduction

Recent advances in technology have enabled the storage of an ever-growing amount of information in digital libraries. In addition to data created digitally, numerous efforts have been made to digitize and store printed or manuscript content. This type of storage offers several advantages, particularly for fragile documents such as heritage documents, as it helps to preserve them for future generations.

Heritage documents, like medieval books, are generally more fragile than traditional printed documents due to natural degradation caused by time, storage, and human contact with the material. Digitization of this type of document provides a range of benefits, such as allowing access to a diverse collection of contents without the need for physical contact with the original material, which helps in preservation and conservation. Additionally, digitized documents can be easily shared and disseminated, allowing for a greater understanding of the past and its cultural heritage.

Several projects aim to digitize and make available ancient documents, as presented in the website "France-England, 700-1200: medieval manuscripts from the BnF and the British Library"<sup>1</sup>, where 800 documents from the *Bibliothèque nationale de France* and from British Library are available. In addition to these documents, the *Bibliothèque nationale de France* keeps part of its collection available on the gallica portal<sup>2</sup>, where thousands of ancient documents are available. The British Library<sup>3</sup> also makes part of its collection available digitally. Collections from other countries can also be accessed using repositories, as the Internet Archive<sup>4</sup>.

With the advancement of digitization, more documents are becoming available, making manual search unfeasible and necessitating an efficient search process. Traditional search systems typically rely on the use of metadata and tags, which can be created by specialists or automatically (YEE et al., 2003;

---

<sup>1</sup> <https://manuscripts-france-angleterre.org/>

<sup>2</sup> <https://gallica.bnf.fr/>

<sup>3</sup> <https://www.bl.uk/manuscripts/>

<sup>4</sup> <https://archive.org/>

CHEN; ZHENG; WEINBERGER, 2013). This type of search is efficient in many cases; however, it restricts the search to the selected metadata, hindering the search for books and documents that are not cataloged. A possible solution for these cases is proposed by Leydier et al. (2009), wherein a word retrieval method is used to create a search engine that is able to search books of different languages automatically. In Leydier's work, the search is performed using the words present in the text. Several methods can be employed for word-spotting, as those presented in Giotis et al. (2017).

A different search system can be performed using the visual content of the documents, as presented in En et al. (2016a). An example of this is some medieval manuscripts with illustrations, which can be used to make reading more comfortable and to represent graphically the history presented in the book. The use of these graphical elements allows for a search in addition to metadata, tags, and text, making it possible to retrieve pages based on an image query. An example of a situation where this type of search is efficient is when a historian performs a search related to a specific coat of arms. It is possible to provide an image with this symbol as an entry and search in a collection of documents for visually similar coat of arms.

The search for visual content can be performed with the Content-Based Image Retrieval (CBIR) and Pattern Spotting (PS) methods. These methods aim to search for objects that are visually similar to a query image in a collection of images. CBIR, also called Image Retrieval (IR), consists of finding images in collections based on visual similarity to a given query, which can consist of an entire image or only a part of it. The Pattern Spotting task, meanwhile, aims to retrieve the exact position of all objects that are visually similar to a query. A visual representation of these tasks is presented in Figure 1.

A recent challenge in the CBIR area is the realization of the retrieval without prior knowledge of which patterns will be researched. In this challenge, no training is performed on the problem queries, requiring a generic solution capable of identifying any object in the researched pages, even if there are differences concerning color, texture, size, shape, and context.

Historical documents possess certain peculiarities that present additional challenges for IR and PS tasks. Documents can be in various states of

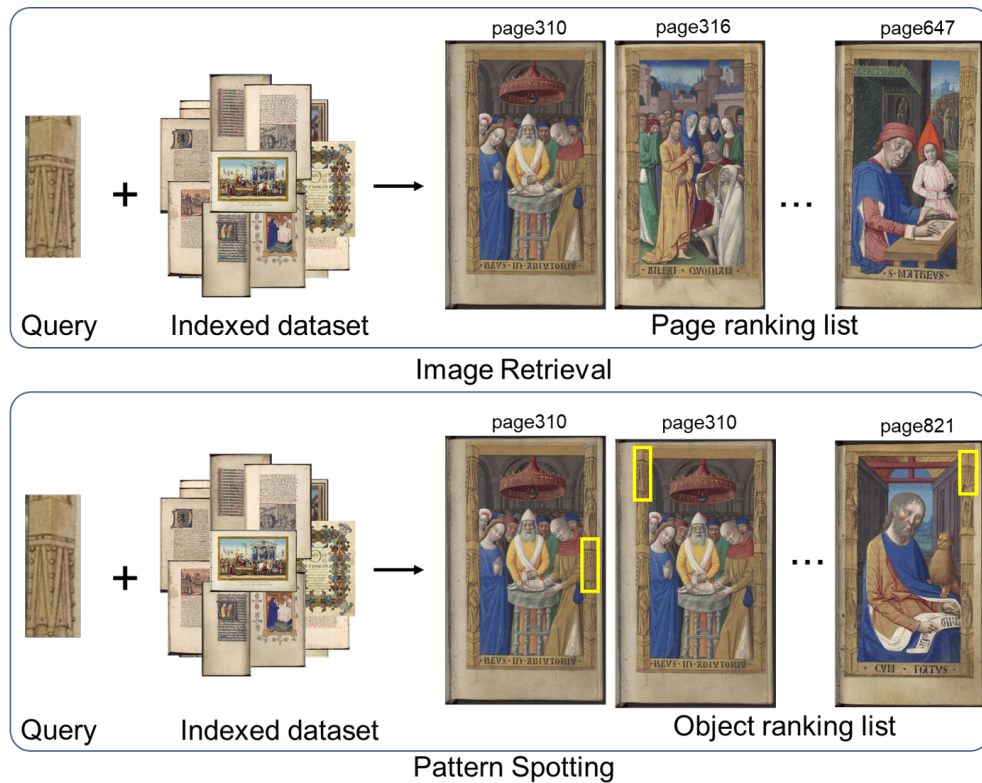


Figure 1 – Visual representation of Content-Based Image Retrieval (top) and Pattern Spotting (bottom) using the DocExplore dataset

conservation, which can alter the background and foreground colors. In some cases, the degradation can generate occlusions in the objects searched. As the same object can appear in different documents, there may be variations in ink, variations in the state of preservation, as well as changes in the size and shape of the objects. In addition to these peculiarities, all objects are hand-drawn and may have been drawn by different people, resulting in differences between all the visually similar objects.

Another particularity of IR and PS is the size of the image objects used as queries. As there is no previous knowledge about the queries, their size is not known either. Although we do not know the sizes, it is natural that historians are interested in details or small objects. For this reason, the system must be able to search for small objects on pages with different sizes. Besides the varied sizes, the pages of the historical documents used in this work do not present a default layout in their content. The existence of different layouts means that the query can be in different parts of the document and may have more than one occurrence in the same document.

Currently, most of the IR and PS methods available in the literature use a process composed of two phases: offline and online. In the offline phase, features that represent the images are extracted and stored. These features can be extracted in two different ways: segmentation-free and segmentation-based. In the segmentation-free approach, features are extracted directly from the entire image, forming a matrix of features. In the segmentation-based approach, a region proposal mechanism is used to define region proposals, which are then represented as feature vectors. In the online phase, a query is provided, and its features are extracted using the same feature extraction strategy used in the offline phase. After this extraction, the features of each image are compared with the features of the query, using a distance metric. Subsequently, a ranking is created to list the images with the most similar regions for the IR task and the positions of the most similar regions for the PS task.

One of the most important steps in the traditional IR and PS process is the feature extraction. The selection of an efficient technique is essential for a correct comparison between the query and the page (GHOSH; AGRAWAL; MOTWANI, 2018). The need for efficient representation becomes more important when there is a significant variation between objects considered visually similar. Nowadays, Deep Learning methods have been outstanding for allowing efficient abstractions of images from different domains, generating superior results than handcrafted representations in several challenges (LECUN; BENGIO; HINTON, 2015; SCHMIDHUBER, 2015).

Among the Deep Learning methods, a variation of the traditional CNNs has been highlighted in areas where the spatial location of the objects present in the image is needed: the fully convolutional neural network (FCN). These networks have as output a map of features, with vectors representing parts of the image rather than just a single representation for the entire image (LONG; SHELHAMER; DARRELL, 2015; WANG et al., 2015). This particularity allows the use of the features generated for the search of elements in parts of the input image, without the need for prior object localization.

A common characteristic in the techniques of IR and PS presented in the literature is the use of feature vectors to compare the candidate (a sub-image of the search image) with the query. A different strategy of comparison can be performed with the use of matrices to represent the local features of the images. Instead of grouping all matrices into one vector, it is possible to

compare each part of the image individually. This can be performed with the cross-correlation strategy (FÖRSTNER, 1986).

## 1.1 Proposal

Based on the concept of matrix representation and comparison of images, novel solutions for IR and PS tasks on historical documents are proposed. In the first proposed method, FCN-based feature maps are extracted from the document images, indexed, and stored during an offline phase. Then, a cross-correlation step compares the feature map extracted from the query with those representing the document images to be searched. This comparison characterizes an online phase in which a similarity heatmap provides a final ranking of retrieved document images (IR task) and each query occurrence position (PS task). The second proposed method includes a step where the features are transformed into binary, allowing for less storage space in the offline phase and less computational complexity in the online phase, thanks to the use of XOR cross-correlation.

## 1.2 Objectives

The main objective of this work is to develop solutions for Image Retrieval and Pattern Spotting tasks on digital collections of historical documents using matrix representations and comparisons. The concern is how the use of convolutional feature maps for image representation and the use of cross-correlation for the calculation of similarity between features affects the results. To this end, the following specific objectives are considered:

- Define and implement a training free image retrieval and pattern spotting method capable of using a fully convolutional neural network for feature extraction;
- Define and implement a method for binarizing the feature maps obtained by a fully convolutional neural network;
- Evaluate the proposed solution and compare it with state of the art.

## 1.3 Hypotheses

The hypotheses of this research are:

- H1) The use of FCN and cross-correlation for the creation of heatmaps allows the correct location and ranking of objects visually similar to a query image in historical document images.
- H2) An FCN model trained on the ImageNet dataset may provide robust features for representing digitized historical documents.
- H3) The use of binary features positively impacts the computational complexity maintaining the capability to locate the searched objects.

## 1.4 Contributions

The main contributions of this thesis are as follows:

- A segmentation-free method for IR and PS tasks applied to historical document images, combining a fully-convolutional model for feature extraction and a cross-correlation strategy for image comparison, representing the new state-of-the-art for the well-known DocExplore dataset;
- A representation strategy based on intermediate layers of an FCN, allowing the use of a model trained on a different domain (transfer learning);
- A binarization strategy focused on features obtained by intermediate activation layers of FCN networks through a global squeeze vector;
- An adaptation of the cross-correlation operation based on binary features, called XOR cross-correlation;
- A new strategy for detection of objects in heatmaps that avoids the use of non-maximum suppression process.

## 1.5 Publications

This PhD work has resulted in the following papers:

- Z. Curi, S. Nicolas, P. Tranouez, A. Britto and L. Heutte, "Image Retrieval and Pattern Spotting on Historical Documents with Binary Descriptors" in 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 2022 pp. 3893-3899.
- Z. Curi, S. Nicolas, P. Tranouez, J. Saavedra, A. Britto and L. Heutte, "A Segmentation-Free Method for Image Retrieval and Pattern Spotting in Historical Documents Using Convolutional Features" The paper is under review.

## 1.6 Document Structure

The document is organized into five chapters. Chapter 2 provides a comprehensive review of existing methods from the literature for IR and PS. Chapter 3 presents the new methods developed for IR and PS. Chapter 4 presents the results obtained from the experiments conducted. Finally, Chapter 5 provides a conclusion of the work and potential future directions.



## 2 Literature Review

This Chapter presents the works related to this study. The main works associated with Convolutional Feature Maps, Image Retrieval, and Pattern Spotting are presented.

### 2.1 Convolutional Feature Maps

Recently, Convolutional Neural Networks (CNNs) have been used with great success in several areas. CNNs are Deep Learning algorithms that can take an image as input and applying a succession of layers of trainable convolutions and spatial sub-sampling to extract features related to various aspects (contours, colors, contrasts, shadows, textures etc.) and objects in the image. These features are generated with the use of learnable weights and biases present in the convolutions. There are several architectures and variations of these networks presented in the literature. A description of how these networks work and their applications can be seen in Gu et al. (2018).

The most popular CNN architectures in the literature utilize a Fully Connected Layer (Dense layer) to generate a single feature vector which represents the entire input image; this approach is useful for tasks that require the entire image to be analyzed, such as classification. However, in certain cases, having a representation of the local features and a structured output is desired, which can be achieved through a Fully Convolutional Network (FCN). Several CNN architectures designed for image classification can be transformed into an FCN architecture by removing the Fully Connected Layers. These networks have as output a feature map containing local features of the image. One of the main advantages of using FCNs is that it enables the input of images of varied sizes and ratios. FCNs are used in a variety of areas, such as semantic segmentation (LONG; SHELHAMER; DARRELL, 2015), object detection (DAI et al., 2016), and visual tracking (WANG et al., 2015).

Figure 2 illustrates a simple FCN network. The network takes an image of arbitrary size as input and produces a feature map as output. One of the advantageous characteristics of using FCNs is that they are capable of accepting

images of varying sizes as input, since there are no dense layers which require a pre-defined size, as is the case with CNNs. The output of an FCN is a matrix with a height, width, and channels; the size of this matrix is determined by the number of pooling layers and strides included in the architecture. This output matrix can be viewed as a map of vectors, each representing a region of the image, otherwise known as a receptive field.

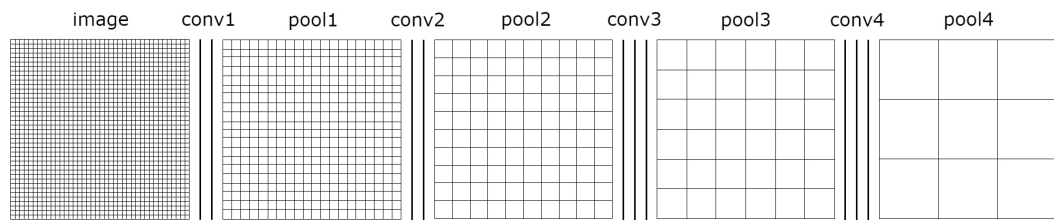


Figure 2 – Example of a simple FCN network that takes an input image of arbitrary size and generates a feature map as output

Convolutional filters are a fundamental operation in FCNs, allowing the networks to learn and extract important features from the input data. The use of these filters eliminates the need for pre-processing the input image, making the process more efficient. During training, the weights of these filters are learned, enabling the network to capture meaningful patterns and correlations in the image data. Figure 3 illustrates a convolutional layer, where a portion of the input matrix is compared with the filter. It is worth noting that the stride used in the pooling and convolutional layers can be adjusted to control the size of the resulting feature map.

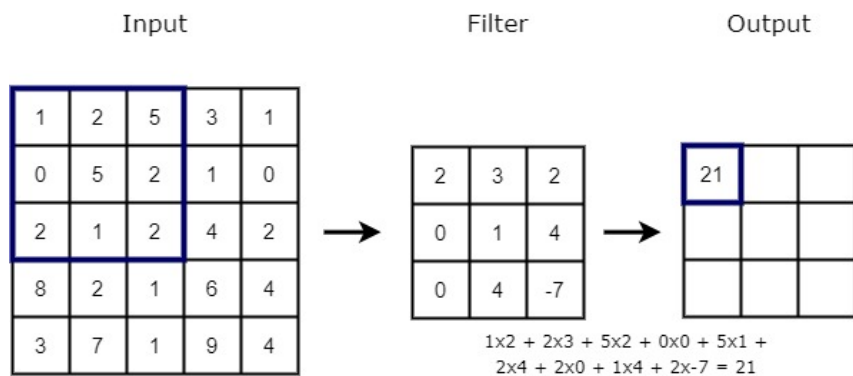


Figure 3 – An illustration of a convolution operation, whereby a region of the input matrix (highlighted in blue) is convolved with a convolutional filter

A convolution represents the summation of the multiplications of the elements of the input matrix with the filter. Given the feature map of the image  $I$  and the filter  $F$ , we aim to calculate  $G$  for each position  $(i, j)$  of the image  $I$ , as denoted in Eq. 2.1.

$$G[i, j] = \sum_{u=-Fw/2}^{Fw/2} \sum_{v=-Fh/2}^{Fh/2} \sum_{x=0}^{Fc} F[u, v, x] * I[i + u, j + v, x] \quad (2.1)$$

where  $Q_h$  represents the height,  $Q_w$  the width, and  $Q_c$  the channels.

The application of convolutional layers enables the summarization of pre-existing features within an image; however, some spatial generalizations are needed to make the feature maps less sensitive to local variations. These generalizations are performed by the pooling layers, which down-sample the feature maps by local generalizations on patches of the feature map. The max pooling method, illustrated in Figure 4, is a common approach, in which the largest value for each patch (distinct colors) of the feature map is calculated. Here, a 2x2 pixel operation with a stride of 2 pixels is applied, resulting in a 4x4 output. Apart from max pooling, other methods such as average pooling and min pooling are also commonly used in the literature; a comparison between these methods can be seen in (ZAFAR et al., 2022).

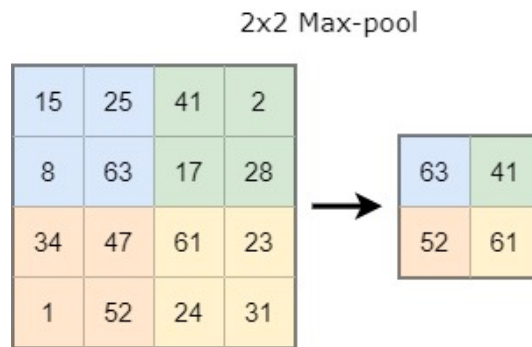


Figure 4 – Example of a 2x2 max-pooling operation

The stride used in the pooling and convolutional layers determines the size of the jump undertaken in the operation. Figure 5 shows a convolution with different values for the stride; it is noticeable that the larger the stride, the smaller the size of the resulting feature map. This property is often leveraged

to reduce the dimensionality of the feature maps and to speed up computation in convolutional neural networks.

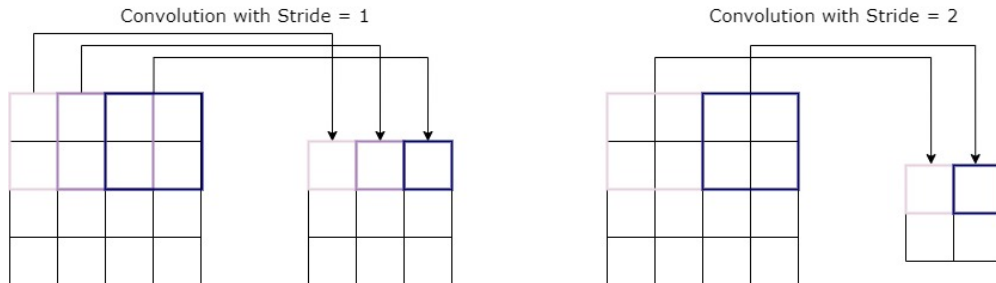


Figure 5 – Example of a convolution operation with different strides values, highlighting the impact of stride on the resulting feature map size

In many situations, it is necessary to add padding to the convolutional and pooling layers. The purpose of padding is to prevent spatial information from being lost. A popular strategy for this is to add zeros around the image, as illustrated in Figure 6. While this does not present a notable change to the generated features for most traditional cases, some situations can present a problem. For instance, when using small images as input, padding during the max pooling operation can cause the edges of the array to not be changed in the correct way, as the values (usually positive) are compared with zeros, resulting in few changes being made. An example of this problem can be seen when comparing the features generated by a single letter with the features of a letter in a digitized text image; the features of the single letter will be different from the features of the letter in context and may have a low similarity even if the object is the same.

Sometimes it is not possible to train with the classes or images of the problem, and it is necessary to use networks trained on different domains. These situations tend to produce irrelevant features due to the influence of the weights set during training. However, some strategies can be employed to reduce this influence. The shallow layers in an FCN represent structural information, but have a generalization caused by the network. The deeper layers have a strong influence from the weights learned during training and are more pertinent for obtaining semantic information, strongly related to the training classes. When adequate training is not conducted, it is expected that shallower or intermediate layers will perform better than when deep layers are utilized.

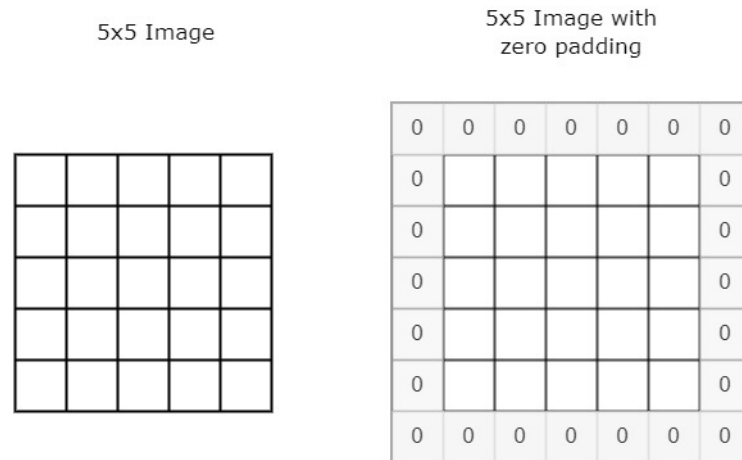


Figure 6 – Padding operation on a 5x5 matrix

Since the methods presented in this research do not consider the training of the network used, a strategy of using different layers to obtain the best features was adopted. Additionally, the characteristics and operation of different FCNs were taken into consideration to select the best network. Further details of the proposed method are provided in Chapter 3. The subsequent section presents the pattern spotting and Image Retrieval methods in the literature.

## 2.2 Image Retrieval and Pattern Spotting

The advances in storage capacity and the power of dissemination enabled by the Internet have given the public access to several digital documents, books, and image data collections. Included among these materials are ancient documents, such as medieval manuscripts. Several projects across the globe strive to digitalize, store, and share this kind of material, thus providing access to a far greater range of individuals. This can be observed in Nikolaidou et al. (2022), where a literature review of historical document datasets is presented. With the extensive sharing, audiences with diverse profiles can become interested in the content available, making the image metasearch techniques inefficient for certain types of search. An example of this is the situation where two identical documents are stored in countries with different languages, and probably their metadata will also be generated in different languages. If the search tags are not set in both languages, then possibly only one of the documents will be retrieved.

A solution for a more generic, language-independent, and indexation-independent search is the use of the visual content of documents. Many ancient documents have graphic elements, used to assist in reading and illustrate the content. These elements can be used as a source of search, independent of language or any type of document cataloging. This can be done with the task called Content-Based Image Retrieval (CBIR).

The CBIR task aims to retrieve images in a large database of digital images based on the similarity with the visual content of a given query. Figure 7 illustrates that process. As can be seen, an image query is compared with all the images in a database of images, and a ranking is returned based on the similarities between the query and each image (or parts of it). One of the first reported works in this area was (CHANG; SHI; YAN, 1987). Since the work of Chang and colleagues, several studies have been published in the literature, presenting different techniques applied to images from several domains.

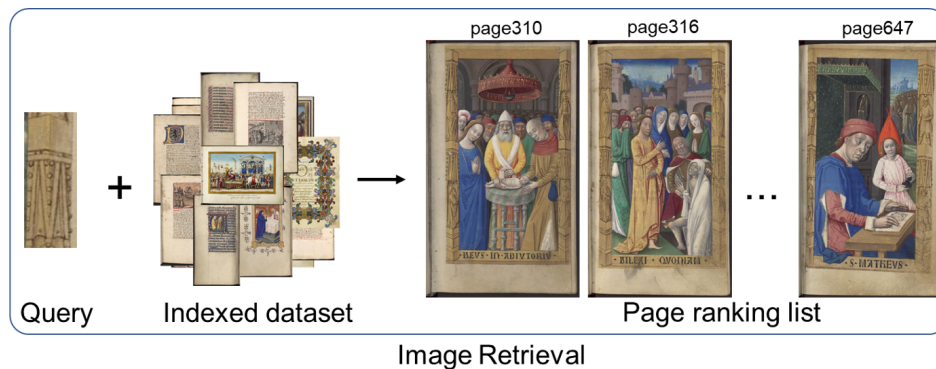


Figure 7 – Illustration of the Image Retrieval process, in which an image query is compared with all images in a database and a ranking is generated based on their similarities

Concerning the nature of queries, we can find in the literature works with a focus on words (GIOTIS et al., 2017), signatures (ZHU et al., 2008; SHARMA et al., 2018), and logos (ALAEI; ROY; PAL, 2016). According to Zhou, Li & Tian (2017), there are three main components in the CBIR: image representation, image organization, and image similarity measurement.

In general, CBIR systems perform two steps: offline and online. Figure 8 provides an overview of the traditional pipeline of these steps when a segmentation-based method is used. During the offline step, candidate objects are extracted from the dataset images and represented in a feature space, which is stored for the next step. During the online step, the user provides a query

that is compared with all images of the dataset. To perform this comparison, the query is represented in the same feature space, and a measure of similarity is used to compare it with the candidates of the search images. The results of the comparison are then ranked, and the best matches are presented to the user (GHOSH; AGRAWAL; MOTWANI, 2018). The offline step is performed only once and can be updated if new pages are added, while the online step is performed in each search. Therefore, it is recommended to include as much processing as possible in the offline step.

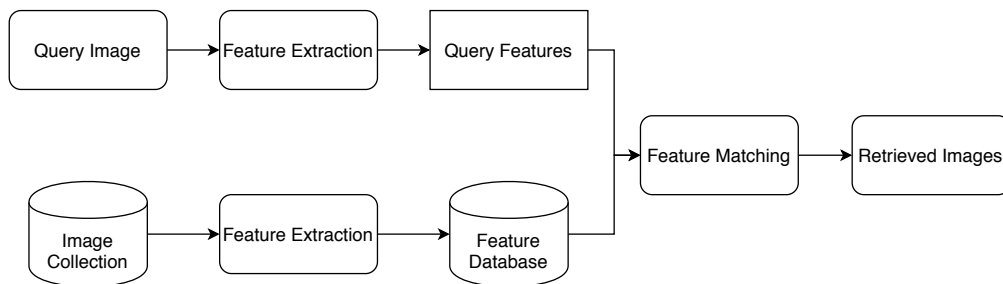


Figure 8 – Overview of a traditional CBIR process

There are two ways to perform CBIR: providing full images as a query or providing sub-images. When using sub-images, the process can be referred to as sub-image retrieval. This approach presents an additional challenge, which may require segmentation of the image to extract Regions of Interest (RoI) and enable the use of a traditional CBIR method. In some datasets, there is an additional challenge as the training set may not be available; thus, features generated by models trained in other datasets or domains must be employed.

When searching for sub-images, it is also possible to perform the task named Pattern Spotting (PS). In PS, the system must return the exact position of the objects visually similar to the query image searched, as shown in Figure 9. The concept of spotting methods emerged with word spotting, which aims to detect words in documents without prior knowledge of the context. One of the first word spotting works was presented in (MANMATHA; HAN; RISEMAN, 1996). As for CBIR, several studies have been published presenting optimizations for this area.

The Word Spotting and the Pattern Spotting present some additional challenges when using historical documents. In the survey presented by Ahmed, Al-Khatib & Mahmoud (2017), some of the challenges of Word Spotting in

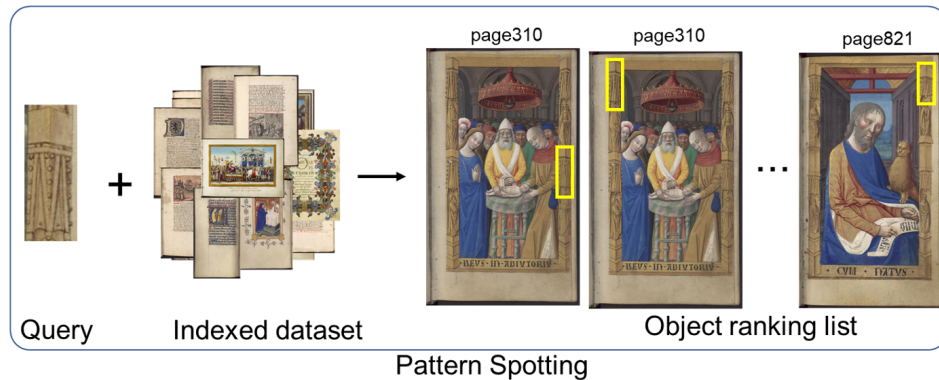


Figure 9 – Illustration of the Pattern Spotting process, in which an image query is compared to images in a database, and an object ranking list is generated based on the similarities

Handwritten documents are discussed. One of the points highlighted by the authors is the different writing styles. Handwritten documents have different authors and were written at different times, which generates varying writing styles across different documents. Another point raised is the indefinite number of words used. It is possible that new words will appear when a new document or page is added to the system, thus the matching techniques must be able to retrieve words that are out of the current vocabulary. A problem also presented in the handwritten historical documents is age, which causes problems of degradation, such as ink bleed. In addition to all these problems, there are some situations where the document capture generates distortions in the edges of the books, elevations in the illumination, misaligned aligned pages, or low resolution. Many of these problems are not exclusive to Word Spotting tasks, being common to all tasks that use ancient documents.

### 2.2.1 Preprocessing

The preprocessing step is essential for some IR and PS methods. In some approaches, this step is used to extract image candidates, which are used in the entire process and are crucial for a satisfactory performance of the method. In some methods, this step is more straightforward, being used to optimize the features extraction or the computational cost of the system.

An example where the preprocessing step is used to optimize the feature extraction is the work presented by Lu & Guo (1999). In this work, a method for background removal using color histogram is presented. The background removal allows to generate a global image feature only with the information



about the object presented in the image.

Removing the background also allows optimization in computational cost for sub-image task retrieval and pattern spotting. This occurs in En et al. (2016d), where a background removal method is used to reduce the number of candidates generated. In the work, the authors perform IR and PS tasks in handwritten historical documents and take advantage of the structure of this type of document. Initially, the image is binarized with the use of adaptive thresholding, then a strategy based on region growing paradigm was used. As the handwritten documents usually have their content centered on the page, the process starts at the center of the page. Pixel borders were iteratively added if the considered border zone has a number of background pixels below a fixed threshold of 5%.

In Úbeda et al. (2019), besides the background removal, all areas with texts are ignored. To identify the textual areas, a subset of the dataset consisting of 79 pages of handwritten historical documents was extracted and manually annotated with bounding boxes of NonText ROI. This subset is used as input for the RetinaNet architecture, and feature maps of different levels are extracted. Using the annotation performed, each feature vector referring to a receptive field of each pyramid level used receives a label (black, text, or non-text). These sets are then used to train a Random Forest classifier at each level. The new classifiers are then used on all pages, allowing the creation of Batches of embeddings with only the non-text predicted classes.

One of the limitations of the RetinaNet architecture used in Úbeda et al. (2019) is the need to use a fixed input size. Since the dataset used in the work has varying sizes, the authors chose to center the pages smaller than the input size in a black canvas and divide the larger images into sub-images with appropriate size. As the use of a black canvas influences part of the receptive fields of the images, the authors also chose to use a background page from a manuscript as a canvas to simulate the special textures of these types of pages.

Another necessary modification in many methods is the change of the color scale of the images. Some methods used for image representation do not accept three-dimensional vectors as input, requiring some operations to binarize the image or transform it to grayscale.

As seen in Ball, Srihari & Srinivasan (2006), when performing the sub-

image retrieval or spotting task, two strategies can be used in relation to the search space: segmentation-based and segmentation-free. In the first approach, candidate word regions are generated on a page, so the search is performed in a set of segmented word images. The advantage of this strategy is the possibility to use most traditional CBIR methods because each candidate can be treated as a new image. The second approach is to scan the image on the page, without the need to define segments. The use of a segmentation-free strategy attempts to perform spotting and segmentation concurrently. In the work of Ball, Srihari & Srinivasan (2006), these two strategies were compared for the word spotting task in handwritten Arabic documents. The authors observed that in the case tested, the use of a segmentation-free strategy yielded better results but needed more time.

The characteristics of historical documents can make segmentation methods do not work correctly. Zagoris, Pratikakis & Gatos (2017) also compare a segmentation-based method and one segmentation-free for word spotting in handwritten documents. In this work, it is pointed out that the selection of the segmentation-based strategy is preferred when the layout is simple enough to correctly segment the words while the segmentation-free strategy performs better when there is considerable degradation on the document.

One of the most popular segmentation methods in the literature is the Binarized Normed Gradients (BING) (CHENG et al., 2014). This method was used in the framework proposed by En et al. (2016c) for the task of Pattern Spotting in historical document images. The BING initially resizes the image using a set of pre-defined sizes. Using the new resized images, a sliding window mechanism is applied to produce 8x8-size subwindows. As the subwindows were created in the resized images, they represent different sizes and ratios in the original image. Each of these subwindows is then scored by a linear model, and the non-maximum suppression (NMS) is used to remove the overlapping regions and thus reduce the total number of regions. Finally, a re-ranking is performed on the remaining regions using another linear model based on SVM. The re-ranking is performed with information about the size of each window.

Another method established in the literature is the selective search (UIJLINGS et al., 2013). This method is used with historical documents in the framework presented in Wiggers et al. (2018) for the sub-image retrieval task and in Wiggers et al. (2019a) for pattern spotting. The selective search is

based on the hierarchical grouping algorithm. This method starts with the graph-based segmentation method presented in (FELZENSZWALB; HUTTENLOCHER, 2004). The segmentation method used is adjusted to return over-segmented results, so more regions are created. After applying this method, an algorithm iteratively groups regions together using the similarities between all the neighboring regions. The selective search uses four similarity measures based on color, texture, size, and shape compatibility. The creation of groups is performed several times, until the whole image becomes a single region. In each iteration of this algorithm, bounding boxes corresponding to segmented parts are created and added to a list, which contains all regions generated in all loops. In Gómez & Karatzas (2017), a modified version of the selective search focused on Word Spotting in the wild is presented.

### 2.2.2 Image representation

The key problem of CBIR is how to efficiently measure the similarity between images. As objects have various changes or transformations, it is impractical to perform a direct comparison at the pixel level. Therefore, the solution for this problem was to create fixed-sized vectors that represent the visual features of the images (ZHOU; LI; TIAN, 2017). Over time, several strategies have been developed for the creation of these vectors. The first works in this area were based on hand-crafted global features, while some current approaches allow the extraction of semantic information using deep learning strategies.

A representation strategy commonly used is the color information. Wengert, Douze & Jégou (2011) investigates the use of this strategy in the CBIR task. In this work, the authors present a color signature generation procedure inspired by the bag-of-words framework, referred to as bag-of-colors. A codebook is learned in a set of real-world images, defining a color set that is able to generalize all the colors of the images. With this codebook, a color histogram is created for all images using the new color set defined. The color of the codebook used is defined based on the Euclidean distance between the original color of each pixel of the image and the colors of the codebook. The histogram is then updated using the inverse document frequency and normalized with L1 vector normalization. This process allows the creation of more robust signatures and prevents the most frequent colors from dominating the

other colors in the final representation, as occurs in other color signature generation methods. In addition to the work presented by Wengert, Douze & Jégou (2011), several other studies in the literature present the use of colors as features to represent images, such as (SWAIN; BALLARD, 1991), (LIU; ZHANG; LU, 2008), (WANG; HUA, 2011), (WANG; ZHANG; YANG, 2014) and (GUO; PRASETYO; CHEN, 2014).

There are several methods that perform image feature extraction using local intensity to extract information about texture. Many of these methods create features based on the center and neighboring pixels' relationships. One example is the Local binary pattern (LBP), initially presented in (OJALA; PIETIKÄINEN; HARWOOD, 1996). As explained in Verma & Raman (2018), in the LBP method, the local information was extracted from each pixel based on the neighboring pixels. All pixels of the grayscale image were used as the center pixel once. That is, for each pixel of the image, a 3x3 block containing all its neighbors is considered. Each center pixel was subtracted by all neighboring pixels (Fig. 10 a and b), and a binary number was used to mark each neighboring pixel based on this subtraction (Fig. 10 c). These binary numbers were responsible for building a pattern that represents each central pixel. This binary pattern was then multiplied by some weights, formed by the sequence of the first eight numbers to the power of 2 (Fig. 10 d) and summed (Fig. 10 e), thus creating the local binary pattern value for that pixel.

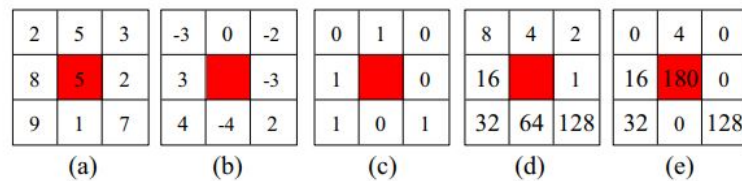


Figure 10 – Local Binary Pattern calculation. (VERMA; RAMAN, 2018)

Verma & Raman (2018) aims to optimize the operation of the LBP by proposing a complementary local feature descriptor that uses the local pixel intensity. The method, called local neighborhood difference pattern (LNDP), is a complement to the LBP, as it extracts relationships among neighboring pixels by comparing them mutually, while the LBP computes the relationships of neighboring pixels with the center pixel. Both LBP and the LNDP transform the local structure of each pixel of the image into a binary pattern. Verma

& Raman (2018) use these patterns, taken from all pixels in the image, to represent the image. The two methods were combined using the sum of the values obtained for each pixel. The Image Retrieval task was performed using the similarity between the map created for the query and the image. Besides LBP and LNDP, there are several other approaches in the literature, such as (WANG; CHEN; YUN, 2012), (LASMAR; BERTHOUMIEU, 2014) and (FADAEI; AMIRFATTAHI; AHMADZADEH, 2017). In Wang, Zhang & Yang (2014) and Liu et al. (2017), methods that integrate color and texture features are presented.

Another low-level feature used in several CBIR methods is shape. The work Zhang & Lu (2004) presents a review of the use of shape features in CBIR. As presented in Tian et al. (2013), the strategies for the extraction of shape features can be classified into two groups: region-based and contour-based. A contour-based method calculates shape features only from the boundary of the shape, while the region-based method extracts features from the entire region. Several works in the literature use shape features in CBIR methods, as (ZHANG; LU, 2002), (WANG; CHI; FENG, 2003), and (SOKIC; KONJICIJA, 2016). Liu et al. (2011) present a method capable of using color, texture, and shape simultaneously. The shape features also make this strategy interesting for some more specific retrieval tasks, as in Cao et al. (2011) and Wang, Kang & Li (2015), where methods of extracting shape features are used for the Sketch-based Image Search task.

As seen in Zhou, Li & Tian (2017), two works with Hand-Crafted Features methods presented a significant advance in the area of content-based visual retrieval. These works are Lowe (2004), which presented a local feature extraction approach called Scale Invariant Feature Transform (SIFT), and Sivic & Zisserman (2003), where it is presented a model capable of creating a compact representation of images, known as Bag-of-Visual-Words (BoW).

Several works in the literature present good results with the use of SIFT. This method has a great discriminative power to capture visual content in various domains. The SIFT is able to identify similar images even if there is a difference in scale, rotation, and changes in lighting (ZHOU; LI; TIAN, 2017). SIFT, initially presented in Lowe (2004), was developed to detect and describe local features in digital images. This method begins with the detection of key points at different image size scales. Additionally, the orientation of

each point is also identified, allowing for keypoints in the same position and scale, but with different directions, to be distinguished. After detection, a local descriptor is created for the detected point. This operation yields four pieces of information for each keypoint: location, scale, orientation, and descriptor. Finally, keypoints between two images are matched by identifying their nearest neighbors.

In Arandjelović & Zisserman (2012), the authors present RootSIFT for the generation of visual vocabulary and hard assignment of descriptors to visual words. The main difference between RootSIFT and the traditional SIFT is the square root kernel, which is employed to substitute the standard Euclidean distance for measuring the similarity between SIFT descriptors. After the extraction of the descriptors of the query with RootSIFT, they were represented with a BoW. Following selection of the best results with TF-IDF, the expansion of Discriminatory Query was used, wherein a linear SVM was utilized to discriminatively learn a weight vector for re-querying. This SVM learns weights for visual words that represent the query object. The learned weights were used to efficiently re-query, and spatial reranking was conducted again.

Another variation of SIFT is the binary SIFT, presented in Peker (2011). The binary SIFT aims to transform the result of the descriptor used by SIFT into a binary vector, thus allowing a fast visual matching and retrieval. The strategy of using binary values to speed up the operation of visual matching is explored by several works in the literature. One of the most important methods for creating binary descriptors in the literature is Robust Independent Elementary Features (BRIEF) (CALONDER et al., 2011). Some papers present variations of this method, such as the Binary Robust Invariant Scalable Keypoints (BRISK) (LEUTENEGGER; CHLI; SIEGWART, 2011), Oriented FAST and Rotated BRIEF (ORB) (RUBLEE et al., 2011) and Fast Retina Keypoint (FREAK) (ALAHY; ORTIZ; VANDERGHEYNST, 2012).

When features extracted with more than one representation are used, it is necessary to employ some strategy to aggregate those features. This operation is required to construct a fixed-size vector, which allows the application of a similarity metric and, consequently, the calculation of similarity between the query and the search images. Three feature aggregation methods stand out among the methods presented in the literature: Bag-of-Visual-Words (BoW),

the Vector of Locally Aggregated Descriptors (VLAD), and the Fisher Vector (FV).

The Bag-of-Visual-Words (BoW) method (SIVIC; ZISSERMAN, 2003) and (CSURKA et al., 2004) uses the same concept as the information retrieval method called bag-of-words. BoW counts features and creates a histogram with the values obtained, enabling the search for similar images or the prediction of the image category. The feature aggregation method known as VLAD, initially presented by (JÉGOU et al., 2010), uses a vector quantization strategy based on k-means and accumulates the quantization residues for features quantized to each visual word; the accumulated vectors are then concatenated into a single vector, which represents the entire image. The Fisher Vector (FV) (PERRONNIN et al., 2010) can be seen as an extension of BoW, differing in its ability to encode higher-order statistics, allowing better representation with fewer visual words, due to the use of the Fisher Kernel (JAAKKOLA TOMMI; HAUSSLER, 1999).

Besides the approaches that use handcrafted visual features, it is possible to use a feature extraction method that extracts the semantic features from the image. This can be done with learning-based features. Among the learned methods used for feature extraction, it is important to mention the probabilistic Latent Semantic Analysis (pLSA) (HOFMANN, 2001) and Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2003).

The use of Deep Learning methods has become popular in several computer vision tasks, including Content-Based Image Retrieval (CBIR). An example of CNN application in CBIR is the work of Tolias, Sivic & Jégou (2015), where CNN features are applied in filtering and re-ranking stages. In the filtering stage, the database images are ranked in terms of similarity to a query image. In the re-ranking stage, the best-ranked results are refined. The authors used pre-trained CNNs (AlexNet and VGG16) to extract feature maps without the fully connected layers. The queries and image maps were compared using cosine similarity in two ways: with the complete map and with map regions. To reorder the list of most similar images, the authors applied a technique called approximate max-pooling localization. This method consists of finding regions where the object can be located and locally refining the best ones with simple heuristics. After re-ranking, the positive images should be in the top-ranked positions. To further refine the results, the authors collected the five

top-ranked images and merged them with the query vector to compute their mean. The similarity to this mean vector was used to re-rank the top images.

One way to optimize the retrieval process time is through the use of binary features. As presented by (LIU et al., 2016), Deep Supervised Hashing allows the creation of binary vectors through the discretization of features generated by Convolutional Neural Networks (CNN). Similarly, other works in the literature seek to create binary hash codes to optimize the comparison between features, such as (WU et al., 2019) and (ZHU et al., 2016). In contrast, (LIN et al., 2016) proposes a deep neural network to learn binary descriptors in an unsupervised manner. This method involves a process to binarize the output of a pre-trained CNN, as well as a new loss function to optimize the network weights to the binarization process.

Several works have demonstrated that the use of pre-trained networks as feature extractors can generate superior results than hand-crafted features. However, context-specific data training is necessary to obtain the best possible results, but obtaining data for this can be difficult. An example of this is presented in Radenović, Tolias & Chum (2018). The authors observed that several works in literature use descriptors based on the activations of CNNs due to their compactness and efficient search capabilities. The main limitation of the use of these models is the need for a large amount of data to perform training or fine-tuning. This is a problem in many areas because there is not enough manually labeled data, and this annotation can require a considerable number of resources. To address this limitation, the authors propose a way to automatically annotate images to allow fine-tuning in CNN models.

Radenovic and colleagues applied a Structure-from-Motion (SfM) 3D reconstruction system to enable the use of unannotated image collections to fine-tune a CNN model. Initially, a BoW-based retrieval image system was used to collect visually similar objects/landmarks. These images were then used as input for the SfM 3D reconstruction system, and the unique images generated from some of the 3D models were used to assemble the dataset used for fine-tuning. The created dataset was composed of tuples containing one query, one positive, and five negative images. In many situations, it is not possible to apply 3D reconstruction techniques, requiring the use of images in a Self-Supervised way. In Siradjuddin, Wardana & Sophan (2019), a method with Autoencoders using CNN for feature extraction was presented.



In (MISHCHUK et al., 2017), it is presented that some hand-crafted methods of local descriptors (such as SIFT) still have better results than learned descriptors for some areas. The main reason for this is the fact that the current local datasets are not large and diverse enough to allow learning of high quality and widely applicable descriptors. To address this issue, the authors propose a new learning loss to the L2Net CNN architecture (TIAN; FAN; WU, 2017). This learning objective mimics the SIFT matching criterion. Initially, a batch with all the anchor and positive pairs is formed. These pairs are extracted from the same 3D point, and a 3D point originates only one pair. With the pairs formed, each of the images is passed through the architecture, and their features are extracted and stored. With the features, the L2 pairwise distance matrix is calculated. With the matrix, the closest non-matching neighbor to each pair is collected and adopted for training. This allows the creation of groups with three examples, the anchor, the positive example, and the negative example. The goal is to minimize the distance between the matching descriptor and the closest non-matching descriptor. The main advantage of this method for choosing the negative example is the assurance that it causes some degree of confusion with the positive example or with the anchor. Another advantage is the reduction in total time compared to the traditional triplet, where three images are provided as input.

Although many efforts have been presented in the literature to train with few data or without the use of labels, some problems still require trained networks in different domains. An example of this is a recent challenge where CBIR is conducted without prior knowledge of the queries that will be used. Wiggers et al. (2019b) present two different approaches using Deep Learning. One of the approaches presents a method using a pre-trained CNN as a feature extractor, and the other uses a pre-trained Siamese Convolutional Neural Network (SCNN) for feature extraction and similarity estimation.

In the strategy proposed by (WIGGERS et al., 2019b), a segmentation method is used to enable sub-image retrieval. In contrast, (ÚBEDA et al., 2019) and (ÚBEDA et al., 2020) present a segmentation-free strategy using the RetinaNet architecture. To achieve this, CNN was used as a tool for local feature extraction, utilizing feature maps at multiple scales. In this work, the feature maps were used as a local feature vector map, where each position of the map is associated with a feature vector corresponding to a receptive

field. The use of multiple scales allows for the extraction of receptive fields of different sizes in a single forward pass.

### 2.2.3 Similarity measures

To create an efficient Content-Based Image Retrieval (CBIR) method, it is necessary to compare the query and images effectively. This comparison requires robust image features and a similarity measure capable of utilizing the maximum of the existing features. Image feature comparison can be made with similarity and dissimilarity measures. The selection of the ideal metric should be based on the features used in image representation. Several works have presented comparisons between various metrics using different types of features, such as in (VADIVEL; MAJUMDAR; SURAL, 2003), (NAIK et al., 2009), and (PATIL; TALBAR, 2010).

Considering  $x$  and  $y$  as two  $d$ -dimensional feature vectors of a search image and a query image, respectively, some of the most commonly used metrics in the literature (PATIL; TALBAR, 2010; VADIVEL; MAJUMDAR; SURAL, 2003) can be defined as follows:

- Euclidean distance

$$d_E(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (2.2)$$

- Manhattan distance

$$d_{MAN}(x, y) = \sum_{i=1}^d |x_i - y_i| \quad (2.3)$$

- Canberra distance

$$d_C(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (2.4)$$

- Bray Curtis distance

$$d_{BC}(x, y) = \sum_{i=1}^d \frac{|x_i - y_i|}{x_i + y_i} \quad (2.5)$$

- Square Chord distance

$$d_{SC}(x, y) = \sum_{i=1}^d (\sqrt{x_i} - \sqrt{y_i})^2 \quad (2.6)$$

- Square Chi-Squared distance

$$d_{CHI}(x, y) = \sum_{i=1}^d \frac{(x_i - y_i)^2}{x_i + y_i} \quad (2.7)$$

- Vector Cosine Angle Distance

$$d_{VCAD}(x, y) = \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2} \sqrt{\sum_{i=1}^d y_i^2}} \quad (2.8)$$

The metrics presented aim to compare two d-dimensional vectors, but in some situations, it is desired to compare matrices. If the matrices to be compared have the same size, all the metrics can be adapted for this type of comparison. If the matrices have different sizes, a feature aggregation strategy, such as Bag-of-Words (BoW), Vector of Locally Aggregated Descriptors (VLAD), or Fisher Vector (FV), can be used to transform the data into a d-dimensional vector for comparison.

Another alternative when dealing with matrices of different sizes is the comparison using cross-correlation. Like the metrics described previously, this approach aims to measure the similarity between the search image and the query image. However, cross-correlation uses image feature matrices as input, rather than just a vector. This type of representation allows for the comparison of local characteristics of different parts of the images individually. This approach and its variations are widely used in template matching works (SARVAIYA; PATNAIK; BOMBAYWALA, 2009), (BRIECHLE; HANEBECK, 2001).

#### 2.2.4 Post-processing

After comparing the query and the images, it is necessary to create a ranking of the best results to return to the user. In some methods, changes can be made before or during this step in order to optimize the result of the comparison. In En et al. (2016a), a method for pattern spotting is presented that uses a post-processing step after the creation of the ranking with the most similar regions. In this step, two changes are made. Initially, a process is performed to join the regions with overlap with other ranked windows. After this union, a template matching method is used to generate a more precise location within the union of these retained regions.

Another strategy for removing the overlap windows in the spotting task results is Non-Maximum Suppression (NMS). NMS consists of calculating the Intersection over Union (IoU) between all regions found in the image. If the IoU is higher than a certain threshold, the region with the highest confidence is kept and all others are eliminated. NMS is popular in object detection methods and can also be used for the pattern spotting task, as demonstrated in (EN et al., 2016b), (ÚBEDA et al., 2020), and word spotting, as in (ROTHACKER; RUSINOL; FINK, 2013).

Another necessary operation, in some cases, is the calculation of the coordinates of the correct window. This operation may be necessary when the image is cropped, resized, or added to a background canvas. In some cases, it is also possible to modify the ranking of the results to optimize the final results.

In Park, Baek & Lee (2005), a re-ranking strategy is used to optimize retrieval effectiveness by leveraging the relationships between retrieved results via clustering. In Tolia, Sicre & Jégou (2015), the re-ranking is performed using its own similarity score. The re-ranking stage allows the best results to be placed in the top positions of the ranking, so the authors use the five top-ranked images and perform a query expansion process, wherein the five images are merged, and a re-ranking is performed again.

## 2.2.5 Datasets of Historical Documents Images

It is important to emphasize the essential role of datasets in conducting research and experiments in IR and PS. Datasets provide a solid foundation for comparing and testing models and strategies. They not only enable the evaluation of the effectiveness and robustness of proposed methods but also allow for result replication and validation, thereby contributing to the advancement of knowledge and the development of better approaches. This section introduces three fundamental datasets for this research: DocExplore, Horae, and Tobacco. These datasets play a crucial role in the domain of historical document analysis, distinguished by their unique characteristics, and they are valuable tools for the fields of IR and PS.

### 2.2.5.1 DocExplore

DocExplore was introduced in En et al. (2016a) and was developed by the academic project also named DocExplore<sup>1</sup>. The documents that compose the DocExplore dataset were collected from the Municipal Library of Rouen, France. Some samples of images from this dataset can be seen in Figure 11.

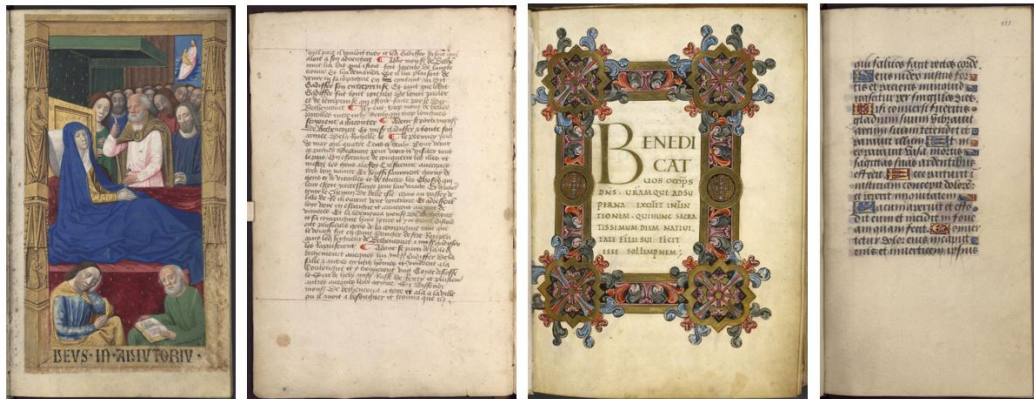


Figure 11 – Sample images from the DocExplore dataset

DocExplore is composed of manuscripts dated from the 10th to 16th centuries. A total of 1500 images comprise the dataset. Most of the dataset images were rescaled to a maximum size of 1024 pixels on each dimension. The images were manually annotated with 35 graphical objects. A total of 1464 objects were annotated; however, due to some differences between the object and others of the same category some of them were considered as junk, resulting in 1447 objects. The 35 object categories used in DocExplore are shown in Figure 12, with the total of objects of each category in parentheses, followed by the number of junks.

Even with the removal of junk from the dataset, there is still a large intra-category variability of the annotated objects. These variations can occur in relation to color, scale, ratio, and the proper object. Figure 13 shows some of the intra-category variations in DocExplore.

For better visualization of the existing variation between the queries, Figure 14 presents a box plot related to the number of pixels of the queries. The values for the number of pixels were obtained by multiplying the height

<sup>1</sup> <http://www.docexplore.eu/>

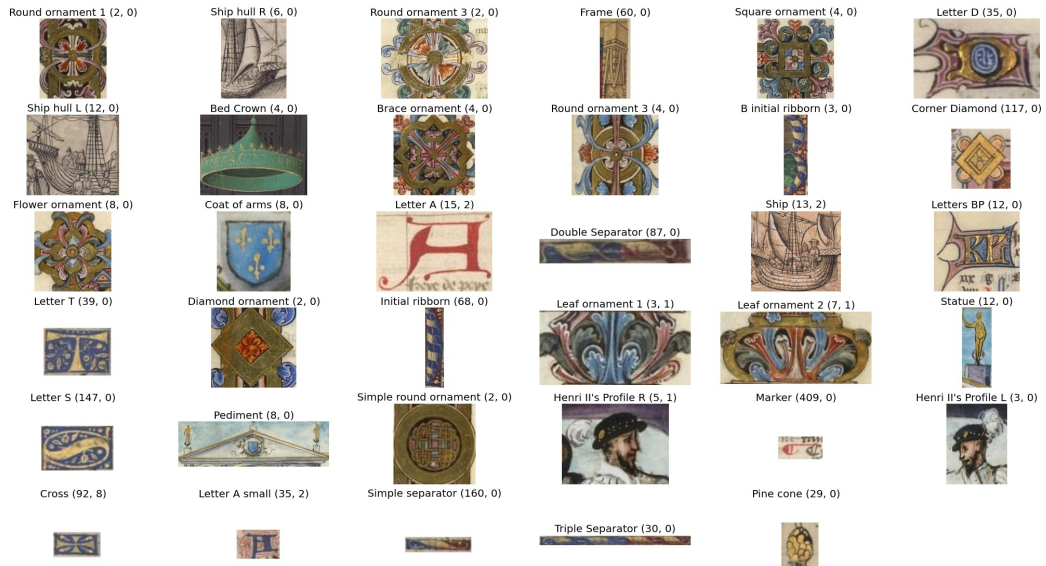


Figure 12 – Categories of objects annotated in DocExplore dataset. (EN et al., 2016a)

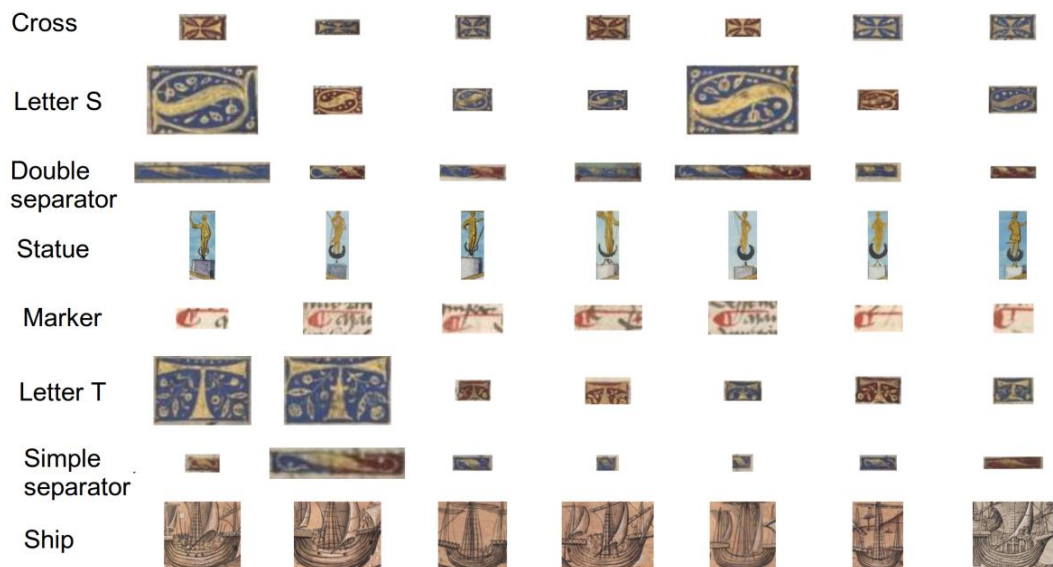


Figure 13 – Intra-category variability of the annotated objects. (EN et al., 2016a)

and width of the image query ( $h \times w$ ). As can be observed in this figure, most of the DocExplore categories have less than 50,000 pixels. The Pediment category presents the greatest difference in relation to the others, being the category with the largest images of this dataset. The categories with the largest size variation between their images are the Bed Crown, Pediment, and the three types of Ship.

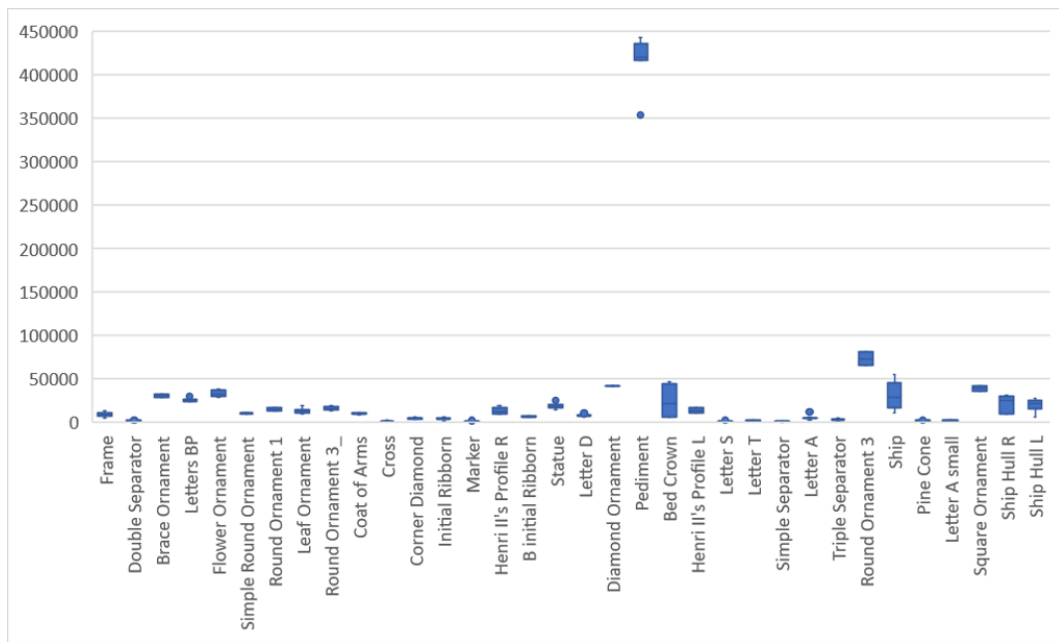


Figure 14 – Box plot related to the size of DocExplore queries

In addition to the variation in the size of some queries, there are differences in relation to the ratio. This difference is evident in the box plot of Figure 15, which shows that most of the DocExplore categories have an aspect ratio variation close to one, indicating that these categories are close to the format of a square. However, the categories Frame, Initial Ribborn, B Initial Ribborn, and Statue have a higher aspect ratio, indicating more rectangular images.

### 2.2.5.2 Horae

The Horae dataset, introduced in Boillet et al. (2019), consists of pages extracted from books of hours, handwritten prayerbooks created in the late Middle Ages. It consists of 557 images with 797 pages, as illustrated in Fig. 16. The images in the dataset can be colored or grayscale and have different

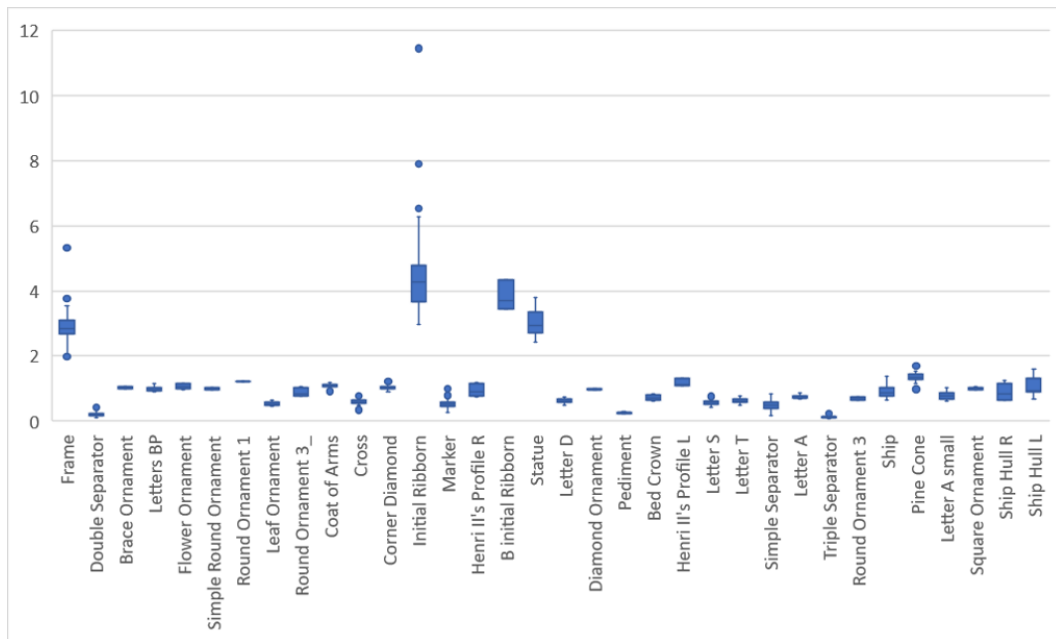


Figure 15 – Box plot related to the aspect ratio of DocExplore queries

resolutions. The Horae dataset is originally annotated in layout components, providing a comprehensive understanding of its structural elements. However, beyond its inherent value for component-based analysis, the dataset’s images themselves hold significant merit for qualitative examination.



Figure 16 – Sample images from the Horae dataset.



## 2.2.5.3 Tobacco800

The Tobacco800 dataset, presented in Zhu & Doermann (2007), is a subset of the scanned document collection Complex Document Image Processing (CDIP) (LEWIS et al., 2006). This subset consists of 1290 grayscale and binary documents from tobacco companies, with annotations on logos and signatures. Some samples of images from this dataset are presented in Figure 17.

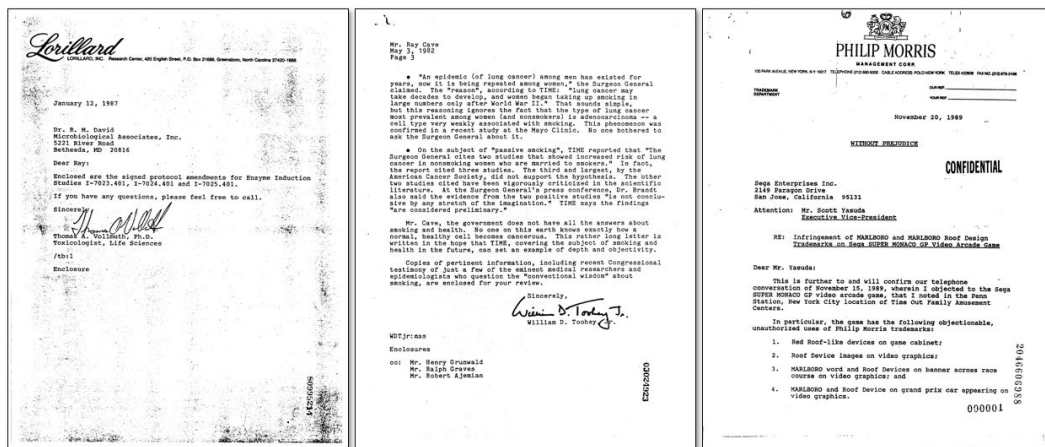


Figure 17 – Sample images from the Tobacco800 dataset

In the context of logo retrieval, our focus has been exclusively on categories featuring two or more occurrences. This targeted approach has led to the identification of 21 categories, encompassing a total of 416 queries. The categories and their respective number of samples are presented in Figure 18.

The queries of the Tobacco800 dataset have a large intra-category variability, as shown in Figure 19, which displays a box plot of the size of the queries. The number of pixels was obtained by multiplying the height and width of the image query ( $h \cdot w$ ). The queries in the Tobacco800 dataset have a more significant size variation than those in the DocExplore dataset. This problem is mainly attributed to the use of different tools to capture documents, which generates images of different sizes.

The variation of the intra-category aspect ratio ( $h/w$ ) is shown in the box plot of Figure 20. In contrast to DocExplore, the difference between the aspect ratio of the queries of Tobacco800 is small. The main reason for this is

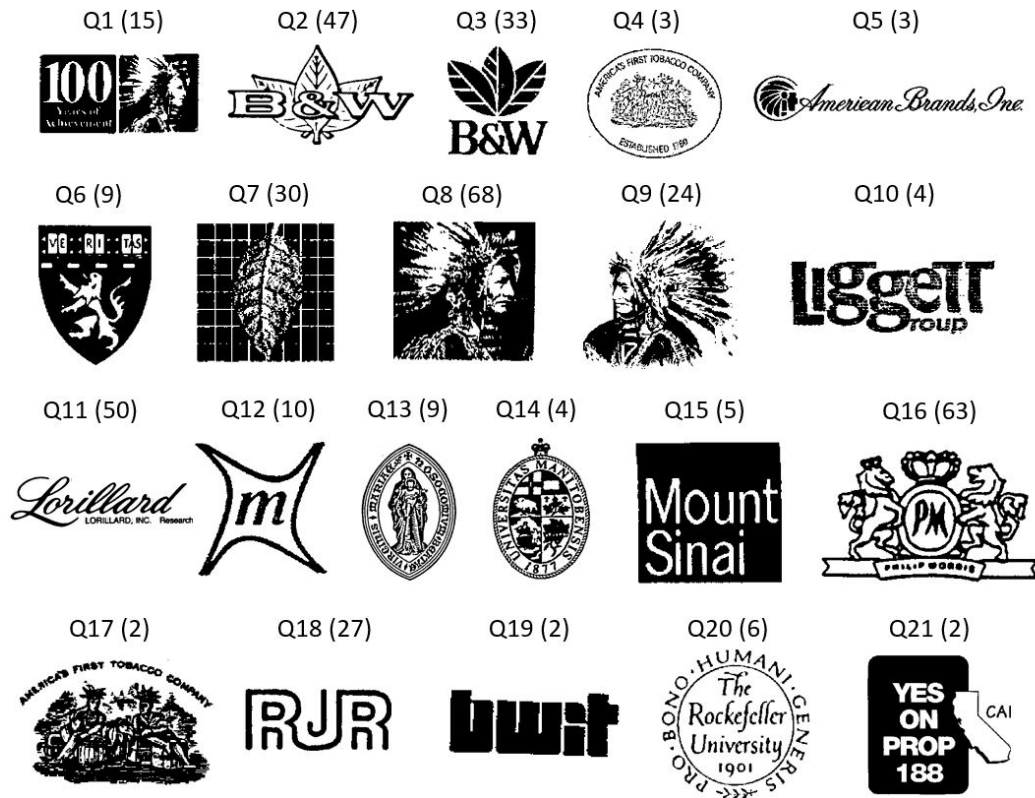


Figure 18 – Categories of objects in the Tobacco800 dataset. The number in parenthesis represents the amount of occurrences

that logos are used as a query.

## 2.2.6 Image Retrieval and Pattern Spotting in Historical Documents

Several works in the literature present methods for the realization of CBIR in different areas. In this work, we are interested in the methods of CBIR and PS for ancient documents. Table 1 presents a summary of the main works in this field, comparing different image representation methods and similarity measures utilized for pattern spotting in historical documents. The table provides an overview of each approach and the datasets used to evaluate the proposed methods.

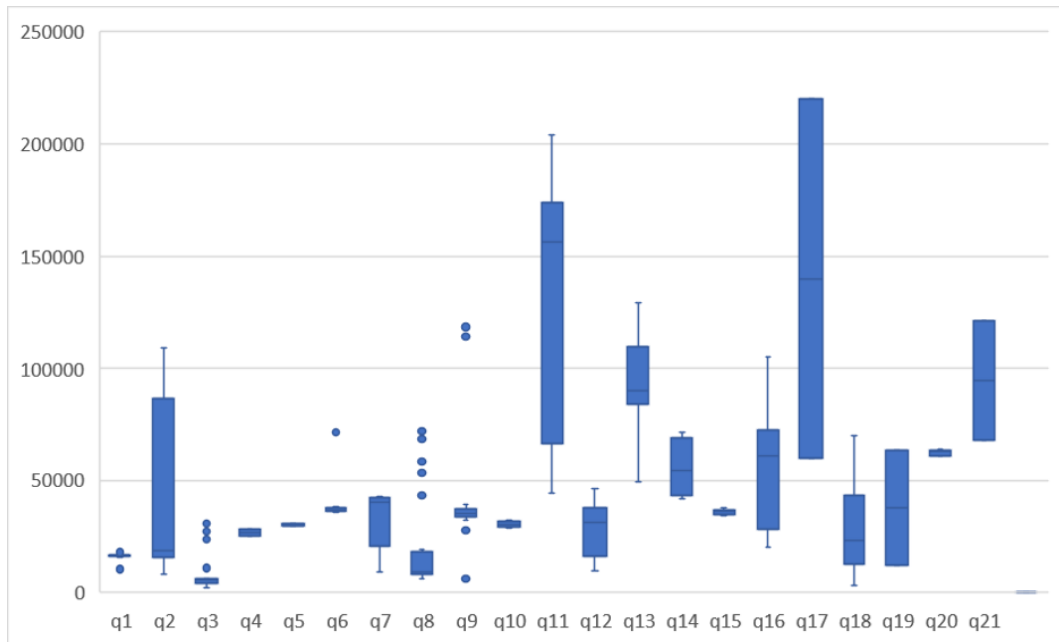


Figure 19 – Box plot related to the size of Tobacco800 queries

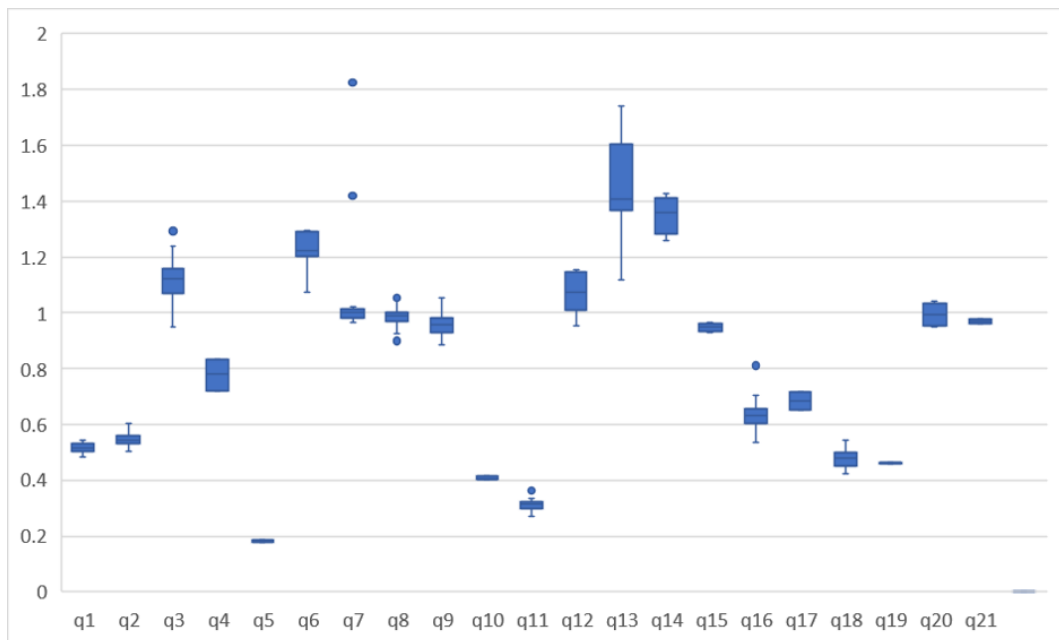


Figure 20 – Box plot related to the aspect ratio of Tobacco800 queries

Table 1 – Summary of the main works for Image Retrieval and Pattern Spotting in ancient documents

| Author                            | Search Space                          | Image Representation                           | Similarity Measure                      | Datasets Used                         |
|-----------------------------------|---------------------------------------|--|---|---------------------------------------|
| (ZHU; KEOGH, 2010)                | Segmentation-free                     | Color Histograms                               | Histogram Intersection                  | Proprietary Dataset                   |
| (RAKTHANMANON; ZHU; KEOGH, 2011)  | Segmentation-free                     | Generalized Hough Transform                    | GHT distance                            | Proprietary Dataset                   |
| (DOVGALECS et al., 2013)          | Segmentation-free                     | BOVW   | Chi-square distance                     | Proprietary Dataset                   |
| (EN et al., 2016a)                | Segmentation-based (BING)             | DenseSIFT with VLAD                            | Cosine distance                         | DocExplore                            |
| (WIGGERS et al., 2019b)           | Segmentation-based (selective search) | SCNN   | Cosine distance                         | DocExplore and Tobacco                |
| (ÚBEDA et al., 2020)              | Segmentation-free                     | CNN  | Cosine distance                         | DocExplore                            |
| (MOHAMMED; MÄRGNER; CIOTTI, 2021) | Segmentation-free                     | FAST keypoint detector with adaptive threshold | NBNN                                    | EFEO, AMADI LontarSet, and DocExplore |
| (DIAS et al., 2022)               | Segmentation-based (selective search) | CNN  | Euclidean distance and Hamming distance | DocExplore                            |

Although this area is important for historians and researchers, few reports are found in the literature when compared to other domains. However, as shown in Table 1, recent works have introduced novel methods for CBIR and PS in ancient documents, using different image representation methods and similarity measures. For instance, in Zhu & Keogh (2010), one of the first methods focused on pattern spotting in ancient documents is presented. In their work, images from a dataset of manuscripts dating back to the fifteenth century are used. The authors present a method with the use of color histograms for the representation of the candidates and the number of pixels in common as a distance measurement. In Rakthanmanon, Zhu & Keogh (2011), information about the shape of binary images is used for the PS task in datasets of scientific and cultural manuscripts dating back to the fourteenth century.

Although the works of Zhu & Keogh (2010) and Rakthanmanon, Zhu & Keogh (2011) make an important contribution to the PS area, the datasets used were annotated with objects located in a uniform background, making the segmentation of these regions simple for the segmentation methods. A dataset with a higher difficulty level is used in Dovgalecs et al. (2013), where a PS method belonging to the system presented in Tranouez et al. (2012) is presented. In the work of Dovgalecs and colleagues, a system for pattern spotting and word spotting is proposed. The evaluation of the PS method was performed qualitatively in a dataset of its own. The proposed method uses Dense SIFT to extract features from the whole image, making it segmentation-free.

A segmentation-based method for CBIR and PS in medieval manuscripts is presented by (EN et al., 2016a). In this method, regions of interest are extracted using Binarized Normed Gradients (BING). The features of these regions and the query are then extracted using DenseSIFT and aggregated using Vector of Locally Aggregated Descriptors (VLAD) to obtain the final vector. The feature vectors obtained are compared using cosine distance. After this comparison, a post-processing step is used to find the correct position of the objects for the spotting task.

An interesting learning-free PS method based on hand-crafted features is presented by Mohammed et al. (MOHAMMED; MÄRGNER; CIOTTI, 2021). The authors use the FAST keypoint detector with the adaptive thresh-

old PCK for obtaining features from the contours found in the images. After normalization on the features, the Normalised Local Naive-Bayes Nearest Neighbour (NBNN) algorithm is used to compute the distance vector.

As in En et al. (2016a) and Mohammed, Märgner & Ciotti (2021), early image retrieval methods used hand-crafted features to represent images. However, in recent years, the popularity of these methods has declined, while CNNs have been established in several areas. The main limitation of the early CNN strategies presented in the literature was the need for a relatively large dataset for training. However, several works in the literature have aimed to find solutions to this limitation, allowing for the fine-tuning of networks with few examples or training without the need for labels (ZHENG; YANG; TIAN, 2017; LECUN; BENGIO; HINTON, 2015).

Although many efforts have been presented in the literature to train CNNs with few data or without the use of labels, some problems still require networks trained in different domains. One case where this occurs is in Wiggers et al. (2019b), where two strategies with the use of image segmentation and Deep Learning are presented for CBIR in ancient documents. One of the approaches is presented as a method using a CNN as a feature extractor, and the other approach uses a pre-trained Siamese Convolutional Neural Network (SCNN) for feature extraction and similarity estimation for CBIR and PS.

Dias et al. (2022) presents a method using SCNN. The authors propose a strategy to filter the regions returned by the selective search, thus reducing the number of comparisons performed in the online stage. In addition, a binary representation method is presented, allowing to optimize both the offline and online phases.

In contrast to methods where a segmentation strategy is used to detect possible objects, some methods are segmentation-free, allowing the comparison and location of objects without the need to create image hypotheses. In Úbeda et al. (2019) and Úbeda et al. (2020), a strategy is presented using RetinaNet network architecture in a segmentation-free approach. For this, CNN was used for local feature extraction. This was done using feature maps at multiple scales. In these works, the feature maps were used as a local feature vector map, where each position of this map is associated with a feature vector referring to a receptive field. The use of multiple scales allows receptive fields of different

sizes to be extracted in a single forward pass. The authors also applied a NonText Classifier step, which was responsible for eliminating all regions with text and keeping only the graphic objects in the images.

As in approaches adopted by Wiggers et al. (2019a) and Úbeda et al. (2020), we perform feature extraction with a network trained in a different domain than the target one. However, one of the novelties presented in our work is the use of intermediate layers of the network architecture. This choice is motivated by the solid structural representation existing in the initial and intermediate layers of a CNN architecture. The main idea is to use several weak local representations to create a strong matrix of features representing the whole image. Unlike the vector comparison used by traditional CBIR and PS methods, we compare the feature maps of different sizes using cross-correlation. This strategy allows us to compare queries and pages with different sizes and formats.

## 2.3 Final Considerations

In this Chapter, we have presented the main works related to feature extraction using FCNs and approaches to image retrieval and pattern spotting. While there are several approaches for feature extraction that can be applied in IR and PS methods, many of them require specific training for each application. Among the approaches discussed, we highlight the use of FCNs, as they allow for the representation of images through feature maps.

Furthermore, we have presented several methods in the literature for IR and PS tasks in historical documents and discussed their characteristics and limitations. Despite recent advances, there are still challenges to be faced in this area. It is important to emphasize that the works presented in this Chapter do not provide direct confirmation or rejection of any of the hypotheses proposed in our research. This is due to the fact that none of the presented methods employ a combination of FCN and cross-correlation without undergoing training using the specific images related to the addressed problem.

In the next Chapter, we will present the proposed methods for IR and PS tasks in ancient documents. We will also provide a detailed description of the evaluation protocol used to validate the effectiveness of the proposed

methods on different datasets.



## 3 Proposed Method

This Chapter presents new methods for Pattern Spotting and Image Retrieval tasks in ancient documents. It combines FCN, transfer learning, cross-correlation, and heatmaps to perform IR and PS tasks without training in the target domain. Additionally, a binary variation of the method is presented, with a feature conversion and XOR cross-correlation approach. Finally, the Chapter presents the test protocol used during the execution of the experiments.

### 3.1 Proposed Method for Image Retrieval and Pattern Spotting

The method proposed in this work follows the phases used in traditional CBIR methods: the online and the offline phases. In the offline phase, the features of the search images are extracted and stored. During the online phase, the features of the query are extracted and compared with the stored features from the search images. After this comparison, a rank list is created based on the calculated similarities. It is important to note that the presented method is segmentation-free. This choice was made to reduce the possibility of overlooking certain objects in a preliminary stage, irrespective of their characteristics. Figure 21 presents an overview of the whole method.

#### 3.1.1 Offline phase

During the offline phase, an FCN extracts a feature map from each document image. The feature maps are then submitted to a PCA for decorrelation and dimensionality reduction. After being normalized using l2-normalization, the transformed feature maps are stored and made available for the online phase. The following two subsections describe in detail the feature extraction and processing performed in the offline phase.

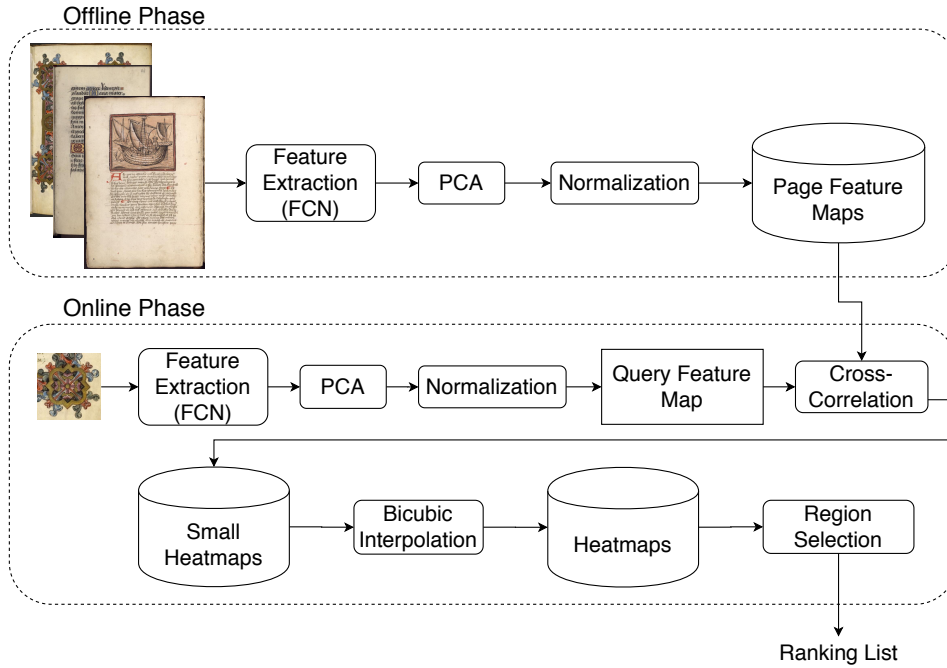


Figure 21 – Overview of the proposed framework for image retrieval and pattern spotting.

### 3.1.1.1 Adaptive Batch

As a fully convolutional version of the network is used for the feature extraction, this network can be fed with images of different sizes. In order to optimize the execution time, it is possible to create image batches, which allows us to simultaneously process multiple images. It is essential to note that, in overcoming this approach, the methodology employed ensures the preservation of the entire image without cropping or segmenting, despite the challenge posed by the requirement for a fixed-size matrix, given the diverse sizes of the historical document pages.

One of the most popular approaches to creating the batches is to resize the images to a predefined size. However, the resizing is not feasible for PS task because the information regarding the objects' size and location in the pages is essential. One approach to avoid losing this information is to add the images into a canvas with a fixed size. Although this option allows access to the objects' spatial location in the generated feature maps, a considerable amount of processing is spent in situations where the images have a significant variation in size, thus reducing system performance. We used a batch with an adaptive canvas to avoid this unnecessary processing.

The adaptive batch used has a different size for each group of images. For the creation of these groups, all the images in the dataset are sorted by their width. The width is used to optimize situations where two pages are presented side by side in a single image, as in an open book. After sorting, groups with  $n$  images are created. The value of  $n$  must be defined by the amount of memory available for feature extraction and does not influence the final ranking of the method. For each group, the largest width and height are used as the size of that batch. With this, a black canvas is created, and the images are added from the top left corner.

### 3.1.1.2 Feature Extraction

The network used to extract features should be able to handle images with various sizes and aspect ratios. However, it is expected that problems may occur when dealing with large or small images. In fact, the processing of large images may induce memory consumption and processing time issues. Such a kind of problem can be solved in two ways: reducing the image size or splitting the image into several parts. Small images may be smaller than the minimum size accepted by the architecture used. This problem can be solved by adding the image to a canvas or increasing the image size using an interpolation method. The definition of the minimum and maximum size required is directly related to the architecture used and the computer resources available. The minimum size is necessary to avoid access to information that does not exist, as when trying to perform a convolution with a mask larger than the feature map without the use of padding. An example of minimum size is the Fully-Convolutional version of the VGG16 architecture (SIMONYAN; ZISSERMAN, 2014) without modifications, which generates an error when an image with a height or width smaller or equal to 15 is used. In contrast, in the Fully-Convolutional version of the ResNet156 architecture (HE et al., 2016), this error does not occur, even if a 1x1 image is used as input. This difference occurs due to the use of padding in the pooling layers of the architecture.

Although the use of architecture with padding in the pooling layers solves the too small images problem, another one is created. Padding adds extra information to the feature maps and can generate a completely different feature vector for the same object if they have a few pixels of difference. Thus, comparing a small object present in an image query with the same item on

a page can return a low similarity. We therefore cannot use padding to offset this small image problem.

Nonetheless, it is important to highlight a characteristic of CNNs architectures that can avoid or minimize this problem. In a traditional architecture, shallow layers generate outputs that contain information about the structure of the image content. In contrast, deeper layers have outputs with stronger representation power and capture the semantic information (LONG; SHELHAMER; DARRELL, 2015). However, this is not necessarily the case when the network is not trained. In general, shallow and intermediate layers are more generic, while deep layers have a strong influence from the classes used in training for the final, fully-connected decision layer. As the framework presented in this work is independent of training, it is possible that a model trained in another domain is used. In those cases, it is expected that the deeper layers will return worse features than the intermediate layers, regardless of the image size. In addition, intermediate layers usually have fewer stride layers, which decreases the minimum size and allows the use of smaller images without the need for modifications.

The need to use a fully convolutional architecture without padding, combined with the advantages of using intermediate layers, is essential for choosing the architecture that will be used. As intermediate layers will be used, it is interesting to consider an architecture with few layers, which allows a faster feature extraction. Among the CNN architectures established in the literature, it is important to mention the VGG16 (SIMONYAN; ZISSERMAN, 2014) for having few layers and not using padding in the pooling layers. These characteristics led us to choose this architecture trained in dataset ImageNet to test the proposed method. The Fig. 22 shows this architecture.

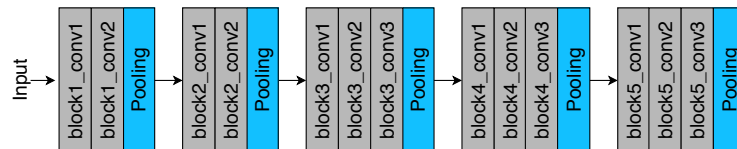


Figure 22 – Fully convolutional version of VGG16 architecture.

### 3.1.1.3 Principal Component Analysis

After the extraction of the features, a principal component analysis (PCA) operation is performed. The PCA aims to optimize the extracted features and provide reduced storage space. The PCA is trained with the feature vectors of a subset of the dataset and applied to all images.

Two parameters need to be defined in this operation: the number of images used in the training subset and the number of components to extract. We intend to define these values by performing experiments. The choice of the best value should also consider the available computing resources.

### 3.1.1.4 Features Normalization

Regardless of the architecture and how deep the network is, the output of a fully convolutional network is a 3-dimensional feature map, which can be considered as a 2-dimensional map containing several feature vectors. Each feature vector represents a part of the image, referred to as a receptive field. We perform a principal component analysis (PCA) to create vectors with independent elements, removing the correlation between them. We analyse the feature vectors of only a subset of the dataset, and then apply it to all images. This tuning is conducted for each tested dataset without the use of any information about the queries.

As our goal is to compare the vectors for creating a ranking, a normalization step is necessary. This occurs because all the data must be on a common scale to allow a direct comparison. For this normalization, we apply l2-normalization on each feature vector of each feature map. Finally, all normalized matrices are stored to be used in the online phase.

## 3.1.2 Online phase

Now the features that compose our dataset images are extracted, we can proceed to the actual querying in the online phase. During this phase, the query image features are compared with the features of the page images.

### 3.1.2.1 Features Extraction and Processing

To allow a correct comparison, the same procedures of feature extraction and processing must be performed on the query image, as those applied during the offline phase to the page images to be indexed. Initially, the feature maps are extracted using the query as input. After the feature map extraction, PCA reduction and l2-normalization are applied to all the feature vectors extracted from the query. The same architecture with the same weights and PCA parameters, as those used during the offline phase, are used during this step.

As in the offline step, the query is used in its original size. However, a treatment must be performed to avoid errors with small queries. This treatment consists in resizing images smaller than the minimum size accepted by the network used. In our implementation, we have opted for the use of bicubic interpolation for resizing.

### 3.1.2.2 Cross-Correlation

At this stage, the features of the query image have been extracted and treated in the same way than those of the page images of the collection to be indexed for future search, allowing the comparison between them. For this purpose, a cross-correlation operation is now used to compare the query features with the search page features. During this operation, the feature map of the considered indexed page is used as the basis and the query feature map as the filter of this cross-correlation operation. This operation is equivalent to calculating the dense inner product between both feature maps. However, it can take advantage of GPU optimized implementations in deep learning frameworks. Fig. 23 presents an illustration of the cross-correlation operation for one position in the image. This operation consists in comparing all local features of the image query (b) with all local features of the search image (a) in a dense way, which generates a similarity map (c), also called heatmap. A padding is added to the search image to preserve the spatial structure of the input image. To create the heatmap, the sum of the product between the matrices is calculated for each possible position (i,j). Given the feature map of the search image  $S$  and the feature map of the query image  $q$ , we aim to calculate  $G = q \otimes S$ , where  $G$  is calculated by:

$$G[i, j] = \sum_{u=-wq/2}^{wq/2} \sum_{v=-hq/2}^{hq/2} q[u, v]S[i + u, j + v] \quad (3.1)$$

In this formula  $wq$  and  $hq$  represent the width and height of the query feature map.

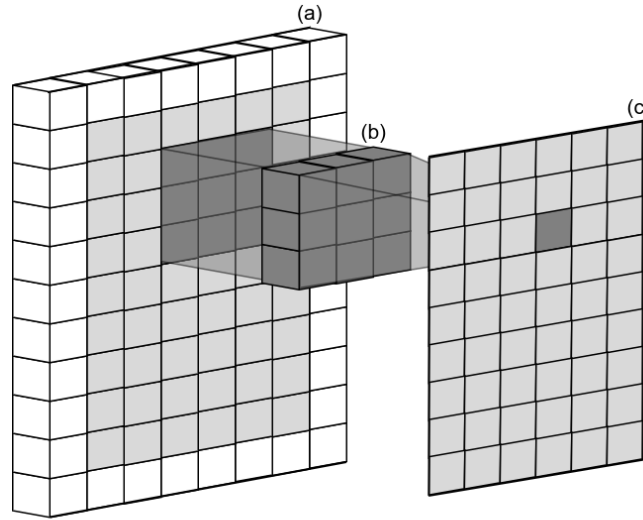


Figure 23 – Cross-correlation function. (a) feature map of search image, (b) feature map of query image, (c) generated heatmap. The white part of (a) indicates the margin.

Each position of the heatmap generated by this cross-correlation operation refers to a comparison between the query features and the features of a part, with the same size as the query, of the indexed page image. This heatmap has similarity values that vary according to the values of the page and query feature vectors. The higher values represent the positions with the most significant similarity and the lower values the positions with the lowest similarity. As all features were normalized before the cross-correlation operation, it is possible to compare the results obtained by different pages directly.

### 3.1.2.3 Bicubic Interpolation

With heatmaps in hand, we need to extract the coordinates of the best values from the map. These values represent the centers of the regions with the strongest similarity between the query and the search image. Since the heatmap created is smaller than the input image, it is necessary to use a step

to increase the heatmap size. Following the proposal by Bertinetto et al. (2016), we perform a bicubic interpolation (Algorithm 1). This interpolation is used to resize the heatmaps to the original input size. With this, it is possible to search directly for the similarity values, without the necessity of extra calculations for coordinates correction.

---

**Algorithm 1** Bicubic Interpolation
 

---

```

1: function BICUBIC_INTERPOLATION(matrix,  $x, y$ )
2:    $x_{\text{int}} \leftarrow \lfloor x \rfloor$ 
3:    $y_{\text{int}} \leftarrow \lfloor y \rfloor$ 
4:    $dx \leftarrow x - x_{\text{int}}$ 
5:    $dy \leftarrow y - y_{\text{int}}$ 
6:    $P \leftarrow \text{matrix}[x_{\text{int}} - 1 : x_{\text{int}} + 2][y_{\text{int}} - 1 : y_{\text{int}} + 2]$ 
7:    $result \leftarrow \text{bicubic\_kernel}(dx) \times \text{bicubic\_kernel}(dy) \times P$ 
8:   return  $result$ 
9: end function
10:
11: function BICUBIC_KERNEL( $t$ )
12:    $a \leftarrow -0.5$ 
13:    $at \leftarrow |t|$ 
14:   if  $at \leq 1$  then
15:     return  $(a + 2) \cdot at^3 - (a + 3) \cdot at^2 + 1$ 
16:   else if  $1 < at \leq 2$  then
17:     return  $a \cdot at^3 - 5a \cdot at^2 + 8a \cdot at - 4a$ 
18:   else
19:     return 0
20:   end if
21: end function

```

---

### 3.1.2.4 Region Selection

The interpolation step ensures that the heatmap has the same size as the input page. Thus, we can directly access the positions with the highest correlation values in this heatmap, to create a ranking. In Algorithm 2, a representation of the method for selecting the best values is provided, with an illustration of this operation shown in Fig. 24. The process is performed from heatmap (c), created using the query (a) and the search image (b). The first step consists in changing all the values located at the borders of the heatmap to zero (d). This change is performed using half the width of the query width for the left and right sides and half the height of the query for the top and bottom sides. This operation is performed for two reasons. First, if the center



were located in any of the positions defined as zero, part of the object would be located outside the image. The second reason is the influence of the margin on the cross-correlation operation.

After setting the border values to zero, it is necessary to collect the  $p$  positions from the heatmap peaks to calculate the objects' location in the search image. This operation is presented in Fig. 24 (e). However, it is not possible to collect these values directly. This occurs because an interpolation operation was performed to expand the heatmap size, which makes the neighboring values of the peak very close to its value. The solution is to replace the neighboring values with zero at each point collected. The amount of changed positions is based on the size of the query. Changing the values of an area of the same size as the query also avoids a large overlap between the returned positions. This operation is performed  $p$  times and generates the same effect as the application of non-maximum suppression of 0.33, which avoids extra calculation for removing overlaps in the following steps.

---

**Algorithm 2** Best Values Selection
 

---

```

1: function SELECT_BEST_VALUES(heatmap, p, query_w, query_h)
2:   border_width  $\leftarrow$  query_w/2
3:   border_height  $\leftarrow$  query_h/2
4:   for i  $\leftarrow$  1 to border_width do
5:     heatmap[i, :]  $\leftarrow$  0
6:     heatmap[i, -i]  $\leftarrow$  0
7:   end for
8:   for j  $\leftarrow$  1 to border_height do
9:     heatmap[j, :]  $\leftarrow$  0
10:    heatmap[-j, :]  $\leftarrow$  0
11:  end for
12:  best_positions  $\leftarrow$  empty list
13:  for k  $\leftarrow$  1 to p do
14:    (x, y)  $\leftarrow$  find_position_with_max_value(heatmap)
15:    best_positions[k]  $\leftarrow$  (x, y)
16:    for  $\Delta x \leftarrow -\text{border\_width}$  to border_width do
17:      for  $\Delta y \leftarrow -\text{border\_height}$  to border_height do
18:        heatmap[x +  $\Delta x$ , y +  $\Delta y$ ]  $\leftarrow$  0
19:      end for
20:    end for
21:  end for
22:  return best_positions
23: end function

```

---

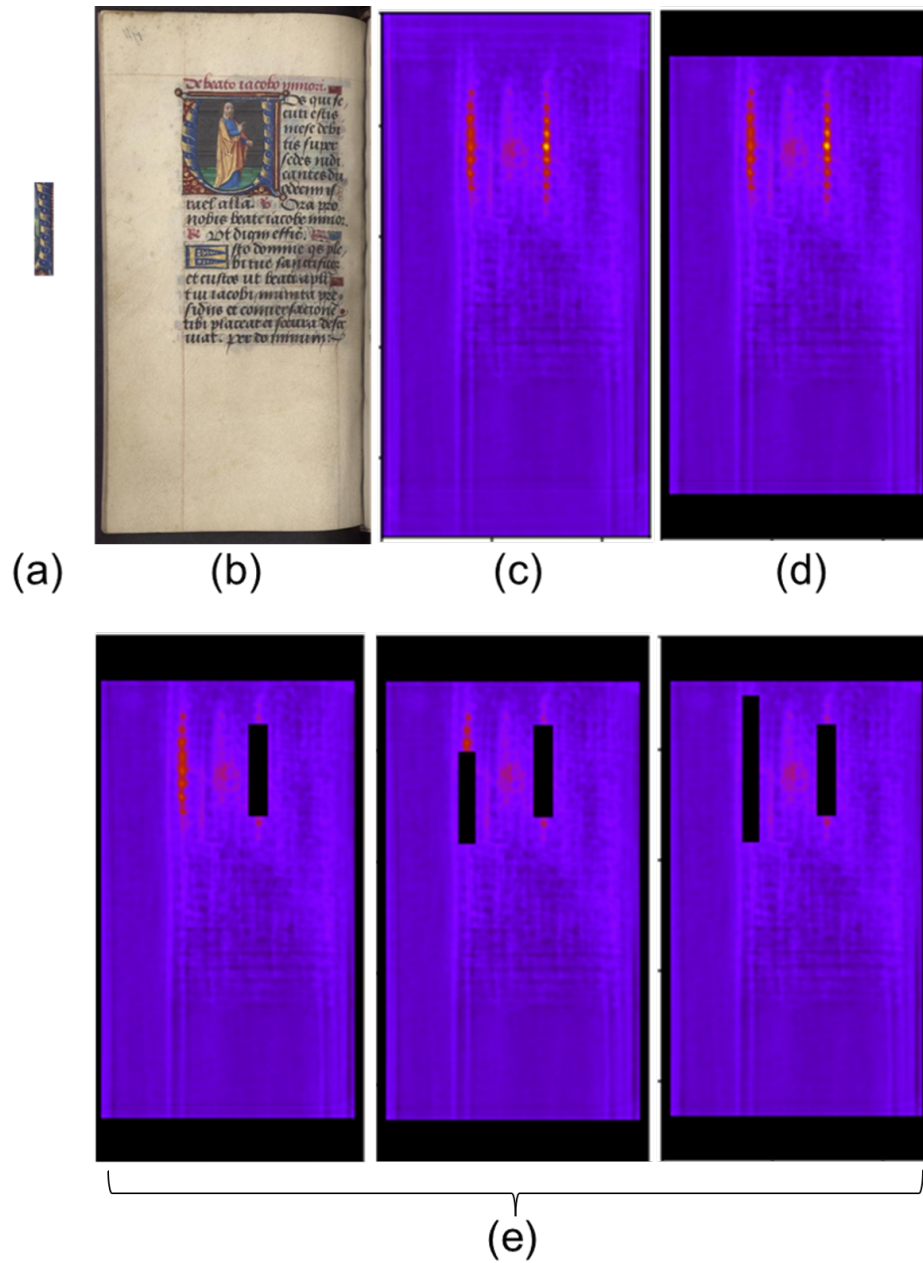


Figure 24 – Process for selecting the best values from the heatmap using  $p = 3$ . (a) input query, (b) input search image, (c) original heatmap, (d) heatmap after removal of borders, (e) selection of best values and removal of neighbors.

### 3.1.2.5 Ranking Creation

With the collection of these points for all the heatmaps, we can create a ranking with the best values returned by the system and its coordinates. This information is used to create the final system return. For the CBIR task, a ranking with the pages with the highest values is created directly. For the PS task, it is necessary to define the coordinates of the positions found. However, no regression operation is performed to find the bounding box size of each object. The original size of the query is used to calculate the new regions to suppress the lack of these values.

### 3.1.2.6 Multi-Scale Input

As an intermediate layer of the FCN is used, it is not expected that the method identifies objects with large size differences. This is a problem when the dataset used is composed of documents captured at different resolutions, generating images of different sizes. To solve this problem, we propose the use of multiple resized queries at the input of the system. This solution allows objects smaller or larger than the query to be identified.

Employing resized queries provides an effective approach to handle object detection at different scales within the images. By using queries at various scales as input, the system becomes capable of detecting objects regardless of their relative size. However, it is important to emphasize that this approach is not without challenges. The necessity of processing multiple input queries increases computational load, rendering the process more resource intensive.

## 3.2 Proposed Method with Binary Features

The main limitations of the proposed method are the memory space used to store the features generated in the offline phase and the computational cost for the cross-correlation in the online phase. Aiming to reduce these limitations, a feature binarization process is proposed. The use of binary features allows for a smaller storage space in the offline phase and the replacement of the traditional cross-correlation operation by the XOR cross-correlation, reducing computational costs. Figure 25 presents an overview of the proposed method with the use of binary features, the main differences between the original method and the binary version are represented in gray.

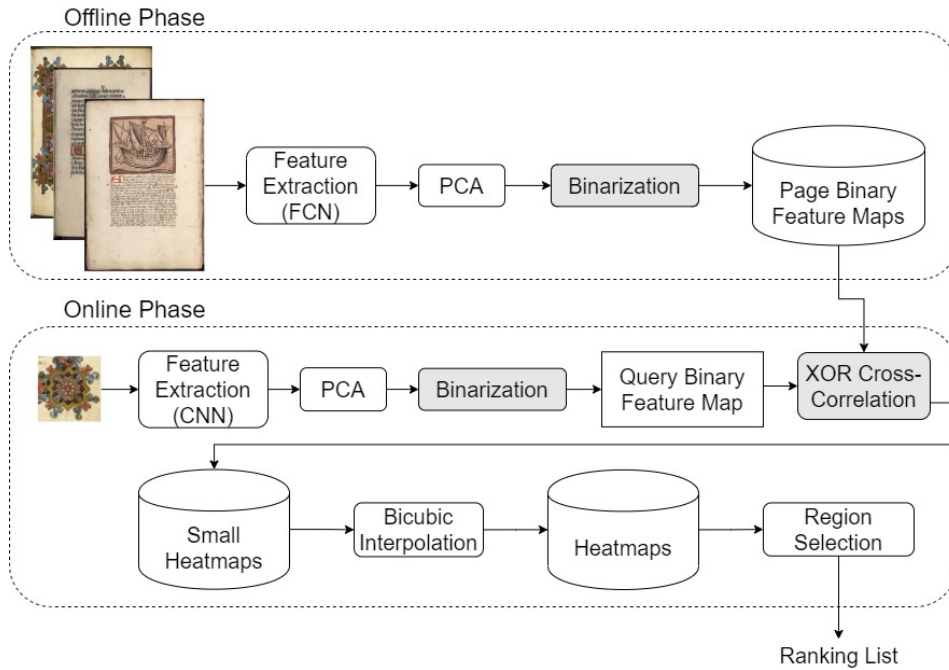


Figure 25 – Overview of the proposed framework for image retrieval and pattern spotting with binary features

### 3.2.1 Binarization Strategy

The proposed binarization strategy thresholds the feature map values based on a global squeeze vector. This vector is obtained by averaging each channel of a given set of images ( $T$ ) randomly selected from the document collection. Figure 26 gives us a general overview of the global squeeze vector creation, which is then used as a threshold vector during the binarization process.

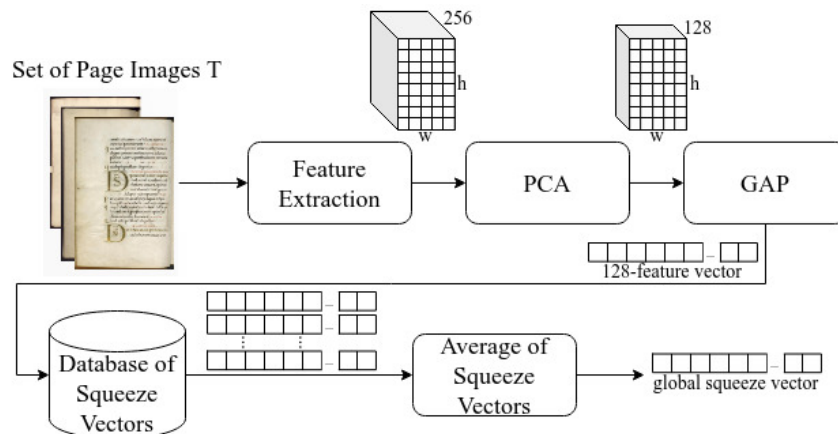


Figure 26 – Proposed method for creating the global squeeze vector

Algorithm 3 describes each step of the function used to compute the global squeeze vector. As we can see, the input is a set  $T$  of page images randomly selected from the document collection. The size of  $T$  is experimentally defined (50 images have shown to be enough).

---

**Algorithm 3** Global Squeeze Vector
 

---

```

1: function CREATE_GSV( $T$ )
2:    $squeeze\_vectors \leftarrow []$ 
3:   for  $image$  in  $T$  do
4:      $feat\_page \leftarrow Feature\_Extraction(image)$ ;
5:      $feat\_pca \leftarrow PCA(feat\_page)$ ;
6:      $feat\_gap \leftarrow GAP(feat\_pca)$ ;
7:      $squeeze\_vectors.append(feat\_gap)$ ;
8:   end for
9:    $GSV \leftarrow Average(squeeze\_vectors)$ ;
10:  return  $GSV$ 
11: end function

```

---

First (line 4 of the algorithm) for each image in  $T$  a feature map ( $feat\_page$ ) is computed using the feature extraction procedure based on an intermediate layer (block 3) of the VGG16, as described in subsection A. Then (line 5) the PCA is used for dimensionality reduction. The number of PCA components, experimentally defined, is the same for the query and page representations. After that (line 6), a Global Average Pooling (GAP) is applied on the feature map (named  $feat\_pca$  in the algorithm) previously reduced by PCA. GAP provides a squeeze vector ( $feat\_gap$ ), where each element is the global average representing a channel of the feature map. This operation allows the creation of a general descriptor for each channel map losing all spatial information.

Finally (line 7), we compute a global squeeze vector (GSV) represented by the average of the squeeze vectors computed on all the images of the set  $T$ . This global squeeze vector will be used to binarize the feature maps extracted from the images to be indexed during the off-line indexing phase and the feature map extracted from the query image during the on-line search phase. To this end, each feature vector is compared with the GSV at each position of the feature map. If the feature vector value is smaller than that in the GSV, it is changed to zero, otherwise to one.

### 3.2.2 XOR Cross-Correlation

With the extraction, processing, and storing of the features of all pages in the offline phase, it is now possible to query the digital collection (online phase). The same extraction and processing steps are performed for the query image in this phase. After this extraction, it is possible to compare the query and all regions of each feature map obtained in the offline step. For the comparison, we use a binary adaptation of the cross-correlation operation. In this adaptation, we replace the multiplication performed between the elements of each matrix with an XOR comparison (Fig 27). Given the feature map of the search image  $S$  and the feature map of the query image  $Q$ , we aim to calculate  $G$  for each position  $(i, j)$  of the search image  $S$ , as denoted in Eq. 3.2.

$$G[i, j] = \sum_{u=-Qw/2}^{Qw/2} \sum_{v=-Qh/2}^{Qh/2} \sum_{x=0}^{Qc} Q[u, v, x] \oplus S[i + u, j + v, x] \quad (3.2)$$

The XOR cross-correlation between the feature maps of a search image and a query returns a heatmap, where lower values represent regions with higher similarity and higher values represent regions with lower similarity. Since the features of all the search images are obtained with the same process, it is possible to directly compare the values of the heatmaps generated with the same query.

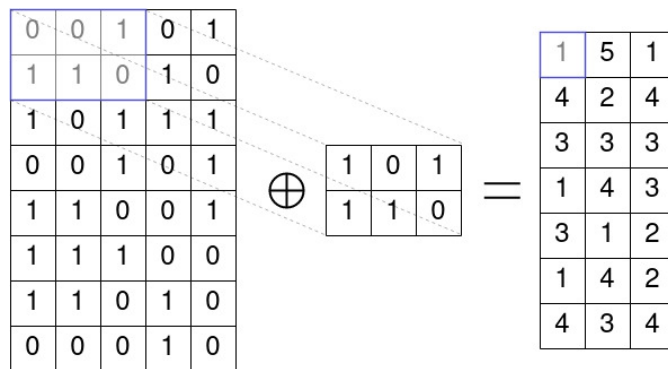


Figure 27 – XOR Cross-Correlation used to compute the heatmap between the query and each document page image.

### 3.3 Evaluation Protocol

The evaluation protocol developed for this work aims to answer the hypotheses presented in Section 1.2. For the implementation of the proposed models, the programming language Python with the TensorFlow 2 library was used. The computational resources used are a virtual machine with an Intel Xeon E5-2660 v4 CPU with six cores, 24GB of RAM, and an Nvidia K80 GPU, as well as a server with an Intel Xeon E5-2630 v4 CPU and 24GB of RAM. Some tests are also performed on the Google Collaborative Tool<sup>1</sup>.

The method of IR and PS will be evaluated with the metric mean average precision (mAP). This metric, presented in equation 3.3, consists of the mean over queries, of the average precision (AP) for each query  $q$  (BEITZEL; JENSEN; FRIEDER, 2009). Where the AP for each query is the area under the precision/recall curve. The AP is presented in equation 3.4, where  $R$  is the number of relevant documents,  $r$  is the rank of each relevant document, and  $P(r)$  is the precision of the top- $r$  retrieved documents (ZHANG; ZHANG, 2009).

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (3.3)$$

$$AP = \frac{\sum_r P(r)}{R} \quad (3.4)$$

Each query was compared once with all the dataset images. For the IR task, the AP was computed directly. For the PS task, the Intersection over Union (IoU) between the occurrences of the query in the ground-truth and in the system return was initially computed. The IoU aims to compute the area of overlap between the position of the ground-truth and the position of the system return, indicating the ratio between the intersection area and the union area between the two. The equation 3.5 presents this equation. All cases where  $\text{IoU} > 0.5$  were considered as correct, and after the IoU calculation, the AP was calculated. Finally, the mAP for all results was calculated. These steps follow the evaluation protocol presented by En et al. (2016a) and Wiggers et al. (2019b).

<sup>1</sup> <https://colab.research.google.com/>

$$IoU = \frac{B_{gt} \cap B_{sys}}{B_{gt} \cup B_{sys}} \quad (3.5)$$

Several works in the literature use datasets composed of historical documents for the Word Spotting task (AHMED; AL-KHATIB; MAHMOUD, 2017). In this work, the focus is on Pattern Spotting, so the datasets for Word Spotting cannot be used without new labeling. One area that can use techniques similar to Pattern Spotting in historical documents is the identification of similar patterns in paintings, in the area known as Visual Link Retrieval (SEGUIN et al., 2016), (CASTELLANO; VESSIO, 2020). Although similar techniques can be used, the paintings present several differences from the historical documents, so they will not be used. Another task performed with paintings is Object Detection, as in Gonthier et al. (2018). In addition to the differences between paintings and historical documents, the datasets used for Object Detection have pre-defined classes, with a strong intra-class variability. These differences hinder the use of Pattern Spotting methods. For this reason, only two datasets found in the literature were suitable for evaluating the proposed method in a quantitative form: DocExplore and Tobacco800.

In addition to the evaluation of the proposed method, the DocExplore and Tobacco800 datasets allow for a comparison with different approaches in the literature. These datasets have been used in other works for IR and PS tasks, thus providing a baseline for the analysis of the results obtained. This also makes these datasets suitable for addressing the research hypothesis of this work. In addition to DocExplore and Tobacco800, the Horae dataset will be used for a qualitative evaluation of the proposed method. A comprehensive presentation of these datasets can be found in Section 2.2.5.

### 3.4 Final Considerations

This chapter has presented novel methods for IR and PS in ancient documents. These methods utilize a combination of FCNs, transfer learning, cross-correlation, and heatmaps to perform IR and PS tasks without requiring training in the target domain. Additionally, a binary variation of the method was introduced, incorporating a feature conversion and XOR cross-correlation approach.



In this chapter, we have provided a detailed description of these methods, including feature extraction and processing, and the process for performing IR and PS tasks. Moreover, the chapter has presented the evaluation protocol employed during the execution of the experiments, ensuring transparency and reproducibility for future research in this area.

The following chapter will present the results and the corresponding discussions on their effectiveness. The experiments aim to assess the proposed methods' performance in IR and PS tasks in ancient documents using different datasets and scenarios.

## 4 Experimental Results

This Chapter presents the results obtained from the experiments, as well as discussions of the results and comparisons with existing work in the literature. Section 4.1 presents the results of the method utilizing float features, while Section 4.2 presents the results of the method with the binarization process. Section 4.3 presents the final considerations, discussing the results and limitations of the methods presented.

### 4.1 Experiments with the Method Using Float Features

The proposed method was evaluated using three distinct datasets: DocExplore, Tobacco800, and Horae. The selection of these datasets was based on their diverse origins, varied capture equipment, and distinct configurations. A description of these datasets is presented in Section 2.2.5.

For the system evaluation in the DocExplore and Tobacco800 datasets, each query was compared once with all the dataset images. For the IR task, the Average Precision (AP) was computed directly. For the PS task, the Intersection over Union (IoU) between the occurrences of the query in the ground truth and in the system return was initially computed. All cases where  $\text{IoU} > 0.5$  were considered as correct, after the IoU calculation the AP was calculated. Finally, the mean Average Precision (mAP) for all results was calculated, providing a more comprehensive evaluation of the proposed method’s performance on these datasets. This evaluation methodology allows for a thorough assessment of the system’s effectiveness in both tasks and enables a direct comparison with existing work in the literature.

#### 4.1.1 Results on DocExplore Dataset

As explained in Section 3.1.1, when training is not conducted with images from the target domain, it is expected that the feature maps provided by the shallow and intermediate layers of the FCN used may better represent document images than those produced by the deep layers. To evaluate the differences between layers of different depths of the network, a comparison was

performed. Table 2 presents the results observed for the DocExplore dataset using different layers of the VGG16 architecture showed in Fig. 22. The VGG16 architecture is a deep model comprising five blocks containing two or three convolutional layers followed by a pooling layer. We have considered the output of the last convolutional layer of blocks 3, 4, and 5. In all cases, we applied a PCA transform reducing the feature vectors to 64 components. The PCA was trained on 100 pages of the dataset. Small queries showing width or height with less than 15 pixels are resized. After feature extraction and creation of the heatmaps, each heatmap’s 15 most significant values greater than 0.1 were collected ( $p = 15$ ). The selection of values higher than 0.1 was performed to avoid selecting values that do not represent a significant similarity. The use of the 15 most significant values from the heatmap was performed to optimize processing and avoid creating a large result vector. During the experiments, no changes were shown in the results generated when using more than 15 results.

| Method | mAP IR        | mAP PS        | Memory (GB) | Avg. Time (s) |
|--------|---------------|---------------|-------------|---------------|
| block3 | <b>0.8075</b> | <b>0.6392</b> | 7.43        | 135.15        |
| block4 | 0.6256        | 0.3774        | 1.89        | 28.67         |
| block5 | 0.2495        | 0.1453        | 0.51        | 21.34         |

Table 2 – Results with different layers of VGG16 architecture for the DocExplore dataset using PCA with 64 components. The time represents the average time used for all stages of the online phase for one query. The memory represents the space used to store features for all images in the dataset.

As shown in Table 2, block3 provided the best representation of the dataset images when compared to block4 and block5. The main reason for the best result obtained from a shallower layer is using a pre-trained model. The network trained in another domain has the deeper layers representing semantic information with no importance to the target problem. In contrast, shallower layers represent more general information related to image structure. Although such a strategy brings better results, it is more expensive in memory and time-consuming, since shallower layers generates large feature maps.

An additional advantage of shallower layers is their ability to represent smaller objects (with less than 2000 pixels), whereas deeper layers tend to ignore them. This aspect can be observed for the PS results for each category in DocExplore, as seen in Table 3. The use of the last layer of Block3 produced the best results in 14 out of the 35 categories of the dataset. It is interesting to note that the 10 categories with the smallest objects showed the best results

with this block. The use of Block4 produced the best result for 10 categories and was tied with Block3 in 11 categories. On the other hand, the use of Block5 was worse in all cases where there was no tie.

Although Block4 has shown superior results to Block3 in many categories, there is a significant difference in the overall mAP observed in Table 2. This is because these categories have minor impact on the result when using the evaluation proposed by (EN et al., 2016a). The top 10 categories with the best results using Block4 have only 98 out of 1446 queries in DocExplore. Additionally, there is a small difference between the mAP of the top-performing categories in Block4 compared to Block3. The categories with the highest results in Block4 have a mean difference of only 0.055 percentage points compared to the same categories in Block3. On the other hand, the categories with superior results in Block3 have a mean difference of 0.240 percentage points when compared to the same categories in Block4.

A key factor to consider is the relevance of small patterns for IR and PS tasks in historical documents. A historian may be interested in details of an image or small markings. Therefore, an efficient IR and PS system should be able to find objects regardless of their size. In light of this, and considering the overall best results, we opted to use Block3 of the VGG16 architecture.

Although the use of shallow layers proved to be more suitable for the addressed problem, their performance in terms of memory and time-consuming is inferior to deeper layers. One alternative to deal with large feature maps is to apply a technique for dimensionality reduction. Table 4 presents the results generated by considering a different number of components in the proposed PCA. As one can see, the best results for both tasks, IR and PS, were observed using 128 components. Considering the PS task and a significance level of 0.05, Friedman’s test showed a significant difference between the PCA results. We can also see it in the Nemenyi post-hoc test in Fig. 28. Although there is a statistical difference between the results that justify using the PCA setting with 128 components, it is important to consider the impact of this parameter on the execution time. If the objective is to perform the search quickly, it is still possible to reduce the number of PCA components keeping a competitive result as also observed in Table 4.

Table 5 compares our best results with related works presented in the

| Category        | # Samples | # AVG<br>Pixels | mAP PS<br>Block3 | mAP PS<br>Block4 | mAP PS<br>Block5 |
|-----------------|-----------|-----------------|------------------|------------------|------------------|
| bateau          | 13        | 31196           | 0,4578           | 0,4395           | 0,3731           |
| bateau_d        | 6         | 21065           | 0,3064           | 0,3172           | 0,2414           |
| bateau_g        | 12        | 18885           | 0,4335           | 0,3935           | 0,3542           |
| BP              | 12        | 24914           | 1,0000           | 1,0000           | 1,0000           |
| croix           | 92        | 1271            | 0,7629           | 0,2855           | 0,0000           |
| D               | 35        | 7202            | 0,8387           | 0,8590           | 0,5843           |
| double_sep      | 87        | 1395            | 0,2561           | 0,0320           | 0,0006           |
| encadrement     | 59        | 8403            | 0,8448           | 0,7857           | 0,4104           |
| grand_A         | 15        | 4847            | 0,7265           | 0,7341           | 0,0346           |
| henri_d         | 5         | 12634           | 0,4331           | 0,4490           | 0,2412           |
| henri_g         | 3         | 12914           | 0,8565           | 0,9028           | 0,7865           |
| losange         | 117       | 4104            | 0,9987           | 0,9973           | 0,6863           |
| marqueur        | 409       | 861             | 0,6566           | 0,2774           | 0,0000           |
| obj_1           | 4         | 23784           | 0,4013           | 0,4498           | 0,2525           |
| obj_2           | 8         | 417541          | 1,0000           | 1,0000           | 1,0000           |
| obj_3           | 2         | 41503           | 1,0000           | 1,0000           | 1,0000           |
| obj_31          | 4         | 30193           | 1,0000           | 1,0000           | 1,0000           |
| obj_34          | 8         | 33014           | 0,8327           | 0,9780           | 0,8827           |
| obj_35          | 4         | 38835           | 1,0000           | 1,0000           | 0,8544           |
| obj_36          | 2         | 14544           | 1,0000           | 1,0000           | 0,8542           |
| obj_37          | 2         | 9950            | 1,0000           | 1,0000           | 1,0000           |
| obj_38          | 4         | 15657           | 1,0000           | 1,0000           | 0,6833           |
| obj_39          | 3         | 10527           | 0,7500           | 0,7500           | 0,3927           |
| obj_40          | 7         | 14273           | 0,6695           | 0,6871           | 0,3783           |
| obj_42          | 2         | 73040           | 1,0000           | 1,0000           | 1,0000           |
| obj_61          | 8         | 9738            | 1,0000           | 1,0000           | 0,6118           |
| pdp             | 29        | 1763            | 0,9935           | 0,8424           | 0,0266           |
| petit_A         | 35        | 1520            | 0,7633           | 0,2711           | 0,0002           |
| rubanlettrine   | 68        | 3495            | 0,5039           | 0,3845           | 0,0599           |
| rubanlettrine_b | 3         | 6381            | 0,6718           | 0,8666           | 0,1149           |
| S               | 147       | 1459            | 0,7928           | 0,2385           | 0,0006           |
| simple_sep      | 160       | 667             | 0,1438           | 0,0027           | 0,0000           |
| status          | 12        | 18489           | 0,9436           | 0,9827           | 0,7625           |
| T               | 39        | 1612            | 0,7081           | 0,1210           | 0,0001           |
| triple_sep      | 30        | 2319            | 0,1543           | 0,0452           | 0,0014           |

Table 3 – Results of PS with different layers of the VGG16 architecture by category for the DocExplore dataset.

| PCA Comp. | mAP IR        | mAP PS        | Memory (GB) | Avg. Time (s) |
|-----------|---------------|---------------|-------------|---------------|
| 8         | 0.6361        | 0.4743        | 0.96        | 22.34         |
| 16        | 0.7504        | 0.5741        | 1.89        | 56.39         |
| 32        | 0.7961        | 0.6259        | 3.73        | 83.01         |
| 64        | 0.8075        | 0.6392        | 7.43        | 135.15        |
| 128       | <b>0.8131</b> | <b>0.6442</b> | 14.82       | 254.38        |

Table 4 – Effect of applying different PCA components to the output of the block3. The time represents the average time used for all stages of the online phase for one query.

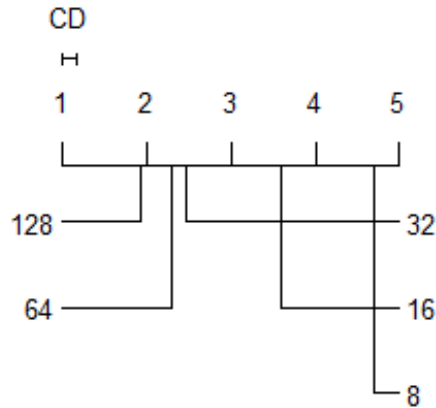


Figure 28 – Nemenyi test for different PCA configurations in the PS task.

literature. As can be observed, the method described in this work presents superior results in both tasks, IR and PS. For the IR task, the proposed method generated a result 40.2% higher than that in (EN et al., 2016a). For the PS task, the result was 136.8% higher than the best result presented in (ÚBEDA et al., 2020).

Although the method presented by Úbeda et al. (2020) uses feature maps with the exact spatial resolution as block3, we can observe a significant difference in the mAP for both tasks. A superior result of our method can be observed even when using a PCA with eight components. The reason for such a positive result is twofold. First, the use of an intermediate layer of the FCN architecture. Second, the query is used without any transformation. It is essential to notice that the query’s size is normalized just when we use block5. The feature maps provided by the block3 of the used FCN allow simultaneous comparison regardless of the image size and ratio. Differently, in (ÚBEDA et al., 2020), the query is always size normalized by adding it to a frame with a fixed size. As discussed previously, the insertion of an artificial frame or the

use of padding changes the features generated, limiting comparisons.

| Method                            | mAP IR        | mAP PS        |
|-----------------------------------|---------------|---------------|
| En et al. (2016a)                 | 0.5801        | 0.1569        |
| Wiggers et al. (2019b)            | 0.3860        | 0.1740        |
| Úbeda et al. (2020)               | 0.5770        | 0.2720        |
| Mohammed, Märgner & Ciotti (2021) | -             | 0.2510        |
| Dias et al. (2022)                | 0.5321        | 0.1996        |
| Proposed Method                   | <b>0.8131</b> | <b>0.6442</b> |

Table 5 – Results for DocExplore dataset (using the evaluation protocol presented in (EN et al., 2016a)).

Tables 6 and 7 presents a comparison between state-of-the-art methods considering different query sizes and aspect ratios. As in Úbeda et al. (2020), we consider as large all queries with  $\log(w*h) > 10$  and use the tolerance of 0.2 in the aspect ratio to define if the query is square or non-square. As in the methods used for comparison, the worst performance in our case occurred with the use of small and non-square queries, while the best performance occurred with big and square queries. In all cases, the proposed method produced a better result than the related works in the literature. Besides the best result, the values presented in Tables 6 and 7 allow us to observe that the proposed method presents a smaller difference in the mAP between big non-square queries and small non-square queries. For the PS task, the methods described by En et al. (2016a) and Úbeda et al. (2020) present a difference between these categories of 171.8% and 137.9%, respectively. In the proposed method, this difference is only 24.7%. This experiment shows that our method is more robust to shape variations in small queries when compared to other works available in the literature.

|       |              | Image Retrieval (mAP) |                     |                 |
|-------|--------------|-----------------------|---------------------|-----------------|
| Size  | Aspect Ratio | En et al. (2016a)     | Úbeda et al. (2020) | Proposed Method |
| big   | square       | 0.881                 | 0.749               | <b>0.937</b>    |
| small | square       | 0.801                 | 0.742               | <b>0.927</b>    |
| big   | non-square   | 0.701                 | 0.660               | <b>0.826</b>    |
| small | non-square   | 0.535                 | 0.459               | <b>0.792</b>    |

Table 6 – DocExplore results for IR considering query sizes and the aspect ratio (using the evaluation protocol presented in (ÚBEDA et al., 2020)).

The boxplot presented in Fig. 29 shows the values obtained for the mAP in the IR task for each category of Docexplore. As can be seen, the

| Size  | Aspect Ratio | Pattern Spotting (mAP) |                     |                 |
|-------|--------------|------------------------|---------------------|-----------------|
|       |              | En et al. (2016a)      | Úbeda et al. (2020) | Proposed Method |
| big   | square       | 0.546                  | 0.681               | <b>0.880</b>    |
| small | square       | 0.102                  | 0.546               | <b>0.858</b>    |
| big   | non-square   | 0.405                  | 0.509               | <b>0.752</b>    |
| small | non-square   | 0.149                  | 0.214               | <b>0.603</b>    |

Table 7 – DocExplore results for PS considering query sizes and the aspect ratio (using the evaluation protocol presented in (ÚBEDA et al., 2020)).

method achieved 100% mAP in all queries of 13 out of 35 categories. In only one case, the mAP was lower than 10%, this happened in the 'rubanletrine' category, with the lowest mAP of 0.34%. This category also presented the greatest variation between the best and worst results, with 75% of the queries having results higher than 87.4%. The category with the lowest average mAP was 'croix', while the category with the lowest median mAP was 'bateau\_d'.

The low average mAP obtained for the 'croix' category can be attributed to the arrangement of objects across the pages. Despite the dataset containing 91 occurrences of this object, they are predominantly concentrated on a few pages, resulting in a scenario where a single page may contain more than 10 instances of the same object. Consequently, the penalty for incorrectly identifying a page is significantly higher. It is noteworthy that, despite achieving the lowest average for the IR task, the 'croix' category showcased satisfactory results in the PS task, as illustrated in the subsequent analysis. The superior performance in the PS task compared to the IR task can be attributed to the presence of multiple objects on the same page. This characteristic allows for a single correct identification in the IR task to correspond to multiple successful identifications in the PS task.



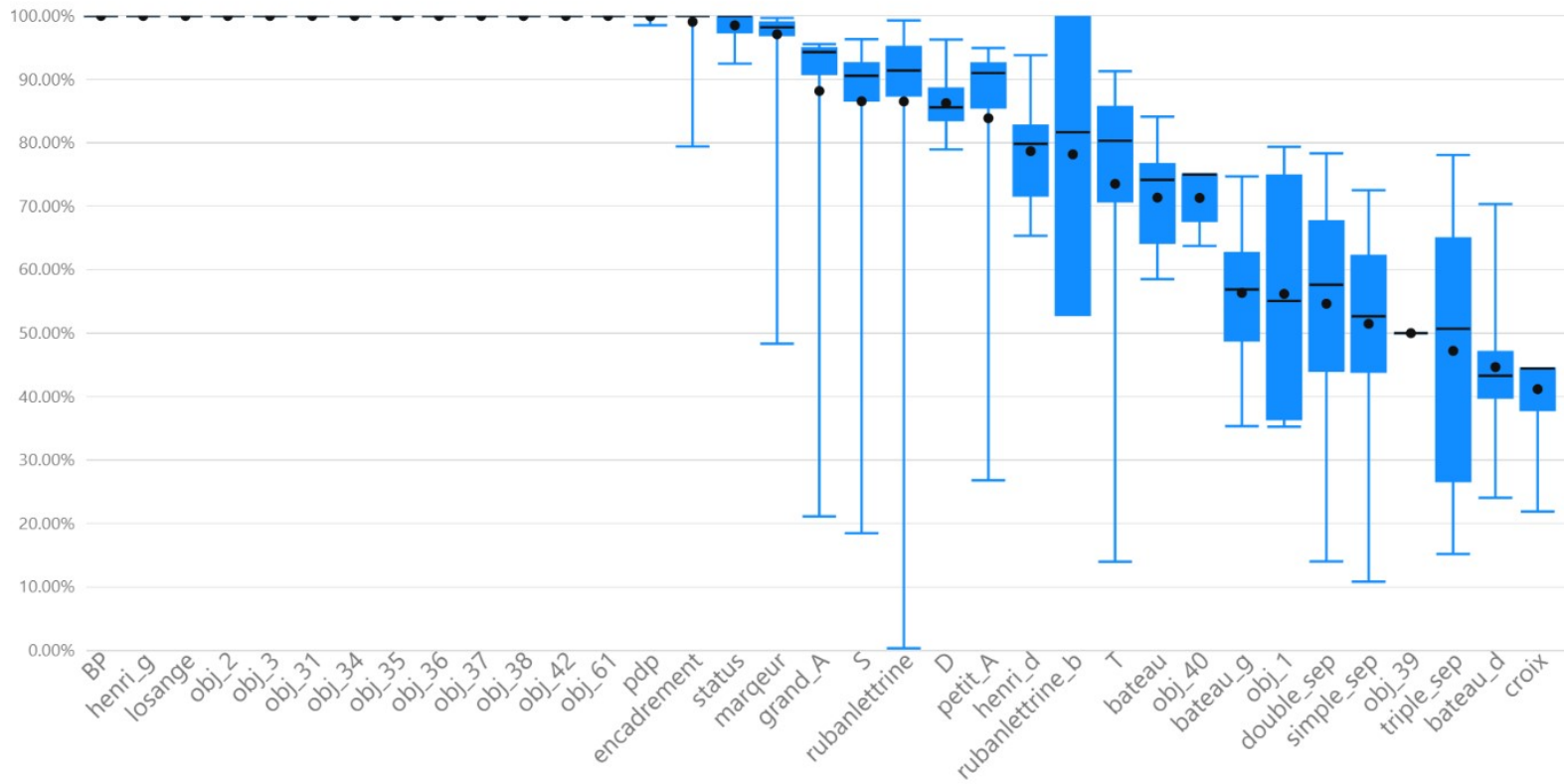


Figure 29 – Box plot with IR results for the DocExplore dataset

The boxplot in Fig. 30 shows the mAP values for the PS task. The method achieved 100% accuracy on all queries of 10 categories. The 'marqueur' category had the largest difference between the best and worst results, which can be attributed to its large number of queries (409) and small size. The categories with the worst means and medians were 'simple\_sep', 'double\_sep' and 'triple\_sep'. The difference between the worst results for the IR and PS tasks is due to the presence of multiple objects on the same page.

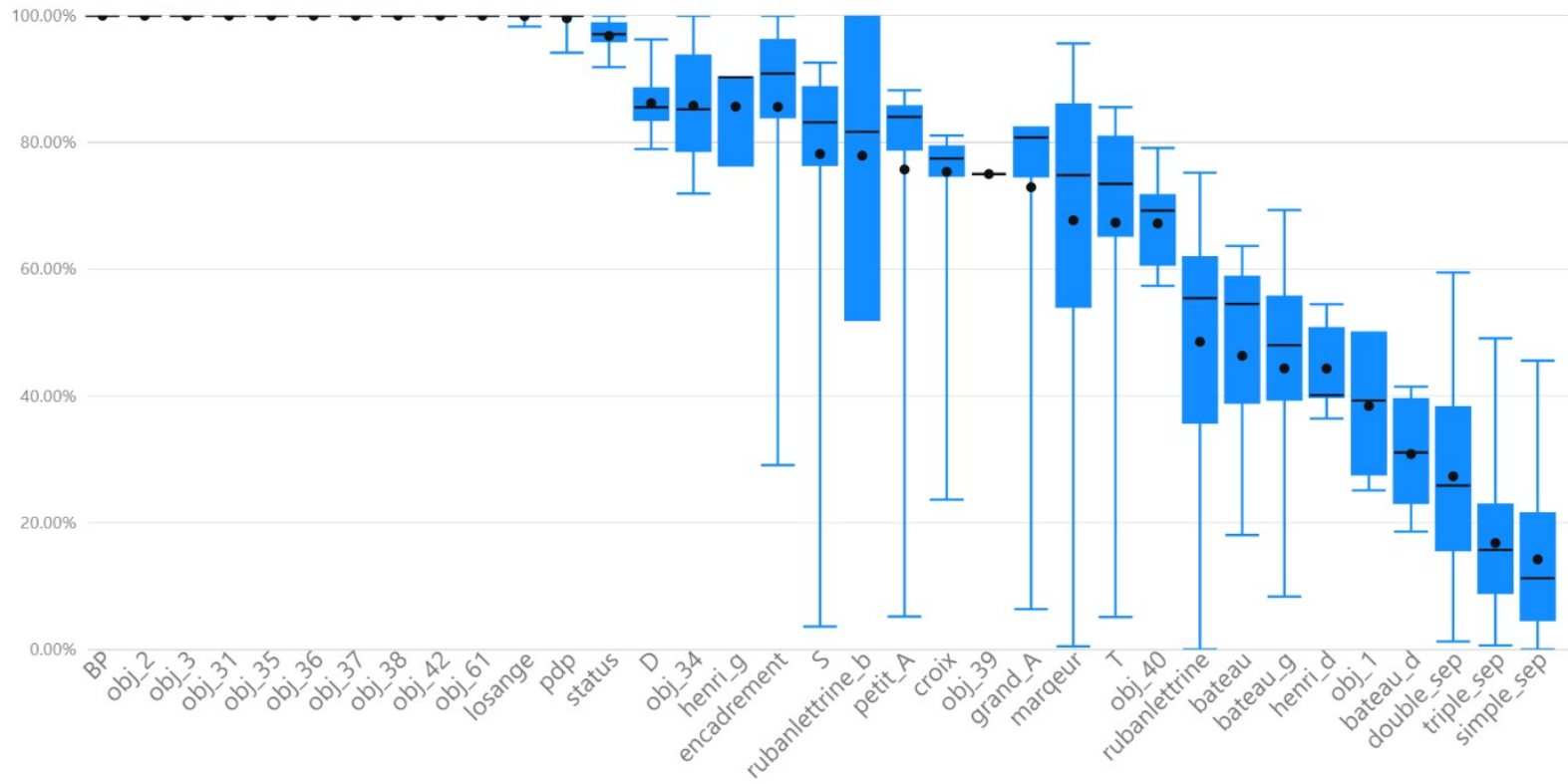


Figure 30 – Box plot with PS results for the DocExplore dataset

Fig. 31 presents some qualitative results related to the best settings of our method. In this figure, a list with the query and the top-5 results is presented. Above each result, we can see the page where it was found and below the normalized similarity. The value observed was divided by the value related to the perfect match, i.e., when the query is compared with itself without any difference. As one can see, the method is able to retrieve objects that are similar to different input queries. However, some of the presented results are computed as errors in the mAP calculation. This problem is observed in the last three categories presented in the image. When the face of King Henri II is used as an entry into the system, it is common for some other faces to be returned. This is caused by the high visual similarity between the drawings. The same problem occurs with the three types of separators and the three types of ships. One of the main reasons for the low performance of the system in these queries is the large visual similarity between different categories, as shown in Fig. 32. There are three separator categories in the dataset. In some cases, the system identifies the double category as two categories: the simple category and the triple category as three occurrences of the simple category or one of the double category. The same problem occurs with the three-ship categories in the dataset.

Although similar categories decrease the system's mAP, it is acceptable that objects visually similar to the search are returned. In a situation where a historian searches for a particular object, the person may have only one image to use as a query, requiring that objects with slight differences are also returned. Unlike the problem with similar images, the Bed Crown category can represent the most significant limitation of the system. This category has only four samples, presented at the top of the Fig. 33. This category can be divided into two size groups: small crowns and big crowns. When searching for a small crown, only the small crowns were returned, and when searching for a large crown, only the large ones were returned. This demonstrates that the method presented in this work is unable to find an object if it has a large size variation concerning the query. The bottom of Fig. 33 shows the heatmaps generated for the four queries in the bed crown category with a page containing an occurrence of that query with the large size. As we can see, the search for a small query in an image with a large query generates peaks in random places, while in images with similar sizes, the peak of the heatmap is located in the




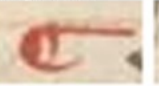




































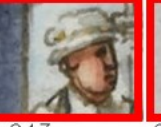












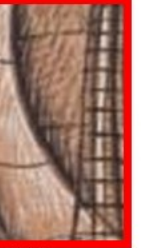
| Query   | 1st   | 2nd   | 3rd  | 4th   | 5th   |
|---|---|---|--|---|---|
|    | page944<br>    | page283<br>    | page869<br>    | page1564<br>   | page1414<br>   |
|   | 0.745   | 0.695   | 0.688  | 0.674   | 0.671   |
|    | page1206<br>   | page415<br>    | page415<br>    | page1206<br>   | page1206<br>   |
|   | 0.885   | 0.578   | 0.542  | 0.539   | 0.266   |
|    | page231<br>    | page715<br>    | page556<br>    | page556<br>    | page556<br>    |
|   | 0.812   | 0.776   | 0.761  | 0.752   | 0.738   |
|   | page1245<br>  | page1132<br>  | page465<br>   | page385<br>   | page796<br>   |
|   | 0.865   | 0.697   | 0.693  | 0.672   | 0.658   |
|  | page445<br>  | page628<br>  | page1278<br> | page1585<br> | page1530<br> |
|   | 0.922   | 0.463   | 0.429  | 0.419   | 0.393   |
|  | page1555<br> | page752<br>  | page66<br>   | page497<br>  | page74<br>   |
|   | 0.810   | 0.718   | 0.711  | 0.710   | 0.708   |
|  | page445<br>  | page1585<br> | page1278<br> | page445<br>  | page445<br>  |
|   | 0.829   | 0.319   | 0.282  | 0.243   | 0.242   |
|  | page497<br>  | page1519<br> | page1201<br> | page206<br>  | page497<br>  |
|   | 0.668   | 0.596   | 0.588  | 0.582   | 0.570   |
|  | page490<br>  | page406<br>  | page176<br>  | page104<br>  | page1586<br> |
|   | 0.829   | 0.505   | 0.465  | 0.418   | 0.386   |

Figure 31 – Retrieval results for DocExplore. The objects in red represent errors.

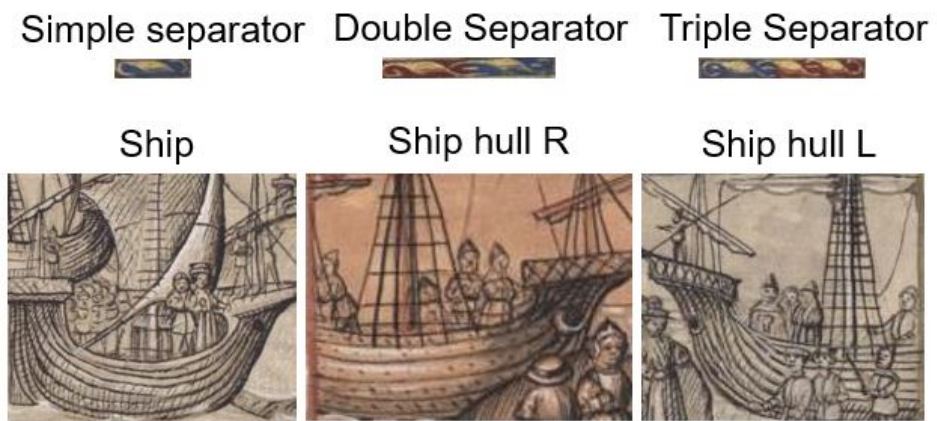


Figure 32 – Categories with the worst results.

center of the object.

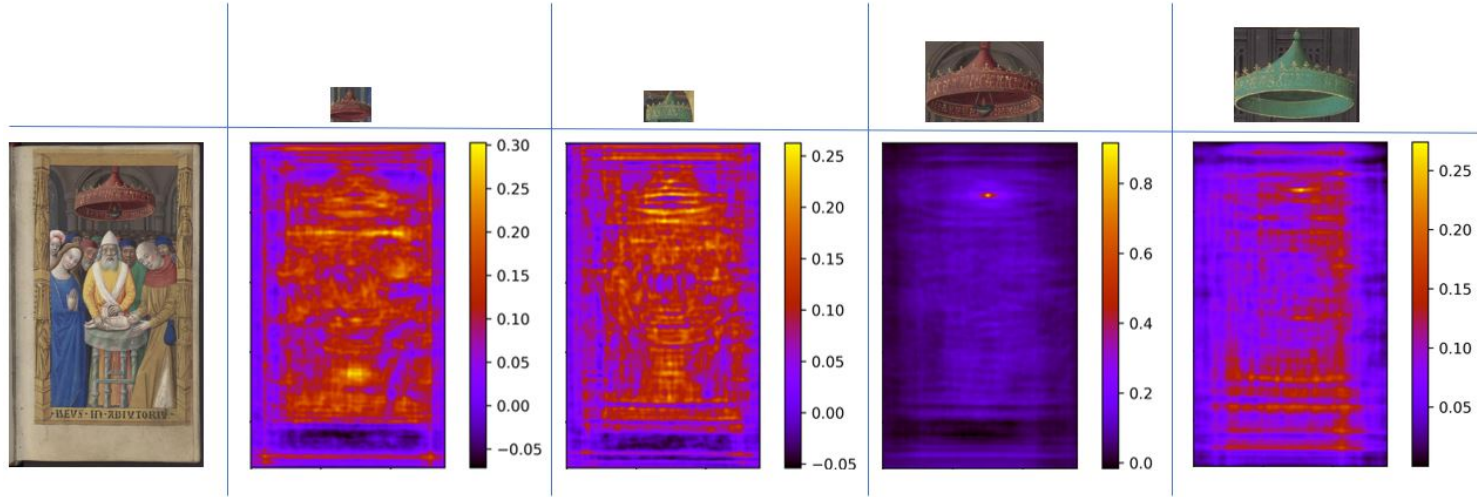


Figure 33 – Samples from bed crown category.

When the objective is to locate objects of different sizes, such as within the Bed Crown category, a viable approach involves using resized queries as inputs to the system. The underlying concept of this strategy involves conducting multiple searches using queries of different dimensions. By employing resized queries, the system can effectively adapt to the size variations present in the target objects. Each search iteration involves using a query that has been resized to a range of scales, thereby accommodating the potential range of sizes for the objects.

Figure 34 exemplifies the adaptation of an object belonging to the bed crown category for the use of multiple input sizes. In this example, scale factors ranging from 0.25 to 2 were considered, in increments of 0.25. It is possible to observe that all the images were normalized to a uniform size. This procedure was adopted with the purpose of enabling a direct comparison of the values resulting from the cross-correlation operation.

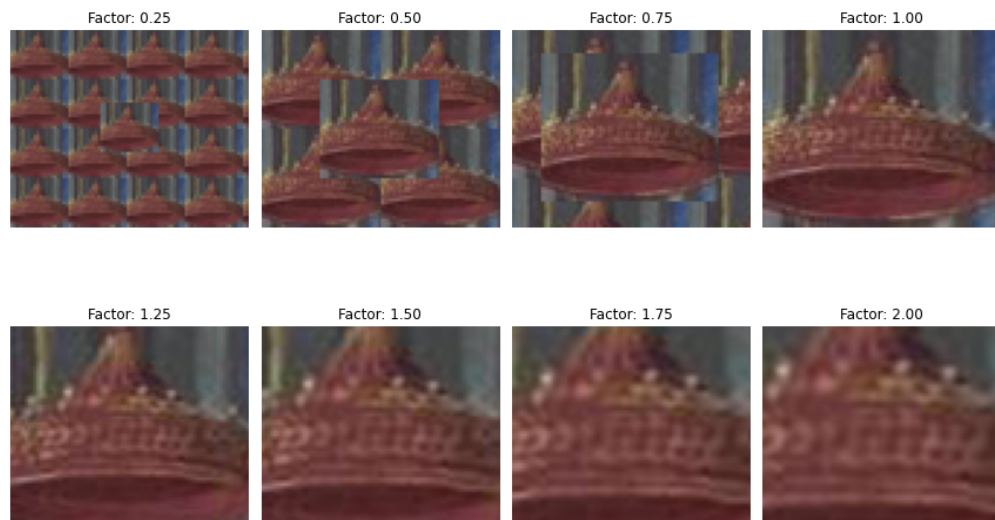


Figure 34 – Query resized for multiple input sizes of the system.

It is worth noting that standardizing the dimensions of the images not only facilitates the precise evaluation of cross-correlation values, but also allows for efficient exploration of parallelism techniques during the implementation of this process. Consequently, the adverse effects on computational performance associated with this processing strategy are mitigated. Through conducting a comprehensive series of experiments, we infer that using a mosaic as the background for resized images yields the most promising results. This approach proves to be superior, as the inclusion of a fixed background, such as the texture



of a blank page or a solid color, introduces additional noise that impairs the fidelity of the comparison.

While the resizing strategy proves effective for specific cases, such as the Bed Crown category, its application to all queries within DocExplore decreases the overall system performance. This occurs because the objects found on the dataset pages do not present significant size variation within categories. The utilization of multiple scales as inputs to the system negatively affects outcomes in such scenarios, leading to an increased incidence of misidentified objects.

### 4.1.2 Results on Horae Dataset

The Horae dataset is not annotated, but we have used it in a set of qualitative analyses. The motivation is to show the generalization power of the proposed method on a different set of pages and queries. We performed the tests with the configuration that generated the best result for the DocExplore dataset. Due to the large size of the Horae document images, we reduced their size to 25% of the original one.

The Horae dataset has some exciting aspects for this analysis. The pages that compose the dataset have been extracted from several books, showing more variability than that of DocExplore. We can observe visually similar objects drawn by different persons using different inks and types of papers. Besides, this dataset also has a greater variety in terms of the acquisition process, such as various image sizes and examples of grayscale and color images.

Fig. 35 presents the experimental results on the Horae dataset. The first object represents the query used as input, and the following the top-5 object results returned by the search. For a better analysis, we used different queries from those used in the DocExplore annotation. As we can observe, the results were satisfactory for all searched queries.

Some interesting results are noticeable in Fig. 35. For instance, the retrieval of colored objects when the input is a grayscale query. Such a capability is helpful in situations where there is only one image to be used as a query. The results returned by the strawberry query for a single page of the dataset are shown in Fig. 36. Even using a query extracted from another page, the proposed method can find four occurrences of the searched object (marked with a white bounding box on the image). However, the method was not able to find















































| Query   | 1st  | 2nd  | 3rd  | 4th  | 5th  |
|---|--|--|--|--|--|
|    | page24<br><br>0.780     | page3637<br><br>0.578   | page1152<br><br>0.578   | page394<br><br>0.571    | page3212<br><br>0.570   |
|    | page3083<br><br>0.780   | page1018<br><br>0.523   | page381<br><br>0.506    | page2919<br><br>0.498   | page1018<br><br>0.485   |
|    | page112<br><br>0.707    | page113<br><br>0.509    | page2954<br><br>0.501   | page3286<br><br>0.501   | page2262<br><br>0.481   |
|   | page112<br><br>0.856   | page386<br><br>0.419   | page523<br><br>0.409   | page3354<br><br>0.409  | page2937<br><br>0.401  |
|  | page279<br><br>0.804  | page253<br><br>0.564  | page217<br><br>0.530  | page1763<br><br>0.481 | page216<br><br>0.462  |
|  | page217<br><br>0.796  | page216<br><br>0.622  | page217<br><br>0.622  | page216<br><br>0.615  | page1902<br><br>0.599 |
|  | page3268<br><br>0.847 | page3268<br><br>0.527 | page3268<br><br>0.458 | page3268<br><br>0.452 | page1716<br><br>0.387 |
|  | page1025<br><br>0.818 | page3459<br><br>0.667 | page3457<br><br>0.591 | page661<br><br>0.565  | page3456<br><br>0.557 |

Figure 35 – Retrieval results for Horae dataset.

the object when an occlusion occurs (black bounding box on the image).



Figure 36 – Results for one page in the Horae dataset. The white bound boxes represent the results found. The black bound box represents the object that was not detected.

### 4.1.3 Results on Tobacco800 Dataset

We performed the logo spotting task in the Tobacco800 dataset to verify the method's performance on documents from a different domain. Due to the best performance observed for the DocExplore dataset, the FCN block3 combined with 128-PCA was used to extract the features. As for the Horae dataset, all images and queries were reduced to 25% of their original size.

Table 8 presents our IR and Logo Spotting results compared to the state-of-the-art. The proposed method provided a higher result than the method presented in Wiggers et al. (2019a) for the top-50 and top-100 results in the IR task. Unlike the result obtained for the Image Retrieval task, the proposed method allowed a higher result than the method presented in Wiggers et al. (2019a) for the top-10 results in the Logo Spotting task. An important fact to be noticed is the proximity between the results obtained for IR and Logo

| Method                 | Image Retrieval Top-k (mAP) |        |        |        | Logo Spotting Top-k (mAP) |        |        |        |
|------------------------|-----------------------------|--------|--------|--------|---------------------------|--------|--------|--------|
|                        | 10                          | 25     | 50     | 100    | 10                        | 25     | 50     | 100    |
| Wiggers et al. (2018)  | 0.7200                      | 0.6100 | 0.5000 | 0.3500 | 0.6400                    | -      | -      | -      |
| Wiggers et al. (2019a) | 0.8720                      | 0.7350 | 0.5460 | 0.3060 | 0.7400                    | -      | -      | -      |
| Proposed method        | 0.8444                      | 0.7092 | 0.5586 | 0.5337 | 0.8494                    | 0.7048 | 0.5481 | 0.5110 |

Table 8 – Image Retrieval and Logo Spotting results for Tobacco800 (using the evaluation protocol presented in (WIGGERS et al., 2019a)).

Spotting tasks. There is only one or no query per page in most cases, so similar results represent a low occurrence of false positives on pages with queries.

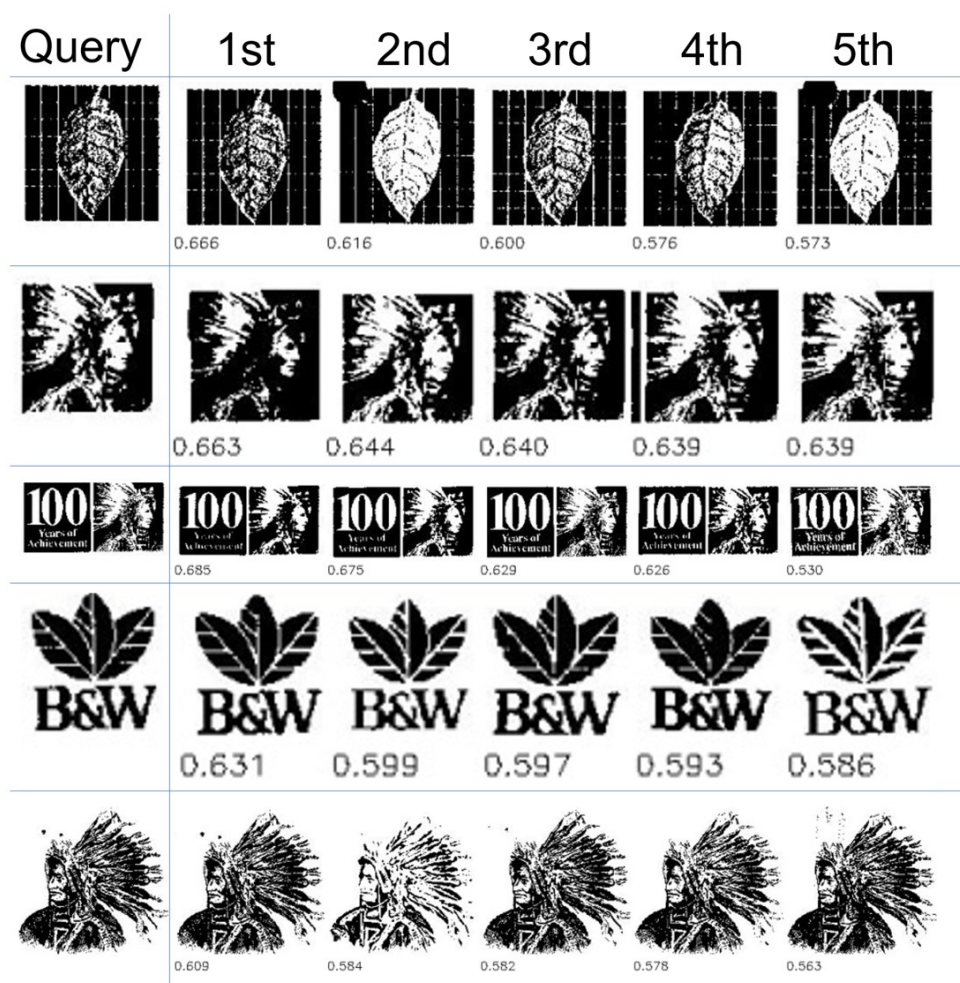


Figure 37 – Retrieval results for Tobacco800 dataset.

One of the main limitations of the Tobacco800 dataset is the lack of an evaluation protocol. Thus, different types of evaluations can be found in the literature, making it difficult to compare the results. A different number of queries and evaluation metrics have been used. Among the related works,

Jain & Doermann (2012) reported an mAP of 0.5900, Le et al. (2013) reported 0.8831 and Le et al. (2014) achieved 0.9115. We can also find works that use part of the dataset for training, addressing a different problem than presented in this work. That is the case of Rusiñol & Lladós (2010) which reports an mAP of 0.826.

As observed for the previous datasets, our method retrieved visually similar images, even with slight differences in visual and size. Figure 37 presents the top five results returned for some queries. It is important to note that the grayscale images or black and white did not reduce the retrieval quality, which means that the method is robust to differences in the color model used. However, the Tobacco800 dataset has some categories of queries with a significant size variation between objects. The proposed method could not return all occurrences in such cases, being necessary to use the same query with different sizes as input.

#### 4.1.4 Discussion and Analysis

Based on the results presented in this section, we can confirm hypotheses H1 and H2 of this research. The effectiveness of applying a method that combines an FCN with feature comparison using cross-correlation operations has been demonstrated across three different datasets. The obtained results indicate that this approach yielded favorable results for both IR and PS tasks, thus confirming hypothesis H1. Additionally, the achieved results outperformed existing methods in the literature.

Furthermore, we have also emphasized the robustness of features generated by an FCN trained on the ImageNet dataset, thereby confirming hypothesis H2. We have demonstrated that appropriately utilizing the intermediate layers of the network enabled a robust representation of the analyzed objects in different datasets, even without the need for additional training.

## 4.2 Experiments with the Method Using Binary Features

### 4.2.1 Results on DocExplore Dataset

As in the method with floating-type features, an intermediate layer of the VGG16 architecture was utilized for feature extraction. One of the

benefits of using an intermediate layer as opposed to a shallow one is that fewer operations are required in the cross-correlation step. To reduce the number of channels, decorrelate the data, and reduce the number of operations in the cross-correlation step, we applied Principal Component Analysis (PCA). To demonstrate the impact of feature reduction in the proposed method, Table 9 presents the results obtained with varying numbers of components in PCA.

| # of PCA Comp. | Float features |               |             | Binary features |               |             |
|----------------|----------------|---------------|-------------|-----------------|---------------|-------------|
|                | mAP IR         | mAP PS        | Memory (GB) | mAP IR          | mAP PS        | Memory (GB) |
| 8              | 0.6361         | 0.4743        | 0.96        | 0.5267          | 0.3739        | 0.08        |
| 16             | 0.7504         | 0.5741        | 1.89        | 0.6598          | 0.4957        | 0.13        |
| 32             | 0.7961         | 0.6259        | 3.73        | 0.7487          | 0.5855        | 0.24        |
| 64             | 0.8075         | 0.6392        | 7.43        | 0.7938          | 0.6321        | 0.47        |
| 128            | <b>0.8131</b>  | <b>0.6442</b> | 14.82       | <b>0.8032</b>   | <b>0.6381</b> | 0.93        |

Table 9 – Experimental results of the proposed method on the DocExplore dataset considering float and binary feature vectors with different sizes (number of PCA components). mAP for IR and PS tasks plus the memory consumption in GigaBytes

As can be observed in Table 9 and as expected, the higher the number of features the better the result obtained by the method in both tasks. However, the larger the number of features the larger is the space needed for indexing. It is important to note that besides optimizing the storage space and the processing time, PCA allows an improvement in the results in terms of mAP. Without the PCA operation the results provided by the method are respectively 0.7651 and 0.5815 for the IR and PS tasks. Even requiring more storage space, this result is lower than when using PCA with 128 and 64 components.

The most important step for optimizing the storage space needed for the indexing of features is binarization. The only adjustable parameter in the binarization process is the number of page images used to generate the global squeeze vector. In our experiments, we noticed no improvement when using more than 50 pages. If we compare in Table 9, the results obtained with float features and regular cross-correlation with those observed with binary features and XOR cross-correlation, we can see a slight loss in terms of accuracy performance. The decrease in mAP is 1.21% (from 0.8131 to 0.8032) for the IR task and 0.95% (from 0.6442 to 0.6381) for the PS task, while the savings of memory consumption is around 93% (from 14.82 GB to 0.93 GB).

Due to the slight difference in mAP between the results, we applied Friedman’s test. Once a significant p-value was observed, we conducted Nemenyi’s posthoc test. Figure 38 presents the results obtained for this test. As

can be observed, the test showed no significant difference between the binary method using PCA with 128 components and the method using float values generated by the PCA with 64 components. When the PCA with 64 components is used, there is no statistical difference on the mAP value between the binary and float representations.

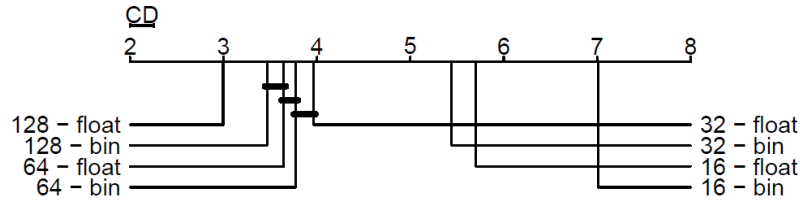


Figure 38 – Nemenyi test considering float and binary representations of different sizes (128, 64, 32 and 16)

The activation values are stored as 32-bit float data in most CNN implementations. The binarization of these values with the method presented in this work allows the reduction to only 1 bit, thus optimizing the memory space and the reading time of the stored files. In addition to the decrease in memory space, the proposed method uses a binary version of the cross-correlation operation, thus reducing processing time. The average search time was reduced by 13.3% when using the binary 128-feature vector compared to the float 128-feature representation.

Figure 39 presents the top 5 results for some queries of the DocExplore dataset. In this figure, the first image is the query used for the search and its size followed by the five resulting retrieved images with the highest similarity score for this query. Our method can detect patterns with slightly different styles, shapes, and sizes. On the second example of retrieval results illustrated by this figure, it is possible to notice that the method can identify similar patterns even with a difference in the background color. The third and fourth rows present the results for the categories with smaller and larger average sizes, respectively. In both cases, the method generated good results. The last query allows assessing the behavior of the method with a complex object. Using a particular face (King Henri II) to search for patterns with a similar face and clothing, it is relevant to point out that some images contain dozens of different faces which are not ranked in the top list.

Table 10 presents a comparison of the results obtained with other works

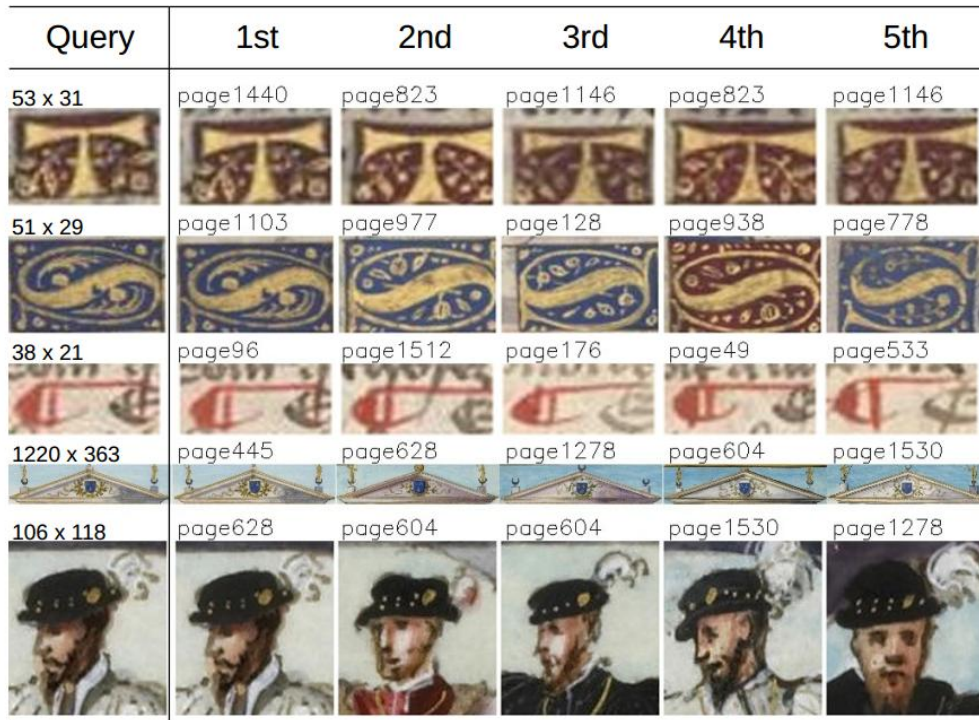


Figure 39 – Qualitative results obtained by applying the binary method on the DocExplore dataset

on the same dataset. All studies presented in this comparison follow the protocol proposed by (EN et al., 2016a). As we can see, the proposed approach presents a higher mAP than all other related works becoming a new state-of-the-art for DocExplore dataset. The observed improvement in mAP for the IR task is 38.46% higher than the best result found in the literature (EN et al., 2016d). Regarding the PS task, the proposed method presents a result 134.6% higher than the state-of-the-art (ÚBEDA et al., 2020). The superiority of the results, when compared to the works in the literature, serves as a factor in confirming hypothesis H3. The application of the proposed binarization strategy not only yielded satisfactory results but also demonstrated the ability to achieve these results in a shorter time frame and with reduced computational complexity compared to the method utilizing float features.

Similar to our work, (WIGGERS et al., 2019a; ÚBEDA et al., 2019; ÚBEDA et al., 2020) use CNN networks trained on other domains for feature extraction. However, a significant difference here is using an intermediate layer (shallow) of the CNN network and cross-correlation operation for comparison. Deep convolutional or fully connected layers for feature extraction usually de-



| Method                            | mAP IR        | mAP PS        |
|-----------------------------------|---------------|---------------|
| En et al. (2016a)                 | 0.5801        | 0.1569        |
| Wiggers et al. (2019b)            | 0.3860        | 0.1740        |
| Úbeda et al. (2020)               | 0.5770        | 0.2720        |
| Mohammed, Märgner & Ciotti (2021) | -             | 0.2510        |
| Dias et al. (2022)                | 0.5321        | 0.1996        |
| Proposed Method Float             | <b>0.8131</b> | <b>0.6442</b> |
| Proposed Method Binary            | 0.8032        | 0.6381        |

Table 10 – Results for DocExplore dataset (using the evaluation protocol presented in (EN et al., 2016a))

mand training on images of the problem domain since deeper layers usually carry strongly semantic information of the images used for training. It is important to remember that we cannot train on queries in our domain (ancient documents) since there is no prior knowledge of them.

Another common characteristic of the related works based on CNN is the use of feature vectors to compare the query and image candidates. We have used a strategy for detecting objects in heatmaps that avoids the non-maximum suppression process, which was adapted for the proposed binary representation using an XOR cross-correlation. This type of comparison allows simultaneous comparison of several parts of a query individually.

### 4.3 Final Considerations

This chapter presents the results obtained from our experiments and compares them with the methods presented in existing literature. The analysis was conducted on two proposed methods: using float features and using binarized features. The results from both methods showed promising performance for IR and PS. Thorough discussion of our findings enabled identification of the strengths and limitations of each method.

To validate the hypotheses of this research, an experimental protocol was applied utilizing three datasets. The proposed method, which float features, was evaluated both quantitatively and qualitatively. Quantitative evaluations were performed on the DocExplore and Tobacco800 datasets, while a qualitative evaluation was conducted on the Horae dataset. The results of the experiments confirmed hypothesis H1, demonstrating superior performance in

the IR and PS tasks compared to methods from the literature. Additionally, the experiments demonstrated the capability of the proposed method to retrieve objects of various sizes, shapes, and colors within the three datasets used.

Experiments were conducted using an FCN architecture based on the VGG16 network, trained on the ImageNet dataset. These experiments confirmed hypothesis H2. Despite utilizing transfer learning, the features obtained through the FCN enabled robust representation of the documents and queries, allowing for accurate identification of the searched objects. It should be noted that this was made possible by utilizing an intermediate layer within the FCN architecture.

One of the major limitations of the float-type feature method is the storage space and retrieval time required. However, the proposed binarization process effectively reduced these limitations by achieving a 93% reduction in storage space. Despite this reduction, the binary method resulted in a minimal decrease of 1.21% in mAP for the IR task and 0.95% for the PS task in the DocExplore dataset. These results validate hypothesis H3, as the use of binary features enables a significant reduction in computational complexity while maintaining the performance benefits of the proposed method.

## 5 Conclusions

In this work, novel solutions for IR and PS in digitized historical documents were presented. Different from existing methods in the literature, where feature vectors are used, a matrix-based approach was presented. The proposed methods use a Fully Convolutional Network (FCN) architecture trained on the Imagenet dataset for feature extraction and cross-correlation approaches to calculate the similarity between the query and the searched images. Such a strategy makes it possible to have a segmentation-free approach in which the whole query’s feature map is compared with all parts of the searched image. Two variations of the method were presented, one with float type features and the other with binary type features. The use of binary features is made possible by the proposed binarization strategy, where the features obtained through intermediate layers of the FCN are transformed using a global squeeze vector.

The presented methods were evaluated using the DocExplore dataset, and to verify their generalization ability, experiments were also conducted using the Horae and Tobacco datasets. The proposed methods proved to be effective in all tests conducted. The experiments with the DocExplore dataset demonstrated superior results compared to the state-of-the-art for both IR and PS. Furthermore, the experiments conducted for the PS task in the Horae dataset and logo spotting in the Tobacco800 dataset confirmed the high generalization capacity of the proposed method. All hypotheses were confirmed as the proposed methods showed consistent and superior performance in all experiments conducted across the different datasets.

This thesis has several conceptual contributions to the fields of IR and PS in historical document images. Firstly, it introduces a novel segmentation-free approach that combines a fully-convolutional model for feature extraction and a cross-correlation strategy for image comparison. The new approach differs from existing methods in the literature by using matrix-based representation and comparison, and represents a significant advancement in the state-of-the-art for the widely-used DocExplore dataset. Secondly, a new representation strategy composed of optimized features from intermediate layers of FCN networks enables transfer learning from models trained in different

domains. Thirdly, a binarization strategy focused on features obtained from intermediate activation layers of FCN networks is presented, using the proposed global squeeze vector. Additionally, a new XOR cross-correlation operation based on binary features is introduced to improve image comparison. Finally, a new strategy for object detection in heatmaps eliminates the need for non-maximum suppression processing.

In summary, the proposed methods presented in this work offers a promising approach for addressing the challenges associated with IR and PS in digitized historical documents. The results obtained from the experiments demonstrate the efficacy and generalization ability of the proposed method, and its potential for practical applications in real-world scenarios.

The results of this research demonstrate the effectiveness of the proposed methods in searching for images in historical documents. The application of these methods can help historians and researchers save time and resources in searching for information and illustrations in historical documents, enriching our knowledge of the past. The methods can be used to find important documents, illustrations, or maps that are often difficult to locate using traditional search methods. The ability to generalize the methods, obtained through the proposed feature extraction and treatment method, allows for various patterns to be used as query inputs. The ability to identify objects with different colors, textures, and in different contexts enables the search for images in historical documents with high efficiency and precision. Moreover, the developed technology can be applied in various fields, such as archives, library science, and museum studies, expanding its potential use. Thus, these methods can play a significant role in helping to preserve history and culture, enabling a better understanding and analysis of the past.

Although we can point out several contributions of this work, some limitations of the proposed method can be highlighted. The main difference between this work and others in the literature is the use of feature matrices to represent a single page or object. This characteristic requires more storage space and more operations to calculate the similarity between objects. To address this issue, we developed a binarization approach that can reduce storage space. Additionally, the use of binary features allowed us to apply an XOR cross-correlation approach, reducing the necessary processing and execution time of the method. Another limitation of the presented approach is

the inability to find objects of varied sizes directly, as in segmentation-based methods. This problem can be solved by using multiple inputs with varied sizes.

In addition to the limitations of the proposed method, we can mention the limitations of the problem addressed. One of the main issues in the IR and PS of historical documents field is the small number of annotated datasets. This limitation restricts the evaluation of works in the field, requiring the use of datasets from other problems to enable robust evaluation, as done in this work.

This work opens up new possibilities for the development of efficient methods for the retrieval of objects in image databases. As future work, it is possible to adapt the new approaches presented to other tasks and other data domains. In this way, it is possible to investigate the use of hand-drawn sketches as input query. The aim of this investigation is to enable historians to search for the desired pattern based on their own drawing, removing the need for manual search of an initial query for system input. In addition, it is possible to develop a training method without the use of system queries, thus allowing the use of deeper layers of the network used and the development of a system focused on the semantics of the objects. The use of deeper layers enables the retrieval of objects with greater differences, but semantically similar to the query. Furthermore, these layers generate smaller feature maps, optimizing the storage space and processing time of the system. Finally, it is possible to investigate the use of generative models for the generation of new images, based on the query. This would enable the system to generate new images that are similar to the query, thus allowing the user to explore a larger set of images as input.

# Bibliography

- AHMED, Rashad; AL-KHATIB, Wasfi G; MAHMOUD, Sabri. A survey on handwritten documents word spotting. *International Journal of Multimedia Information Retrieval*, Springer, v. 6, n. 1, p. 31–47, 2017. 30, 71
- ALAEI, Alireza; ROY, Partha Pratim; PAL, Umapada. Logo and seal based administrative document image retrieval: a survey. *Computer Science Review*, Elsevier, v. 22, p. 47–63, 2016. 29
- ALAHY, Alexandre; ORTIZ, Raphael; VANDERGHEYNST, Pierre. Freak: Fast retina keypoint. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2012. p. 510–517. 37
- ARANDJELOVIĆ, Relja; ZISSERMAN, Andrew. Three things everyone should know to improve object retrieval. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2012. p. 2911–2918. 37
- BALL, Gregory R; SRIHARI, Sargur N; SRINIVASAN, Harish. Segmentation-based and segmentation-free methods for spotting handwritten arabic words. In: *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*. [S.l.: s.n.], 2006. 32, 33
- BEITZEL, Steven M.; JENSEN, Eric C.; FRIEDER, Ophir. Map. In: \_\_\_\_\_. *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009. p. 1691–1692. ISBN 978-0-387-39940-9. Disponível em: <[https://doi.org/10-1007/978-0-387-39940-9\\_492](https://doi.org/10.1007/978-0-387-39940-9_492)>. 70
- BERTINETTO, Luca; VALMADRE, Jack; HENRIQUES, Joao F; VEDALDI, Andrea; TORR, Philip HS. Fully-convolutional siamese networks for object tracking. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 850–865. 63
- BLEI, David M; NG, Andrew Y; JORDAN, Michael I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003. 38
- BOILLET, Mélodie; BONHOMME, Marie-Laurence; STUTZMANN, Dominique; KERMORVANT, Christopher. Horae: an annotated dataset of books of hours. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. [S.l.: s.n.], 2019. p. 7–12. 46
- BRIECHLE, Kai; HANEBECK, Uwe D. Template matching using fast normalized cross correlation. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Optical Pattern Recognition XII*. [S.l.], 2001. v. 4387, p. 95–102. 42

- CALONDER, Michael; LEPETIT, Vincent; OZUYSAL, Mustafa; TRZCINSKI, Tomasz; STRECHA, Christoph; FUA, Pascal. Brief: Computing a local binary descriptor very fast. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 34, n. 7, p. 1281–1298, 2011. 37
- CAO, Yang; WANG, Changhu; ZHANG, Liqing; ZHANG, Lei. Edgel index for large-scale sketch-based image search. In: IEEE. *CVPR 2011*. [S.l.], 2011. p. 761–768. 36
- CASTELLANO, Giovanna; VESSIO, Gennaro. Towards a tool for visual link retrieval and knowledge discovery in painting datasets. In: SPRINGER. *Italian Research Conference on Digital Libraries*. [S.l.], 2020. p. 105–110. 71
- CHANG, Shi-Kuo; SHI, Qing-Yun; YAN, Cheng-Wen. Iconic indexing by 2-d strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, n. 3, p. 413–428, 1987. 29
- CHEN, Minmin; ZHENG, Alice; WEINBERGER, Kilian. Fast image tagging. In: *International conference on machine learning*. [S.l.: s.n.], 2013. p. 1274–1282. 17, 18
- CHENG, Ming-Ming; ZHANG, Ziming; LIN, Wen-Yan; TORR, Philip. Bing: Binarized normed gradients for objectness estimation at 300fps. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 3286–3293. 33
- CSURKA, Gabriella; DANCE, Christopher; FAN, Lixin; WILLAMOWSKI, Jutta; BRAY, Cédric. Visual categorization with bags of keypoints. In: PRAGUE. *Workshop on statistical learning in computer vision, ECCV*. [S.l.], 2004. v. 1, n. 1-22, p. 1–2. 38
- DAI, Jifeng; LI, Yi; HE, Kaiming; SUN, Jian. R-fcn: Object detection via region-based fully convolutional networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2016. p. 379–387. 24
- DIAS, Caio da S; BRITTO, Alceu De S; BARDDAL, Jean P; HEUTTE, Laurent; KOERICH, Alessandro L. Pattern spotting and image retrieval in historical documents using deep hashing. In: IEEE. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2022. p. 2869–2875. 51, 53, 78, 96
- DOVGALECS, Vladislavs; BURNETT, Alexandre; TRANOUEZ, Pierrick; NICOLAS, Stéphane; HEUTTE, Laurent. Spot it! finding words and patterns in historical documents. In: IEEE. *2013 12th International Conference on Document Analysis and Recognition*. [S.l.], 2013. p. 1039–1043. 51, 52
- EN, Sovann; NICOLAS, Stéphane; PETITJEAN, Caroline; JURIE, Frédéric; HEUTTE, Laurent. New public dataset for spotting patterns in medieval

- document images. *Journal of Electronic Imaging*, International Society for Optics and Photonics, v. 26, n. 1, 2016. 8, 11, 18, 42, 44, 45, 51, 52, 53, 70, 75, 77, 78, 79, 95, 96, 113
- EN, Sovann; PETITJEAN, Caroline; NICOLAS, Stéphane; HEUTTE, Laurent; JURIE, Frédéric. Pattern localization in historical document images via template matching. In: IEEE. *2016 23rd International Conference on Pattern Recognition (ICPR)*. [S.l.], 2016. p. 2054–2059. 43
- EN, Sovann; PETITJEAN, Caroline; NICOLAS, Stéphane; HEUTTE, Laurent; JURIE, Frédéric. Region proposal for pattern spotting in historical document images. In: IEEE. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.], 2016. p. 367–372. 33
- EN, Sovann; PETITJEAN, Caroline; NICOLAS, Stéphane; HEUTTE, Laurent. A scalable pattern spotting system for historical documents. *Pattern Recognition*, Elsevier, v. 54, p. 149–161, 2016. 32, 95
- FADAEI, Sadegh; AMIRFATTAHI, Rassoul; AHMADZADEH, Mohammad Reza. Local derivative radial patterns: a new texture descriptor for content-based image retrieval. *Signal Processing*, Elsevier, v. 137, p. 274–286, 2017. 36
- FELZENSZWALB, Pedro F; HUTTENLOCHER, Daniel P. Efficient graph-based image segmentation. *International journal of computer vision*, Springer, v. 59, n. 2, p. 167–181, 2004. 34
- FÖRSTNER, Wolfgang. A feature based correspondence algorithm for image matching. *ISPRS ComIII, Rovaniemi*, p. 150–166, 1986. 21
- GHOSH, Neha; AGRAWAL, Shikha; MOTWANI, Mahesh. A survey of feature extraction for content-based image retrieval system. In: SPRINGER. *Proceedings of International Conference on Recent Advancement on Computer and Communication*. [S.l.], 2018. p. 305–313. 20, 30
- GIOTIS, Angelos P; SFIKAS, Giorgos; GATOS, Basilis; NIKOU, Christophoros. A survey of document image word spotting techniques. *Pattern Recognition*, Elsevier, v. 68, p. 310–332, 2017. 18, 29
- GÓMEZ, Lluís; KARATZAS, Dimosthenis. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition*, Elsevier, v. 70, p. 60–74, 2017. 34
- GONTHIER, Nicolas; GOUSSEAU, Yann; LADJAL, Said; BONFAIT, Olivier. Weakly supervised object detection in artworks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 0–0. 71



- GU, Jiuxiang; WANG, Zhenhua; KUEN, Jason; MA, Lianyang; SHAHROUDY, Amir; SHUAI, Bing; LIU, Ting; WANG, Xingxing; WANG, Gang; CAI, Jianfei et al. Recent advances in convolutional neural networks. *Pattern Recognition*, Elsevier, v. 77, p. 354–377, 2018. 24
- GUO, Jing-Ming; PRASETYO, Heri; CHEN, Jen-Ho. Content-based image retrieval using error diffusion block truncation coding features. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 25, n. 3, p. 466–481, 2014. 35
- HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. 58
- HOFMANN, Thomas. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, Springer, v. 42, n. 1-2, p. 177–196, 2001. 38
- JAAKKOLA TOMMI; HAUSSLER, David. Exploiting generative models in discriminative classifiers. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1999. p. 487–493. 38
- JAIN, Rajiv; DOERMANN, David. Logo retrieval in document images. In: IEEE. *2012 10th IAPR International Workshop on Document Analysis Systems*. [S.l.], 2012. p. 135–139. 92
- JÉGOU, Hervé; DOUZE, Matthijs; SCHMID, Cordelia; PÉREZ, Patrick. Aggregating local descriptors into a compact image representation. In: IEEE. *2010 IEEE computer society conference on computer vision and pattern recognition*. [S.l.], 2010. p. 3304–3311. 38
- LASMAR, Nour-Eddine; BERTHOUMIEU, Yannick. Gaussian copula multivariate modeling for texture image retrieval using wavelet transforms. *IEEE Transactions on Image Processing*, IEEE, v. 23, n. 5, p. 2246–2261, 2014. 36
- LE, Viet Phuong; NAYEF, Nibal; VISANI, Muriel; OGIER, Jean-Marc; TRAN, Cao De. Document retrieval based on logo spotting using key-point matching. In: IEEE. *2014 22nd international conference on pattern recognition*. [S.l.], 2014. p. 3056–3061. 92
- LE, Viet Phuong; VISANI, Muriel; TRAN, Cao Dê; OGIER, Jean-Marc. Improving logo spotting and matching for document categorization by a post-filter based on homography. In: IEEE. *2013 12th International Conference on Document Analysis and Recognition*. [S.l.], 2013. p. 270–274. 92
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. 20, 53

- LEUTENEGGER, Stefan; CHLI, Margarita; SIEGWART, Roland Y. Brisk: Binary robust invariant scalable keypoints. In: IEEE. *2011 International conference on computer vision*. [S.l.], 2011. p. 2548–2555. 37
- LEWIS, David; AGAM, Gady; ARGAMON, Shlomo; FRIEDER, Ophir; GROSSMAN, D; HEARD, Jefferson. Building a test collection for complex document information processing. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. [S.l.: s.n.], 2006. p. 665–666. 48
- LEYDIER, Yann; OUJI, Asma; LEBOURGEOIS, Frank; EMPTOZ, Hubert. Towards an omnilingual word retrieval system for ancient manuscripts. *Pattern Recognition*, Elsevier, v. 42, n. 9, p. 2089–2105, 2009. 18
- LIN, Kevin; LU, Jiwen; CHEN, Chu-Song; ZHOU, Jie. Learning compact binary descriptors with unsupervised deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 1183–1192. 39
- LIU, Guang-Hai; LI, Zuo-Yong; ZHANG, Lei; XU, Yong. Image retrieval based on micro-structure descriptor. *Pattern Recognition*, Elsevier, v. 44, n. 9, p. 2123–2133, 2011. 36
- LIU, Haomiao; WANG, Ruiping; SHAN, Shiguang; CHEN, Xilin. Deep supervised hashing for fast image retrieval. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2064–2072. 39
- LIU, Peizhong; GUO, Jing-Ming; CHAMNONGTHAI, Kosin; PRASETYO, Heri. Fusion of color histogram and lbp-based features for texture image retrieval and classification. *Information Sciences*, Elsevier, v. 390, p. 95–111, 2017. 36
- LIU, Ying; ZHANG, Dengsheng; LU, Guojun. Region-based image retrieval with high-level semantics using decision tree learning. *Pattern Recognition*, Elsevier, v. 41, n. 8, p. 2554–2570, 2008. 35
- LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440. 20, 24, 59
- LOWE, David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, n. 2, p. 91–110, 2004. 36
- LU, Yi; GUO, Hong. Background removal in image indexing and retrieval. In: IEEE. *Proceedings 10th International Conference on Image Analysis and Processing*. [S.l.], 1999. p. 933–938. 31

- MANMATHA, Raghavan; HAN, Chengfeng; RISEMAN, Edward M. Word spotting: A new approach to indexing handwriting. In: IEEE. *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.], 1996. p. 631–637. 30
- MISHCHUK, Anastasiia; MISHKIN, Dmytro; RADENOVIC, Filip; MATAS, Jiri. Working hard to know your neighbor’s margins: Local descriptor learning loss. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2017. p. 4826–4837. 40
- MOHAMMED, Hussein; MÄRGNER, Volker; CIOTTI, Giovanni. Learning-free pattern detection for manuscript research. *International Journal on Document Analysis and Recognition (IJDAR)*, Springer, p. 1–13, 2021. 51, 52, 53, 78, 96
- NAIK, Jay; DOYLE, Scott; BASAVANHALLY, Ajay; GANESAN, Shridar; FELDMAN, Michael D; TOMASZEWSKI, John E; MADABHUSHI, Anant. A boosted distance metric: application to content based image retrieval and classification of digitized histopathology. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2009: Computer-Aided Diagnosis*. [S.l.], 2009. v. 7260, p. 72603F. 41
- NIKOLAIDOU, Konstantina; SEURET, Mathias; MOKAYED, Hamam; LIWICKI, Marcus. A survey of historical document image datasets. *arXiv preprint arXiv:2203.08504*, 2022. 28
- OJALA, Timo; PIETIKÄINEN, Matti; HARWOOD, David. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, Elsevier, v. 29, n. 1, p. 51–59, 1996. 35
- PARK, Gunhan; BAEK, Yunju; LEE, Heung-Kyu. Re-ranking algorithm using post-retrieval clustering for content-based image retrieval. *Information processing & management*, Elsevier, v. 41, n. 2, p. 177–194, 2005. 43
- PATIL, Sanjay; TALBAR, Sanjay. Content based image retrieval using various distance metrics. In: SPRINGER. *International Conference on Data Engineering and Management*. [S.l.], 2010. p. 154–161. 41
- PEKER, Kadir A. Binary sift: Fast image retrieval using binary quantized sift features. In: IEEE. *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*. [S.l.], 2011. p. 217–222. 37
- PERRONNIN, Florent; LIU, Yan; SÁNCHEZ, Jorge; POIRIER, Hervé. Large-scale image retrieval with compressed fisher vectors. In: IEEE. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.], 2010. p. 3384–3391. 38

- RADENOVIĆ, Filip; TOLIAS, Giorgos; CHUM, Ondřej. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 41, n. 7, p. 1655–1668, 2018. 39
- RAKTHANMANON, Thanawin; ZHU, Qiang; KEOGH, Eamonn J. Mining historical documents for near-duplicate figures. In: IEEE. *2011 IEEE 11th International Conference on Data Mining*. [S.l.], 2011. p. 557–566. 51, 52
- ROTHACKER, Leonard; RUSINOL, Marçal; FINK, Gernot A. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In: IEEE. *2013 12th International Conference on Document Analysis and Recognition*. [S.l.], 2013. p. 1305–1309. 43
- RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. Orb: An efficient alternative to sift or surf. In: IEEE. *2011 International conference on computer vision*. [S.l.], 2011. p. 2564–2571. 37
- RUSIÑOL, Marçal; LLADÓS, Josep. Efficient logo retrieval through hashing shape context descriptors. In: *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. [S.l.: s.n.], 2010. p. 215–222. 92
- SARVAIYA, Jignesh N; PATNAIK, Suprava; BOMBAYWALA, Salman. Image registration by template matching using normalized cross-correlation. In: IEEE. *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. [S.l.], 2009. p. 819–822. 42
- SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015. 20
- SEGUIN, Benoit; STRIOLO, Carlotta; KAPLAN, Frederic et al. Visual link retrieval in a database of paintings. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2016. p. 753–767. 71
- SHARMA, Nabin; MANDAL, Ranju; SHARMA, Rabi; PAL, Umapada; BLUMENSTEIN, Michael. Signature and logo detection using deep cnn for document image retrieval. In: IEEE. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.], 2018. p. 416–422. 29
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 58, 59
- SIRADJUDDIN, Indah Agustien; WARDANA, Wrida Adi; SOPHAN, Mochammad Kautsar. Feature extraction using self-supervised convolutional autoencoder for content based image retrieval. In: IEEE. *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*. [S.l.], 2019. p. 1–5. 39

SIVIC, Josef; ZISSERMAN, Andrew. Video google: A text retrieval approach to object matching in videos. In: IEEE. *Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2*. [S.l.], 2003. p. 1470. 36, 38

SOKIC, Emir; KONJICIJA, Samim. Phase preserving fourier descriptor for shape-based image retrieval. *Signal Processing: Image Communication*, IEEE, v. 40, p. 82–96, 2016. 36

SWAIN, Michael J; BALLARD, Dana H. Color indexing. *International journal of computer vision*, Springer, v. 7, n. 1, p. 11–32, 1991. 35

TIAN, Dong ping et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, Citeseer, v. 8, n. 4, p. 385–396, 2013. 36

TIAN, Yurun; FAN, Bin; WU, Fuchao. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 661–669. 40

TOLIAS, Giorgos; SICRE, Ronan; JÉGOU, Hervé. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 38, 43

TRANOUEZ, Pierrick; NICOLAS, Stéphane; DOVGALECS, Vladislavs; BURNETT, Alexandre; HEUTTE, Laurent; LIANG, Yiqing; GUEST, Richard; FAIRHURST, Michael. Docexplore: overcoming cultural and physical barriers to access ancient documents. In: *Proceedings of the 2012 ACM symposium on Document engineering*. [S.l.: s.n.], 2012. p. 205–208. 52

ÚBEDA, Ignacio; SAAVEDRA, Jose M; NICOLAS, Stéphane; PETITJEAN, Caroline; HEUTTE, Laurent. Pattern spotting in historical documents using convolutional models. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. [S.l.: s.n.], 2019. p. 60–65. 32, 40, 53, 95

ÚBEDA, Ignacio; SAAVEDRA, Jose M; NICOLAS, Stéphane; PETITJEAN, Caroline; HEUTTE, Laurent. Improving pattern spotting in historical documents using feature pyramid networks. *Pattern Recognition Letters*, Elsevier, v. 131, p. 398–404, 2020. 11, 40, 43, 51, 53, 54, 77, 78, 79, 95, 96

UIJLINGS, Jasper RR; SANDE, Koen EA Van De; GEVERS, Theo; SMEULDERS, Arnold WM. Selective search for object recognition. *International journal of computer vision*, Springer, v. 104, n. 2, p. 154–171, 2013. 33

VADIVEL, AKMSSA; MAJUMDAR, AK; SURAL, Shamik. Performance comparison of distance metrics in content-based image retrieval applications.

In: *International Conference on Information Technology (CIT), Bhubaneswar, India*. [S.l.: s.n.], 2003. p. 159–164. 41

VERMA, Manisha; RAMAN, Balasubramanian. Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval. *Multimedia Tools and Applications*, Springer, v. 77, n. 10, p. 11843–11866, 2018. 8, 35, 36

WANG, Fang; KANG, Le; LI, Yi. Sketch-based 3d shape retrieval using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 1875–1883. 36

WANG, Jingdong; HUA, Xian-Sheng. Interactive image search by color map. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM New York, NY, USA, v. 3, n. 1, p. 1–23, 2011. 35

WANG, Lijun; OUYANG, Wanli; WANG, Xiaogang; LU, Huchuan. Visual tracking with fully convolutional networks. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 3119–3127. 20, 24

WANG, Xing-yuan; CHEN, Zhi-feng; YUN, Jiao-jiao. An effective method for color image retrieval based on texture. *Computer Standards & Interfaces*, Elsevier, v. 34, n. 1, p. 31–35, 2012. 36

WANG, Xiang-Yang; ZHANG, Bei-Bei; YANG, Hong-Ying. Content-based image retrieval by integrating color and texture features. *Multimedia tools and applications*, Springer, v. 68, n. 3, p. 545–569, 2014. 35, 36

WANG, Zhiyong; CHI, Zheru; FENG, Dagan. Shape based leaf image retrieval. *IEE Proceedings-Vision, Image and Signal Processing, IET*, v. 150, n. 1, p. 34–43, 2003. 36

WENGERT, Christian; DOUZE, Matthijs; JÉGOU, Hervé. Bag-of-colors for improved image search. In: *Proceedings of the 19th ACM international conference on Multimedia*. [S.l.: s.n.], 2011. p. 1437–1440. 34, 35

WIGGERS, Kelly L; BRITTO, Alceu S; HEUTTE, Laurent; KOERICH, Alessandro L; OLIVEIRA, Luiz Eduardo S. Document image retrieval using deep features. In: IEEE. *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2018. p. 1–8. 33, 91

WIGGERS, Kelly L; BRITTO, Alceu S; HEUTTE, Laurent; KOERICH, Alessandro L; OLIVEIRA, Luiz S. Image retrieval and pattern spotting using siamese neural network. In: IEEE. *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2019. p. 1–8. 11, 33, 54, 90, 91, 95

- WIGGERS, Kelly Lais; JUNIOR, Alceu de Souza Britto; KOERICH, Alessandro Lameiras; HEUTTE, Laurent; OLIVEIRA, Luiz Eduardo Soares de. Deep learning approaches for image retrieval and pattern spotting in ancient documents. *arXiv preprint arXiv:1907.09404*, 2019. 40, 51, 53, 70, 78, 96
- WU, Dayan; DAI, Qi; LIU, Jing; LI, Bo; WANG, Weiping. Deep incremental hashing network for efficient image retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 9069–9077. 39
- YEE, Ka-Ping; SWEARINGEN, Kirsten; LI, Kevin; HEARST, Marti. Faceted metadata for image search and browsing. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.: s.n.], 2003. p. 401–408. 17, 18
- ZAFAR, Afia; AAMIR, Muhammad; NAWI, Nazri Mohd; ARSHAD, Ali; RIAZ, Saman; ALRUBAN, Abdulrahman; DUTTA, Ashit Kumar; ALMOTAIRI, Sultan. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, MDPI, v. 12, n. 17, p. 8643, 2022. 26
- ZAGORIS, Konstantinos; PRATIKAKIS, Ioannis; GATOS, Basilis. Unsupervised word spotting in historical handwritten document images using document-oriented local features. *IEEE Transactions on Image Processing*, IEEE, v. 26, n. 8, p. 4032–4041, 2017. 33
- ZHANG, Dengsheng; LU, Guojun. Generic fourier descriptor for shape-based image retrieval. In: IEEE. *Proceedings. IEEE International Conference on Multimedia and Expo*. [S.l.], 2002. v. 1, p. 425–428. 36
- ZHANG, Dengsheng; LU, Guojun. Review of shape representation and description techniques. *Pattern recognition*, Elsevier, v. 37, n. 1, p. 1–19, 2004. 36
- ZHANG, Ethan; ZHANG, Yi. Average precision. In: \_\_\_\_\_. *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009. p. 192–193. ISBN 978-0-387-39940-9. Disponível em: <[https://doi.org/10.1007/978-0-387-39940-9\\_482](https://doi.org/10.1007/978-0-387-39940-9_482)>. 70
- ZHENG, Liang; YANG, Yi; TIAN, Qi. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 40, n. 5, p. 1224–1244, 2017. 53
- ZHOU, Wengang; LI, Houqiang; TIAN, Qi. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017. 29, 34, 36

ZHU, Guangyu; DOERMANN, David. Automatic document logo detection. In: IEEE. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. [S.l.], 2007. v. 2, p. 864–868. 48

ZHU, Guangyu; ZHENG, Yefeng; DOERMANN, David; JAEGER, Stefan. Signature detection and matching for document image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 31, n. 11, p. 2015–2031, 2008. 29

ZHU, Han; LONG, Mingsheng; WANG, Jianmin; CAO, Yue. Deep hashing network for efficient similarity retrieval. In: *Thirtieth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2016. 39

ZHU, Qiang; KEOGH, Eamonn. Mother fugger: mining historical manuscripts with local color patches. In: IEEE. *2010 IEEE International Conference on Data Mining*. [S.l.], 2010. p. 699–708. 51, 52



# Appendix

# APPENDIX A – Supplementary Experiments

## A.1 Detailed Exploration of Experimental Results

Throughout this thesis, multiple results and analyses of conducted experiments have been presented. The purpose of this appendix is to enhance the understanding of the method and results, providing a more comprehensive and detailed view of the analyses conducted during the research. The results presented in this appendix not only expand the initial scope of the investigation but also reaffirm concepts already established in the literature.

The detailed experiments in this appendix serve the specific purpose of refining the precision and reliability of previously reached conclusions while justifying the choices made at each stage of the proposed method. This more refined methodological approach contributes to a more solid and robust comprehension of the details of the proposed method.

It is worth noting that, in all presented experiments, a subset of queries covering all pages of the DocExplore dataset was used. This reduced dataset was applied to decrease the execution time of preliminary experiments and optimize the method implementation process. Each test was conducted individually, potentially resulting in distinct baselines, as the central idea of the experiments is to analyze the impact of each stage, irrespective of the overall method result.

It is crucial to highlight that, due to the execution of some experiments during the method implementation phase, baseline results may be lower than the outcome. This is a pertinent aspect to consider when interpreting the findings. For reference, when using the selected subset, the method presented in (EN et al., 2016a) yielded a mAP of 0.4642 for IR and 0.1387 for PS.

### A.1.1 L2 normalization

The feature normalization is one of the stages which has a considerable impact on the results of the proposed method. To this end, the L2 normalization is used. Table 11 presents the results obtained with and without normalization. Two normalization approaches were tested: the first normalizes each channel individually, while the second normalizes all the elements of different channels positioned at the  $x, y$  coordinate.

| Method                       | mAP IR        | mAP PS        |
|------------------------------|---------------|---------------|
| Baseline                     | 0.6228        | 0.4722        |
| L2 norm - each channel       | 0.7186        | 0.5789        |
| L2 norm - different channels | <b>0.8075</b> | <b>0.6392</b> |

Table 11 – Results of feature normalization for the DocExplore subset.

### A.1.2 Heatmap Normalization

One of the essential stages of the method involves normalizing the heatmap obtained through the correlation operation. This step is important, since some images have a greater variety of specific colors or textures, while others are simpler, containing only text and small objects. The expectation is that the network weights will increase the values of the feature maps for documents with colors and textures. Higher values tend to generate more expressive results in the convolution with the query, especially when the query attributes have more positive than negative values. The approach chosen for normalization was to reduce each heatmap element to the average of the entire heatmap. Table 12 shows the results results obtained by this strategy for the DocExplore subset.

| Method                | mAP IR        | mAP PS        |
|-----------------------|---------------|---------------|
| Baseline              | 0.5790        | 0.3639        |
| Heatmap Normalization | <b>0.6311</b> | <b>0.4054</b> |

Table 12 – Results of heatmap normalization for the DocExplore subset.

### A.1.3 Number of Selected Regions

One of the variables of the proposed method is the number of regions selected in the heatmap generated. In the results presented, 15 regions were

adopted, and Table 13 shows the results obtained during the experiments to determine this value. As can be seen, a reduction in the number of regions results in a slight increase in performance for IR. In addition, using a smaller number of regions offers the advantage of reducing processing time. However, this approach also has a disadvantage, evidenced by a decline in the result for PS. This decline is attributed to the fact that many pages in the dataset contain multiple occurrences of the same query, resulting in the system ignoring some objects. Considering this characteristic, the choice was made to maintain the use of 15 regions, even though this results in lower performance for IR. This thoughtful choice takes into account the need to balance processing efficiency with preserving quality in pattern identification.

| # Regions | mAP IR        | mAP PS        |
|-----------|---------------|---------------|
| 3         | <b>0.6287</b> | 0.3884        |
| 4         | 0.6277        | 0.4352        |
| 5         | 0.6268        | 0.4499        |
| 6         | 0.6260        | 0.4568        |
| 7         | 0.6254        | 0.4615        |
| 10        | 0.6242        | 0.4684        |
| 15        | 0.6228        | <b>0.4722</b> |
| 20        | 0.6219        | 0.4709        |

Table 13 – Results of using different numbers of regions for the DocExplore subset.

#### A.1.4 Use of Multiple Inputs

One of the limitations of the proposed method is the identification of objects with different sizes than in the original query. It is assumed that objects can be smaller or larger than the input query. To reduce this problem, we propose an approach where the query is resized and the system is applied several times. To enable comparison, the size of the input image needs to be standardized. For this process, different query input formats were evaluated, illustrated in figure 40. The Table 14 shows the results obtained.

As can be seen, the most effective approach was to use mosaics in the input; however, it is important to note that these results are still lower than those obtained without this process. The lower results were due to the fact that the DocExplore dataset does not have a large size variation in its objects, which meant that this process could not be properly validated. However, tests

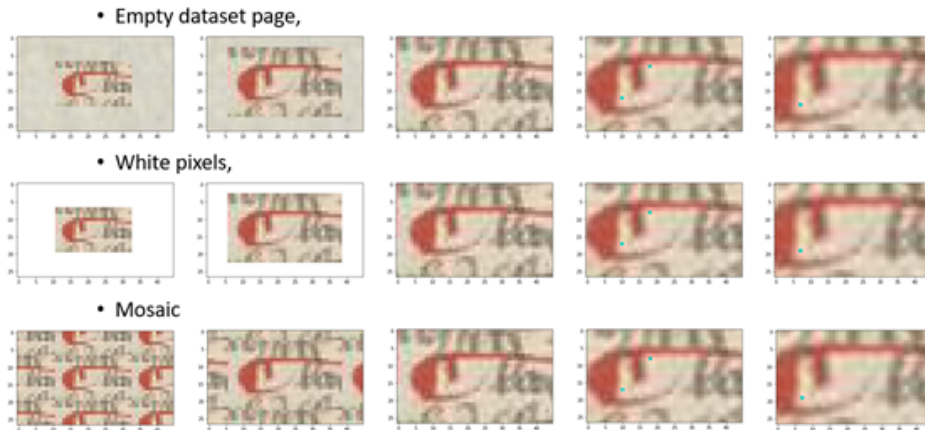


Figure 40 – Different input styles evaluated in experiments with multiple inputs.

| Input Format | mAP IR        | mAP PS        |
|--------------|---------------|---------------|
| Baseline     | <b>0.8221</b> | <b>0.6862</b> |
| Empty Page   | 0.6920        | 0.5864        |
| White Pixels | 0.6826        | 0.5666        |
| Mosaic       | 0.7668        | 0.6544        |

Table 14 – Results of using multiple inputs for the DocExplore subset.

in isolated cases, presented in the results chapter, showed that the use of multiple inputs worked to address the limitation of queries with different sizes.

### A.1.5 Gaussian Filter

In order to improve the comparisons made, tests were carried out using a Gaussian filter on the query features. In this investigation, the Gaussian filter was applied after the normalization operation. The purpose of this approach is to assign different weights to different regions of the query, reducing the value of the borders and giving greater importance to its center. The results of this experiment, considering different sigma values, are presented in Table 15.

| Method                          | mAP IR        | mAP PS        |
|---------------------------------|---------------|---------------|
| Baseline                        | <b>0.8170</b> | <b>0.6809</b> |
| Gaussian Filter $\sigma = 0.25$ | 0.8166        | 0.6808        |
| Gaussian Filter $\sigma = 0.50$ | 0.8080        | 0.6735        |
| Gaussian Filter $\sigma = 0.75$ | 0.7942        | 0.6574        |

Table 15 – Results of experiments with a Gaussian filter applied to query features for the DocExplore subset.

As observed, in all cases, there was a decrease in the results, with the larger sigma values generating lower results. The primary determinant for this reduction lies in the fact that the center of the query does not contain all the necessary information for its complete representation. An illustrative example is the comparison between the letters Q, O, and D, where the distinction occurs at the extremities. However, it is relevant to note that, in some specific cases, the use of the filter enabled superior results.

### A.1.6 Sobel Filter

Another experiment aimed at optimizing the quality of the features was to apply the Sobel filter to the query and image features. In this experiment, the Sobel filter was used on the features of queries and pages after extraction by FCN. The idea behind the Sobel filter is to detect the contours present in the images and use this information to make the comparison.

As shown in Table 16, the adoption of this strategy resulted in a reduction in the results for both tasks studied. For this reason, this filter was not incorporated into the method presented.

| Method       | mAP IR        | mAP PS        |
|--------------|---------------|---------------|
| Baseline     | <b>0.8170</b> | <b>0.6809</b> |
| Sobel Filter | 0.7390        | 0.5962        |

Table 16 – Results of experiments with a Sobel filter for the DocExplore subset.

### A.1.7 HOG for Pos-processing

This experiment aimed to verify if a post-processing step using Histogram of Oriented Gradients (HOG) features has a positive impact on the results. In this experiment, both the original query and all identified regions were resized to 128x128. After resizing, the HOG features were extracted and compared with the cosine similarity. After the comparison, the results were re-ordered based on the values obtained. The results (Table 17) revealed that this strategy led to a reduction in the obtained results. Although this approach was not incorporated into the method, the results were important in illustrating the superiority of the features obtained by the employed FCN.

| <b>Method</b> | <b>mAP PS</b> |
|---------------|---------------|
| Baseline      | <b>0.6339</b> |
| HOG 8x8       | 0.3699        |
| HOG 16x16     | 0.3384        |

Table 17 – Results of experiments with pos-processing for the DocExplore subset.

### A.1.8 Threshold

To reduce the number of results returned by the method, we implemented a process using thresholds. The idea is to discard results below a certain threshold, since they may be considered too low to be valid. Table 18 shows the results of the proposed method for different threshold values. As the table illustrates, there was no difference in the results between using the 0.1 threshold and the system with no threshold. Therefore, we decided to use the value of 0.1 in all the experiments performed.

| <b>Method</b> | <b>mAP IR</b> | <b>mAP PS</b> |
|---------------|---------------|---------------|
| Baseline      | <b>0.8170</b> | <b>0.6809</b> |
| Threshold 0.2 | 0.8142        | 0.6793        |
| Threshold 0.1 | <b>0.8170</b> | <b>0.6809</b> |

Table 18 – Results of the proposed method with varying threshold values for the DocExplore subset.